



Universidad de Granada

DARP: A NEW ROUTING ALGORITHM FOR LARGE COMMUNICATION INFRASTRUCTURES

Memoria presentada por

FRANCISCO JOSÉ ESTÉVEZ ORTIZ

Para optar al grado de
**DOCTOR POR LA UNIVERSIDAD DE GRANADA CON MENCIÓN
INTERNACIONAL**

perteneciente al programa de doctorado
TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Fdo. Francisco José Estévez Ortiz

Abril 2016

VISTO BUENO

El **Dr. D. Jesús González Peñalver**, Profesor Titular de Universidad en el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada y el **Prof. Dr. D. Peter Glösekötter**, Catedrático de Universidad del Departamento de Ingeniería Electrica e Informática de la Universidad de Ciencias Aplicadas de Münster,

CERTIFICAN:

Que la memoria titulada:

"DARP: A new routing algorithm for large communication infrastructures"

ha sido realizada por **D. Francisco José Estévez Ortiz** bajo nuestra dirección en conjunto entre los Departamentos de Arquitectura y Tecnología de Computadores de la Universidad de Granada e Ingeniería Electrica e Informática de la Universidad de Ciencias Aplicadas de Münster para optar al grado de **Doctor por la Universidad de Granada con Mención Internacional**.

En Granada, a 28 de Abril 2016.

Los Directores de la tesis doctoral:

Fdo. Jesús González Peñalver

Fdo. Peter Glösekötter

COMPROMISO DE RESPETO DE LOS DERECHOS DE AUTOR/A

El doctorando **D. Francisco José Estévez Ortiz** y los directores de la tesis **Dr. D. Jesús González Peñalver** y **Prof. Dr. D. Peter Glösekötter**.

Garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

En Granada, a 28 de Abril 2016.

Los Directores de la tesis doctoral:

Fdo. Jesús González Peñalver

Fdo. Peter Glösekötter

A ti, por estar siempre a mi lado y apoyarme con mis más locas ideas.

To you, for always being beside me and support my craziest ideas.

RESUMEN

Esta tesis doctoral presenta dos algoritmos, un algoritmo de enruteamiento para redes inalámbricas de sensores basadas en el estándar IEEE 802.15.4 y un algoritmo de auto-selección de enlaces de comunicaciones basado en la calidad del enlace que permite auto-ajustar de forma dinámica el rol que los nodos desarrollan en el algoritmo de enruteamiento. El algoritmo de enruteamiento está diseñado para ofrecer un bajo tiempo de convergencia y un uso reducido de mensajes de control en entornos en los que una baja densidad de nodos hace difíciles las comunicaciones. Con esta motivación, se ha desarrollado esta tesis doctoral, la cual presenta las siguientes aportaciones científicas:

La creación de un algoritmo de auto-selección de enlaces de comunicaciones y de auto-ajuste del rol ejecutado por cada nodo en la red cuando se inicializa. Mediante el uso de dicho algoritmo se minimiza el tiempo de formación de la red, el tráfico en la creación de la red y el consumo de energía.

La creación de un algoritmo de enruteamiento, el cual permite minimizar el tráfico debido al uso de sub-redes virtuales, así como organizar jerárquicamente la topología de la red. Se ha desarrollado un modelo de simulación que permita afinar los diferentes parámetros que caracterizan el algoritmo y su testeo previa implantación.

La integración de una versión del algoritmo de enruteamiento en una pila de comunicaciones, de forma que conformen un protocolo de comunicaciones que permita explotar los conceptos de localidad espacial y organización lógica en los que se basa el algoritmo de enruteamiento desarrollado.

ABSTRACT

This Ph.D. presents two algorithms, one is a routing algorithm for IEEE 802.15.4-based wireless sensor networks. The second is a best-link selection algorithm based on the quality of the links, which allows dynamically adjusting the role of a node in the routing algorithm. The routing algorithm is designed to offer a low convergence time and a reduced use of control messages in low-density environments. Due to these objectives, this Ph.D. provides the following main scientific contributions:

The first one is the development of a best-link selection algorithm, which auto-adjusts the role of a node. Using this algorithm, the convergence time, message overhead and energy consumption during the formation phase are minimized

The second one is the creation of a routing algorithm, which minimizes the traffic due to the use of virtual sub-networks, as well as hierarchically organizes the network topology. A simulation model for the routing algorithm has been developed to be able to tune the different configuration parameters that characterize the routing algorithm and in order to test them before deployments.

The third one is the integration of the routing algorithm in a wireless communication stack, so that form a communication protocol, which allows exploiting the principle of locality and the logical organization on which the routing algorithm is based.

PUBLICACIONES CIENTÍFICAS

Algunas de las ideas, imágenes y datos que se exponen en esta Tesis Doctoral han aparecido publicadas previamente en algunas publicaciones:

- **F. Estevez, G. Rebel, J. González, P. Gloesekoetter** *DARP: Dynamic and adaptive radio protocol for Wireless Sensor Networks*. Aceptado en *Electronic Letters*, 2014. Índice de Impacto: 0.93.
- **F. Estevez, G. Rebel, J.M. Castillo-Secilla, P. Gloesekoetter, J. González** *Cognitive Radio as solution for Saturation Problems in Wireless Sensor Networks: A survey of Coginitive Radio policies*. En las Actas del V Jornadas de Computación Empotrada: *Conferencia JCE2014, Sociedad de Arquitectura y Tecnología de Computadores (SARTECO), Valladolid (España)*, 2014.

Things may come to those who wait, but only the things left by those who hustle.

— Abraham Lincoln.

AGRADECIMIENTOS

Muchas personas se han cruzado en mi camino a lo largo de estos años y me gustaría, otorgarles parte de este mérito.

He tenido la suerte de aprender mucho de ellos, en los buenos y en los malos momentos, por ello me gustaría comenzar agradeciendo a todos aquellos que en algún momento han colaborado activamente en esta, mi tesis doctoral. Jose Manuel Palomares, Miguel Damas, José Luis Carmona, Jose María Castillo y por supuesto mis directores, Jesús González y Peter Gloesekoetter, sin los cuáles nada de esto sería posible.

Jesús, gracias por aguantar todas mis consultas, todas las ausencias y sobre todo, gracias por servirme de faro en tantos momentos de dificultad, sin tu ayuda, esta tesis no sería posible.

Peter, a ti debo agradecerte no solo por tus consejos y ayuda, sino por haberme enseñado tanto y sobretodo por haberme brindado la oportunidad de crecer, crecer a nivel personal y a nivel profesional como nunca habría imaginado, gracias de corazón.

No puedo olvidar a un gran amigo y compañero, Jose María, con el que tanto en común comparto y que tanto me ha aconsejado y ayudado, incluso en los peores días.

Por supuesto, a mi familia, a mi hermana, que tantos dolores de cabeza me ha dado, pero a la que tanto aprecio y que de alguna manera, siempre me ha apoyado. A mi padre y a mi difunta madre, que espero se sientan orgullosos, después de todo el esfuerzo que pusieron en educarme, gracias. Si de algo estoy orgulloso, no es de esta tesis, es de los valores y la educación que me habéis dado y por supuesto de que vosotros seáis mis padres.

No podría olvidar a mi pareja Cristina, ella es quien me ha aguantado, quien ha aprendido y a quien he martirizado con mi tesis. Ella ha sido el gran apoyo de esta aventura y ella es quien debe sentirse parte de este éxito. Ya sabes que a partir de hoy, nos esperan grandes aventuras.

Por último, no quiero olvidar a D. José Jurado, él me hizo recapacitar y tomar la decisión que me ha permitido llegar hasta aquí, gracias José.

LISTA DE CONTENIDOS

i	LARGE SUMMARY (ENGLISH VERSION)	1
1	THESIS SUMMARY	3
1.1	Short Introduction	4
1.2	Short Literature Review	5
1.3	Dynamic and Adaptive Routing Protocol (DARP)	7
1.3.1	Header and message types	8
1.3.2	DARAL Description	10
1.4	Experiment Summary	13
1.4.1	DARP Behavior analysis	13
1.4.2	Comparative study of protocol performance	13
1.4.3	Study of DARP additional features	14
ii	INTRODUCCIÓN	15
2	INTRODUCCIÓN	17
2.1	Motivación	18
2.2	Campos de aplicación	19
2.2.1	Sistemas agrarios inteligentes	20
2.2.2	Ciudades inteligentes	20
2.2.3	Internet de las Cosas	21
2.3	Marco de trabajo	23
2.3.1	Caracterización de las comunicaciones en WSN	23
2.3.2	Densidad de nodos en WSN	24
2.3.3	Tiempo de convergencia en WSN	25
2.3.4	Número de Mensajes de control en WSN	26
2.4	Planteamiento	26
2.4.1	Problemática	26
2.4.2	Densidad de nodos asociada a los grandes escenarios	27
2.4.3	Objetivos científicos de la Tesis	30
2.5	Estructura de la Tesis	31
iii	REVISIÓN BIBLIOGRÁFICA	33
3	COMUNICACIONES INALÁMBRICAS	35
3.1	Estándares de comunicaciones inalámbricas	37
3.1.1	ISA100.11a	37
3.1.2	ISO 18000-7	37
3.1.3	IEEE 802.11	38
3.1.4	IEEE 802.15.1	39
3.1.5	IEEE 802.15.3	39
3.1.6	IEEE 802.15.4	40
3.2	Algoritmos de capa de red	43
3.2.1	Algoritmos de Enrutamiento	43
3.2.2	Algoritmos de Enrutamiento basados en Clustering	47
3.3	Dinamismo en una WSN	48
3.4	Conclusiones	49

iv	MÉTODO Y MATERIALES	51
4	PROTOCOLO DARP	53
4.1	DARP: Dynamic and Adaptive Routing Protocol	53
4.1.1	DARAL: Dynamic and Adaptive Routing ALgorithm	54
4.1.2	Cabeceras y tipos de mensajes	56
4.1.3	DRSP: Dynamic Role Selection Process	59
4.1.4	Sub-redes Virtuales	62
4.2	Conclusiones	64
v	RESULTADOS EXPERIMENTALES	67
5	ANÁLISIS DE COMPORTAMIENTO DE DARP	69
5.1	Configuración de las capas PHY y MAC	70
5.1.1	Comunicaciones Inalámbricas	70
5.1.2	Simulador	70
5.1.3	Configuración del estándar IEEE 802.15.4	72
5.2	Metodología de ajuste de los parámetros de DARAL	73
5.3	Ánalisis de la configuración óptima propuesta	74
5.3.1	Umbral de mínima calidad: TH _{baselevel}	74
5.3.2	Umbral de decisión de rol: TH _{role}	77
5.3.3	Límite de nodos por sub-red: L _{nodes}	80
5.3.4	Límite de reintentos: DARAL_MAX_RETRIES	83
5.3.5	Temporizador de conexión T _{link}	86
5.3.6	Tiempo de vida de los enlaces T _{alive}	89
5.3.7	Temporizador de desconexión T _{down}	91
5.3.8	Temporizador de reconexión T _{reconnect}	94
5.3.9	Tiempo de espera para respuestas T _{ack}	97
5.4	Conclusiones	100
6	ESTUDIO COMPARATIVO DE DIFERENTES PROTOCOLOS PARA WSN	103
6.1	Configuración de los Experimentos	104
6.1.1	Protocolos de enrutamiento más utilizados	104
6.1.2	Configuración de los escenarios de simulación	104
6.1.3	Configuración de la aplicación	106
6.1.4	Obtención de resultados estadísticos	107
6.2	Resultados experimentales para el análisis del Tiempo de Convergencia	107
6.2.1	Cálculo del tiempo de convergencia	107
6.2.2	Tamaño del escenario: PEQUEÑO	108
6.2.3	Tamaño del escenario: MEDIANO	110
6.2.4	Tamaño del escenario: GRANDE	112
6.3	Resultados experimentales para el análisis del Número de Mensajes de Control Enviados	114
6.3.1	Cálculo del número de mensajes de control enviados	114
6.3.2	Tamaño del escenario: PEQUEÑO	115
6.3.3	Tamaño del escenario: MEDIANO	118
6.3.4	Tamaño del escenario: GRANDE	120
6.4	Conclusiones	124
7	EVALUACIÓN DE OTRAS CARACTERÍSTICAS ADICIONALES	125
7.1	Análisis de la escalabilidad	125
7.1.1	Escenarios Muy Grandes	126

7.1.2	Escenarios Enormes	128
7.2	Análisis del consumo de energía durante la fase de conexión	130
7.3	Análisis del aprovechamiento del canal	131
7.4	Análisis de la minimización del número de colisiones	133
7.5	Análisis de la respuesta ante eventos dinámicos	133
vi	DISCUSIÓN Y CONCLUSIONES	137
8	DISCUSIÓN	139
8.1	Discusión sobre la configuración óptima del protocolo de enrutamiento para grandes escenarios	140
8.2	Discusión sobre el protocolo de comunicaciones y los algoritmos propuestos	141
8.2.1	Mejora en la creación de rutas en grandes escenarios	142
8.3	Discusión sobre la evaluación de características adicionales de DARP	143
8.4	Conclusiones	145
9	CONCLUSIONES	147
9.1	Producción científica	148
9.2	Principales aportaciones científicas	149
9.3	Investigaciones futuras	150
vii	DISCUSSION AND CONCLUSIONS (ENGLISH VERSION)	151
10	DISCUSSION	153
10.1	Discussion about the optimum configuration of DARAL for routing in large scenarios	154
10.2	Discussion on the routing protocol and the algorithms	155
10.2.1	Improvement in the route creation for large scenarios	155
10.3	Discussing the additional features of DARP	157
10.4	Conclusions	158
11	CONCLUSIONS	159
11.1	Scientific Production	160
11.2	Main Scientific Contributions	161
11.3	Future Research	162
viii	APÉNDICES	163
A	PLATAFORMAS ACTUALES EN WSN	165
A.1	Plataformas Hardware para WSN	166
A.1.1	MICAz	166
A.1.2	TelosB/Tmote Sky	166
A.1.3	WaspMote	167
A.1.4	Arduino/Genuino	167
A.1.5	RedBee-Econotag	168
A.2	Plataformas Software para WSN	168
A.2.1	TinyOS	168
A.2.2	Contiki	169
A.2.3	Nano-RK	170
A.2.4	FreeRTOS	172
A.3	Plataformas de Simulación para WSN	172
A.3.1	NS-2/NS-3	173

A.3.2	OMNeT++/OMNEST	173
A.3.3	TOSSIM	174
A.3.4	Cooja	175
B	OMNET++ TUTORIAL	177
B.1	First Step: Installing a Framework	178
B.2	From the TicToc Tutorial to a real Node	179
B.3	Useful Functions and tips	187
B.3.1	cModule Class	188
B.3.2	ModuleAccess.cc	188
B.3.3	cMessage	188
B.3.4	cPacket Class	189
B.3.5	Additional Tips	189
B.4	Example of a real network	190
B.5	Additional Documentation	190
C	DARP TECHNICAL MANUAL	191
C.1	Specifications	191
C.1.1	Related Documents	192
C.2	Implementation Details	192
C.2.1	State Machine	193
C.2.2	Virtual PAN ID	193
C.2.3	Available Operations in DARP	194
C.2.4	Frame Format	198
ix	BIBLIOGRAFÍA	203
BIBLIOGRAFÍA		205

LISTA DE FIGURAS

Figura 2.1	Cabezas de ganado en un latifundio uruguayo.	19
Figura 2.2	Ejemplo de Ciudad Inteligente y sus diferentes servicios.	21
Figura 2.3	Ejemplo de Internet de las Cosas y sus diferentes áreas de aplicación.	22
Figura 2.4	Evolución del tiempo de convergencia respecto a la densidad para <i>RPL</i> .	28
Figura 2.5	Evolución del tiempo de convergencia respecto a la densidad para <i>AODV</i> .	28
Figura 2.6	Evolución del número de mensajes de control enviados respecto a la densidad para <i>RPL</i> .	29
Figura 2.7	Evolución del número de mensajes de control enviados respecto a la densidad para <i>AODV</i> .	30
Figura 3.1	Pila de Comunicaciones para WSN	37
Figura 3.2	Ejemplo de sistema RFID	38
Figura 3.3	Bandas de Frecuencias y tipos de modulaciones en IEEE 802.15.4	41
Figura 3.4	Trama 802.15.4	41
Figura 3.5	Posibles Topologías basadas en IEEE 802.15.4	42
Figura 3.6	Ejemplo de envío de mensajes RREQ en <i>AODV</i> .	45
Figura 3.7	[Ejemplo de envío de mensajes de respuesta RREP en <i>AODV</i> .]	45
Figura 4.1	Pila de Comunicaciones del protocolo <i>DARP</i>	54
Figura 4.2	Ejemplo de red PAN basada en <i>DARAL</i> .	55
Figura 4.3	Ejemplo A de enrutamiento en <i>DARAL</i> .	60
Figura 4.4	Ejemplo B de enrutamiento en <i>DARAL</i> .	60
Figura 4.5	Ejemplo C de enrutamiento en <i>DARAL</i> .	61
Figura 4.6	Ejemplo de funcionamiento de <i>DRSP</i> .	61
Figura 4.7	Representación del algoritmo <i>DRSP</i> en UML.	62
Figura 4.8	Relación entre VCs y sub-redes.	63
Figura 5.1	Estructura modular de un nodo <i>DARP</i> en OM-NeT++.	71
Figura 5.2	Número de nodos conectados para diferentes niveles de $T_{H_{baselevel}}$.	75
Figura 5.3	Ánálisis del Tiempo de Convergencia para diferentes niveles de $T_{H_{baselevel}}$.	75
Figura 5.4	Ánálisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de $T_{H_{baselevel}}$.	76
Figura 5.5	Ánálisis del Número de mensajes de control enviados durante todo la simulación para diferentes niveles de $T_{H_{baselevel}}$.	76
Figura 5.6	Ánálisis del Consumo de energía medio por nodo para diferentes niveles de $T_{H_{baselevel}}$.	77
Figura 5.7	Ánálisis del Número de nodos conectados para diferentes niveles de $T_{H_{role}}$.	78

- Figura 5.8 Análisis del Tiempo de Convergencia para diferentes niveles de TH_{role} . 78
- Figura 5.9 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de TH_{role} . 79
- Figura 5.10 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de TH_{role} . 79
- Figura 5.11 Análisis del Consumo de energía medio por nodo para diferentes niveles de TH_{role} . 80
- Figura 5.12 Número de nodos conectados para diferentes niveles de L_{nodes} . 81
- Figura 5.13 Análisis del Tiempo de Convergencia para diferentes niveles de L_{nodes} . 81
- Figura 5.14 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de L_{nodes} . 82
- Figura 5.15 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de L_{nodes} . 82
- Figura 5.16 Análisis del Consumo de energía medio por nodo para diferentes niveles de L_{nodes} . 83
- Figura 5.17 Número de nodos conectados para diferentes niveles de DARAL_MAX_RETRIES. 84
- Figura 5.18 Análisis del Tiempo de Convergencia para diferentes niveles de DARAL_MAX_RETRIES. 84
- Figura 5.19 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de DARAL_MAX_RETRIES. 85
- Figura 5.20 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de DARAL_MAX_RETRIES. 85
- Figura 5.21 Análisis del Consumo de energía medio por nodo para diferentes niveles de DARAL_MAX_RETRIES. 86
- Figura 5.22 Número de nodos conectados para diferentes niveles de T_{link} . 86
- Figura 5.23 Análisis del Tiempo de Convergencia para diferentes niveles de T_{link} . 87
- Figura 5.24 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{link} . 87
- Figura 5.25 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{link} . 88
- Figura 5.26 Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{link} . 88
- Figura 5.27 Número de nodos conectados para diferentes niveles de T_{alive} . 89
- Figura 5.28 Análisis del Tiempo de Convergencia para diferentes niveles de T_{alive} . 89

- Figura 5.29 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{alive} . 90
- Figura 5.30 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{alive} . 90
- Figura 5.31 Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{alive} . 91
- Figura 5.32 Número de nodos conectados para diferentes niveles de T_{down} . 92
- Figura 5.33 Análisis del Tiempo de Convergencia para diferentes niveles de T_{down} . 92
- Figura 5.34 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{down} . 93
- Figura 5.35 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{down} . 93
- Figura 5.36 Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{down} . 94
- Figura 5.37 Número de nodos conectados para diferentes niveles de $T_{reconnect}$. 94
- Figura 5.38 Análisis del Tiempo de Convergencia para diferentes niveles de $T_{reconnect}$. 95
- Figura 5.39 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de $T_{reconnect}$. 96
- Figura 5.40 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de $T_{reconnect}$. 96
- Figura 5.41 Análisis del Consumo de energía medio por nodo para diferentes niveles de $T_{reconnect}$. 97
- Figura 5.42 Número de nodos conectados para diferentes niveles de T_{ack} . 97
- Figura 5.43 Análisis del Tiempo de Convergencia para diferentes niveles de T_{ack} . 98
- Figura 5.44 Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{ack} . 98
- Figura 5.45 Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{ack} . 99
- Figura 5.46 Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{ack} . 99
- Figura 6.1 Tiempo de Convergencia medio en escenarios pequeños de baja densidad. 109
- Figura 6.2 Tiempo de Convergencia medio en escenarios pequeños de media densidad. 109
- Figura 6.3 Tiempo de Convergencia medio en escenarios pequeños de alta densidad. 110
- Figura 6.4 Tiempo de Convergencia medio en escenarios medianos de baja densidad. 111

- Figura 6.5 Tiempo de Convergencia medio en escenarios medianos de media densidad. 111
- Figura 6.6 Tiempo de Convergencia medio en escenarios medianos de alta densidad. 111
- Figura 6.7 Tiempo de Convergencia medio en escenarios grandes de baja densidad. 113
- Figura 6.8 Tiempo de Convergencia medio en escenarios grandes de baja densidad. 113
- Figura 6.9 Tiempo de Convergencia medio en escenarios grandes de baja densidad. 113
- Figura 6.10 Número medio de mensajes de control enviado para escenarios pequeños de baja densidad, considerando AODV. 116
- Figura 6.11 Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios pequeños de baja densidad. 116
- Figura 6.12 Número medio de mensajes de control enviado para escenarios pequeños de media densidad, considerando AODV. 116
- Figura 6.13 Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios pequeños de media densidad. 117
- Figura 6.14 Número medio de mensajes de control enviado para escenarios pequeños de alta densidad, considerando AODV. 117
- Figura 6.15 Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios pequeños de alta densidad. 117
- Figura 6.16 Número medio de mensajes de control enviado para escenarios medianos de baja densidad, considerando AODV. 118
- Figura 6.17 Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios medianos de baja densidad. 119
- Figura 6.18 Número medio de mensajes de control enviado para escenarios medianos de media densidad, considerando AODV. 119
- Figura 6.19 Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios medianos de media densidad. 119
- Figura 6.20 Número medio de mensajes de control enviado para escenarios medianos de alta densidad, considerando AODV. 120
- Figura 6.21 Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios medianos de alta densidad. 120
- Figura 6.22 Número medio de mensajes de control enviado para escenarios grandes de baja densidad, considerando AODV. 121
- Figura 6.23 Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios grandes de baja densidad. 121

- Figura 6.24 Número medio de mensajes de control enviado para escenarios grandes de media densidad, considerando AODV. [122](#)
- Figura 6.25 Análisis detallado del número medio de mensajes de control enviado por *DARP* en escenarios grandes de media densidad. [122](#)
- Figura 6.26 Número medio de mensajes de control enviado para escenarios grandes de alta densidad, considerando AODV. [122](#)
- Figura 6.27 Análisis detallado del número medio de mensajes de control enviado por *DARP* en escenarios grandes de alta densidad [123](#)
- Figura 7.1 Comparación del tiempo de convergencia en escenarios muy grandes para los diferentes protocolos. [126](#)
- Figura 7.2 Comparación detallada entre *O-RPL* y *DARP* del tiempo de convergencia en escenarios muy grandes. [126](#)
- Figura 7.3 Comparación del número de mensajes de control durante la fase de conexión en escenarios muy grandes para los diferentes protocolos. [127](#)
- Figura 7.4 Análisis del número de mensajes de control durante la fase de conexión en escenarios muy grandes de *DARP*. [127](#)
- Figura 7.5 Análisis del Consumo de energía medio por nodo para escenarios muy grandes. [128](#)
- Figura 7.6 Comparación del tiempo de convergencia en escenarios enormes para los diferentes protocolos. [128](#)
- Figura 7.7 Comparación del número de mensajes de control durante la fase de conexión en escenarios enormes para los diferentes protocolos. [129](#)
- Figura 7.8 Comparación del número de mensajes de control durante la fase de conexión en escenarios enormes para los diferentes protocolos. [129](#)
- Figura 7.9 Análisis del Consumo de energía medio por nodo para escenarios enormes. [130](#)
- Figura 7.10 Comparación de los consumos energéticos de los protocolos en la fase de conexión. [130](#)
- Figura 7.11 Análisis del consumo energético de *DARP* detallado. [131](#)
- Figura 7.12 Comparativa del aprovechamiento del canal de los diferentes protocolos. [132](#)
- Figura 7.13 Análisis del aprovechamiento del canal de *DARP* detallado. [132](#)
- Figura 7.14 Comparativa del número de colisiones de los protocolos. [133](#)
- Figura 7.15 Comparativa del *tiempo de convergencia* en un escenario dinámico. [134](#)
- Figura 7.16 Comparativa del *número de mensajes de control enviados durante la fase de conexión* en un escenario dinámico. [134](#)

Figura 7.17	Análisis del Consumo de energía medio por nodo para escenarios dinámicos. 135
Figura B.1	Importing a Project into workspace. 178
Figura B.2	TicToc Example Network. 179
Figura B.3	Real Simple Node. 180
Figura B.4	Simple Network Topology. 180
Figura B.5	New Simple Module Dialog Windows. 181
Figura B.6	New Compound Module Dialog Window. 185
Figura C.1	DARP State Machine. 193
Figura C.2	Message flow in an EN Association Operation. 194
Figura C.3	Message flow in a VC Association Operation. 194
Figura C.4	Message flow in a successful Status Control Operation. 195
Figura C.5	Message flow in an unsuccessful Status Control Operation. 195
Figura C.6	Message flow in a successful Temporal Disassociation Operation. 196
Figura C.7	Message flow in a successful Full Disassociation Operation. 197
Figura C.8	Message flow in a Data Sending Operation. 197

LISTA DE TABLAS

Tabla 3.1	Resumen general de frecuencias y tipos de modulación. 41
Tabla 4.1	Cabecera de Control en DARP. 56
Tabla 5.1	Configuración del medio de simulación y de la capa PHY del estándar IEEE 802.15.4. 72
Tabla 5.2	Configuración de la capa MAC del estándar IEEE 802.15.4. 73
Tabla 5.3	Configuración original adoptada para la calibración de DARP. 74
Tabla 5.4	Configuración óptima de DARP. 100
Tabla 6.1	Configuración de los parámetros de la ecuación 6.2. 105
Tabla 6.2	Escenarios de simulación. 106
Tabla 6.3	Configuración de los parámetros del módulo <i>Traffic Generator</i> . 106
Tabla 6.4	Tiempo de Convergencia medio en escenarios pequeños. 109
Tabla 6.5	Tiempo de Convergencia medio en escenarios medianos. 110
Tabla 6.6	Tiempo de Convergencia medio en escenarios grandes. 112
Tabla 6.7	Número de mensajes de control medio en escenarios pequeños. 115

Tabla 6.8	Número de mensajes de control medio en escenarios medianos. 118
Tabla 6.9	Número de mensajes de control medio en escenarios grandes. 121
Tabla 8.1	Características <i>DARP</i> 141
Tabla 10.1	<i>DARP Features</i> 155
Tabla A.1	Comparativa de Plataformas de Simulación para WSN: Resumen de Características 175
Tabla C.1	<i>DARP control header Description.</i> 198
Tabla C.2	Available packets type for a node association. 199
Tabla C.3	Available packets type for a network control status. 199
Tabla C.4	Available packets type for a node disassociation. 199
Tabla C.5	Available packets type for a data sending. 199
Tabla C.6	Available Routing types in <i>DARP</i> 200

ACRÓNIMOS

6LowPAN IPv6-Based Low Power Wireless Personal Area Networks

ACK Acknowledgement

AES Advanced Encryption Standard

API Application Program Interface

APL Application Layer

ARPA Advanced Research Projects Agency

ARPANET Advanced Research Projects Agency Network

ASIC Application-Specific Integrated Circuit

AODV Ad-Hoc On-Demand Distance Vector

CSMA Carrier Sense Multiple Access

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CSMA/CD Carrier Sense Multiple Access with Collision Detection

DARAL Dynamic and Adaptive Routing Algorithm

DARP Dynamic and Adaptive Routing Protocol

DARPA Defense Advanced Research Projects Agency

DPM Dynamic Power Management

DRSP Dynamic Role Selection Process

DVS Dynamic Voltage Scaling

DSSS Direct Sequence Spread Spectrum

ED Energy Detection

FFD Full Function Device

GTS Guaranteed Time Slot

IoT Internet of Things

ISM	Industrial Scientifical Medical
LAN	Local Area Network
LQI	Link Quality Indicator
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access Control
MAN	Metropolitan Area Network
NCM	Network Access Manager
NTP	Network Time Protocol
NWK	Network Layer
QoS	Quality of Service
PAN	Personal Area Network
PHY	Physical Layer
RF	Radio Frequency
RFD	Reduced Function Device
RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
RSSI	Received Signal Strength Indicator
SoC	System-on-Chip
TDMA	Time Division Multiple Access
vID	DARAL Virtual Sub-Network Identificator
WAN	Wide Area Network
WISDEN	Wireless Sensor Network for Structural-response Data Acquisition
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Networks

Parte I

LARGE SUMMARY (ENGLISH VERSION)

1

THESIS SUMMARY

CONTENTS

1.1	Short Introduction	4
1.2	Short Literature Review	5
1.3	Dynamic and Adaptive Routing Protocol (DARP)	7
1.3.1	Header and message types	8
1.3.2	DARAL Description	10
1.4	Experiment Summary	13
1.4.1	DARP Behavior analysis	13
1.4.2	Comparative study of protocol performance	13
1.4.3	Study of DARP additional features	14

The present chapter presents a short version of this Thesis in English and it is organized just as the Thesis, based on sections. The first one presents a short introduction, the second one, a short literature revision, the third one contains this Thesis proposal, and the fourth one, the experiments carried out.

The discussion of the results of the experiments and the conclusions can be found in Chapters 10 and 11, respectively.

1.1 SHORT INTRODUCTION

The wireless communication area is wide but there are some well-known standards which mark its development. These standards are *IEEE 802.11* [43], *IEEE 802.15.1* [44], *IEEE 802.15.3* [45], and *IEEE 802.15.4* [46]. These technologies are a major area of research, mainly *IEEE 802.11* and *IEEE 802.15.1*, but considering the Smart Cities as scenario of this Thesis, *IEEE 802.15.4* is the most suitable standard, because it is oriented towards wireless sensor networks (WSN) and low-rate communications. This standard regulates only the access to the medium by definition of the physical (PHY) and medium access (MAC) layers. There are different routing protocols, due to the lack of an advanced routing method in the standard. These routing protocols are needed for complex scenarios like Smart Cities, which are one of the most complex possible scenarios. Moreover, other features more than topology are required, and routing algorithms are fully qualified to provide them. Reliability, dynamism, energy efficiency, channel exploitation, etc., are some necessities for modern wireless communications that can only be covered by advanced routing protocols.

RPL-6LoWPAN [87] or *AODV* [69] are some of these well-known *IEEE 802.15.4*-based routing algorithms. Each one possesses some interesting features, but they also present disadvantages when exposed to a scenario like the Smart City. This Thesis focuses on the analysis of *convergence time* and *number of messages sent during the forming phase* due to the necessity of a high dynamism, and also due to energy and throughput requirements (a lower overhead requires a lower consumption and allows a better channel exploitation).

The development of a Smart City infrastructure through embedded electronics and wireless communications leads to the question of the number of nodes for a Smart City. Well, that is an uncertain question; it depends on the service offer and on the government. But considering the network communications, number is not as important as density. For wireless communications, density is a challenge, because in high-density environments, the available spectrum tends to collapse, leading the network to be unstable. So, analyzing the most used applications (e.g. city lights, water and traffic monitoring, etc.) supported by the Smart City architecture, their density rarely scales over 1 node per m². That cannot be considered as a high-density scenario. Thus, for this Thesis, the required node density of a future coming Smart City remains open, but in our case, the Smart City architecture is considered as a low- or medium-density scenario.

DARP, *DARAL*, and *DRSP* are the proposals of this Thesis. They suggest a routing algorithm based on a dynamically-allocated hierarchical clustering, which uses the quality of the links as reference parameter, minimizing the control message overhead, convergence time and energy consumption during the forming phase of the network. To

achieve energy-efficiency, the nodes minimize their active time transferring the routing tasks to some dynamically-selected nodes. Additionally, DARAL organizes the nodes in virtual sub-networks, containing the traffic by zones and also increasing efficiency in terms of convergence time.

This work experimentation focuses on a fine tune of this work proposal, DARP, deeply analyzing the performance of each configuration parameter and looking for a convergent configuration for large-size scenarios under low or medium density conditions. It also addresses the problems of convergence time [41] through different scenarios, together with the *number of messages sent during the forming phase*, based on a comparison between other well-known routing algorithms. Finally, it studies different additional features as channel exploitation, energy consumption, reduction of collisions, and response against dynamical events.

1.2 SHORT LITERATURE REVIEW

Routing algorithms are an important part in Smart City and IoT projects, due to the critical function they play. Although there are different routing methodologies like multicast [23] [76], mesh [83], or graph-based [64], routing algorithms based on clustering [35] [77] are designed to improve different parameters such as QoS, energy consumption, traffic reduction, or range maximization [88]. Thus, our proposal, DARAL, is based on clustering techniques. Within clustering-based routing algorithms, there are mainly hierarchical cluster tree algorithms, but there are other alternatives like the spanning tree proposed by Saravanan M. et al in [59].

Usually, the implementation of a hierarchical clustering scheme is based on the definition of two different roles or functionalities for the nodes of the network, the cluster-heads and non-head nodes. A cluster is formed by a cluster-head and a set of non-head nodes, where the nodes of the cluster communicate between them (sensor-to-sensor) and mainly with their cluster-head, which also leads the inter-cluster communications.

As the role selection mechanism is a fundamental aspect in hierarchical cluster-based routing algorithms, there are different cluster-head election schemes that consider a wide range of parameters such as location, residual energy, or LQI. For example, M. Jiasong et al. [51] described an adaptive routing optimization based on the energy balancing algorithm for hierarchical networks in ZigBee, where they limited the number of hops depending on the battery available, limiting the range of a certain node too. Another possibility for the cluster-head election is the one proposed by the MultihopLQI routing algorithm [34], where a tree of multiple hops is dynamically built for routing tasks by the analysis of the impact of an LQI threshold in the routing formation,

considering *MinLQI* and *MaxLQI* values. However, this cluster-head election scheme does not take in account some important parameters such as QoS, convergence time, or control overhead, focusing instead on the analysis of path length and network lifetime. This is a common pitfall in most of the common cluster-head election schemes. As long as they need to analyze different parameters to produce a measurable result, they suffer from two major drawbacks: an increment of the convergence time and also a message overhead, making them unsuitable for Smart City or IoT projects.

There is also a wide diversity in hierarchical cluster-based routing algorithms, mainly due to different scenario characteristics. Thus, each routing algorithm usually focuses on a specific set of parameters related to a concrete scenario, looking for their improvement. For example, J. H. Chang and L. Tassiulas [50] explored the maximum lifetime of the routing in WSN through the reformulation of the energy efficiency problem. They improved the total energy efficiency of the system using the network lifetime increment instead of minimizing the total energy consumed to reach the destination. Another approach is presented by C. F. Huang and Y. C. Tseng in [29], where the problem of the coverage for WSN is analyzed, assuming that all the nodes are under a certain coverage grade. But focusing on Smart City/IoT applications, and as K. Machado et al. commented in their article [61], routing solutions for these scenarios should consider different traffic patterns, such as one-to-many, many-to-one, and many-to-many, due to the dynamical nature of Smart City/IoT applications. This requirement has favored that most of the Smart City/IoT proposals are based on two well-known routing algorithms: Ad-Hoc On demand Distance Vector (AODV) and IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), mainly because both present a low-complexity and a good performance in these scenarios.

AODV is a routing algorithm that discovers the routes using Route Request (RREQ) and Route Reply (RREP) messages, ensuring that there are no loops and, at the same time, trying to find the shortest possible route. A certain origin node broadcasts an RREQ to their neighbor nodes, looking for a certain destination node. These neighbors look in their routing tables for the destination node. If there is not any match, these neighbors also broadcast the original RREQ to another nodes, and the process continues until a node finds the destination node in its routing table. But if there is a match, the node with the match resends the original RREQ to the destination node and once the destination node processes the RREQ, it answers with an RREP that is routed towards the origin node, setting the route and closing the route discovery process. As this routing method is fired on-demand and is based on a minimal number of hops, it does not ensure energy-efficiency, convergence time, or control overhead efficiency. Moreover, *AODV* only stores one route for a destination node, resulting in an additional use of resources if there is any problem with that route, due to the necessity of discovering a new one. The lack of a mechanism to ensure energy, convergence time efficiency, and a minimal control overhead results

in a waste of resources for constrained scenarios with low-/medium-densities, due to the instability of links and routes.

On the other hand, *RPL* routes are based on Destination Oriented Directed Acyclic Graphs (DODAGs). A *DODAG* is a graph based on nodes and links forming the path to the network root, which is basically the sink or network coordinator. Thus, *RPL* has been designed and optimized for the transmission of data from sensor nodes towards the root node, following a many-to-one scheme. For the construction and maintenance of a *DODAG*, *RPL* nodes locally multicast *DODAG* Information Object (DIO) messages pseudo-periodically, which contain information that allows a node to discover an existing *DODAG*, together with its configuration parameters. Once a node receives a DIO message from a neighbor, it will be able to join an existing *DODAG*. However, if a node is trying to connect to the network and does not receive any DIO message, it can send a *DODAG* Information Solicitation (DIS) message to request the immediate transmission of DIO messages. Once the DIS message is sent, additional DIS messages may be sent until a DIO message is received in response. But if a DIO message is not received after a certain time, the node may decide to become the root of a new *DODAG*. *RPL* also supports downward routes by using Destination Advertisement Object (DAO) messages, which are generated and sent upwards by non-root nodes to announce themselves as possible destination nodes. The lack of energy and convergence time efficiency, jointly with the control overhead, results, as in the *AODV* case, in a waste of resources. This problem was analyzed by H. Kermajani and C. Gomez in [41], where an optimized *RPL* *DODAG*-route forming method (O-RPL) was proposed to reduce the convergence time for dense scenarios. [41] also shows an interesting test-bed to analyze both the convergence time and control overhead in terms of node degree (density).

As described above, existing routing algorithms present some drawbacks related to the lack of an integrated QoS, an efficient convergence time, and a minimal control overhead in the network formation stage for Smart City/IoT scenarios, considering these scenarios as low-/medium-density environments. With this goal in mind, we propose a new routing algorithm based on the LQI, which is hierarchically organized in sub-networks, which improves the convergence time and the message control overhead for Smart City/IoT scenarios with a small-/medium-density where coverage is a critical parameter.

1.3 DYNAMIC AND ADAPTIVE ROUTING PROTOCOL (DARP)

The novelty of this Thesis is the development of a non-beaconing routing algorithm based on a dynamical clustering, which is based on the link quality between nodes. *DARP* [39] is the routing protocol, but it is based on a routing algorithm called *DARAL*, which is also a contribution of this Thesis. It is designed for large-size scenarios under

low-/medium-density conditions, like Smart Cities, focusing on infrastructure support. The clusters are configured like virtual sub-networks and they can work autonomously and in parallel.

Before entering in a fine grain description, it is necessary to define some general concepts about *DARP* and *DARAL*, like the sub-network concept and the different types of role. The sub-network concept comes from the clustering techniques and the idea of Smart Cities organization. In Smart City approaches, nodes usually communicate in the neighborhood. Thus, if clustering is applied, traffic can be contained in a determined area, allowing the reduction of interference with other nodes and also minimizing the necessity for looking for a node among the complete network topology. In order to improve the clustering concept, the virtual sub-network concept extends this idea, applying it in routing tasks. Each virtual sub-network is identified with a virtual identification (vID), which is used to route messages among the network tree. Figure 4.2 in Chapter 4 shows a global Personal Area Network (PAN) with its own identifier, but internally sub-divided into 3 different sub-networks. The first one is where the network root is located, while the other two group different nodes.

Another general concept about *DARAL* is the node role, considering that every node should be similar in terms of hardware. *DARAL* runs a dynamical role selection process (*DRSP*) at the start-up of every node, which allows us to select the role to be played by a node in the network. Figure 4.2 shows the two available roles in *DARAL*, end node (EN) and virtual coordinator (VC). ENs are nodes that only communicate with the VC of its sub-network. VCs have the same functionality that ENs, but also manage virtual sub-networks and store routing tables.

1.3.1 Header and message types

First of all, it is necessary to describe how *DARAL* works, presenting the different message types and describing them, whereas describing the control header of the algorithm. As it is well known, every layer in a network communication stack adds a control header to the packet. Annex C shows the *DARAL* header and Table 4.1 in Chapter 4 too, right after the IEEE 802.15.4 MAC header in a network layer packet. It sums up 27 octets, distributed as follows: operation code, which identifies the different control messages; packet length, used to read the payload length; routing type, which indicates the type of routing; a hop limit, not used at the moment; checksum and message id, used to avoid errors and duplication; source and destination virtual network id, used to primarily address the messages; and source and destination address used to address the messages in a last step.

Operation code defines the type of control message used by *DARAL*, which counts with 15 different messages, used for connection, dele-

tion, and routing tasks. Below, a short description of these messages is included:

- ASSOCIATION_REQ: Each node sends this message as broadcast, to indicate that it requests a connection.
- ASSOCIATION REP: A VC answers the *ASSOCIATION_REQ* message with this one, indicating that it can accept the node.
- ASSOCIATION REP ACK: If a node configures itself as EN, it sends this message to its VC.
- ASSOCIATION PAN ID REQ: If a node configures itself as VC, it sends this message to its VC (also known as parent), requiring a new vID. This message contains the information that identifies the node.
- ASSOCIATION PAN ID REQ ACK: This message is the answer from the network root to the *ASSOCIATION PAN ID REQ* and it contains the updated information (the new vID) for the node.
- ASSOCIATION PAN ID ASSIGN: The VC (or parent) assigns the new vID to the new VC.
- ASSOCIATION PAN ID ASSIGN ACK: This message is an answer to the *ASSOCIATION PAN ID ASSIGN* message and it confirms that the new sub-network is currently operative.
- ASSOCIATION INFORM: A certain VC sends this message to the network root when a new node (VC or EN) is connected in its sub-network.
- ASSOCIATION INFORM ACK: It is the network root acknowledgment to an *ASSOCIATION INFORM* message.
- KEEP ALIVE REQ: This message is the request of a VC for every node in its sub-network, in order to test the nodes' connectivity
- KEEP ALIVE REQ ACK: A specific answer for a *KEEP_ALIVE_REQ* message.
- PURGE REQ: A VC sends this message to the network root when it detects that a node in its sub-network is down.
- PURGE REQ ACK: This message is generated by the network root as a response to a *PURGE_REQ* and it informs each node until the destination that a certain node is not available anymore.
- DATA: It sends a data frame, it can contain other protocol frames (e.g. 6LoWPAN-IPv6).
- DATA ACK: It confirms the reception of a *DATA* frame.

The routing type is a particular element in the control header and *DARAL* uses 3 different types:

- Gateway: It represents a message going up in the network tree, using the link with the upper VC.
- Forwarding: It means that a message goes down in the network tree until it reaches its destination sub-network.
- Parsing: It is the last step for a message transmission. This type is used when a message reaches the destination sub-network and it is going to be delivered to the destination node, using from now on the node address to route.

Figures 4.3, 4.4, and 4.5 show these kinds of messages graphically, showing the nodes represented as a triangle the VCs, which manage a virtual sub-network. Cases 4.3 and 4.4 show a communication between different clusters. In both cases, the node EN_{2F} generates a packet with destination EN_{3D} in 4.3 and EN_{1B} in 4.4. As long as EN_{2F} is an EN without routing tables, it needs to send the message with the routing type *Gateway* to its VC. For this example, the VC from the sub-network 2 does not know the destination node, so, it also routes the message to its VC (in this case, the *Network Root*) using also the routing type *Gateway*. As the *Network Root* knows the complete routing tree, it can find a route to forward the packet using the routing type *Forwarding* 4.3, or find the destination node using the routing type *Parsing* 4.4.

Another example is shown in Figure 4.5 and it shows an intra-cluster communication. As long as a node sends a message to another node in the same sub-network, the VC can route it directly without external intervention. I.e., the EN_{2F} sends a message to EN_{2C} . The message goes firstly to its VC, which finds the node in the sub-network. Then, the VC sends the message to EN_{2C} with the routing type *Parsing*.

1.3.2 *DARAL Description*

Under this section, the different details that form *DARAL* are explained and, to begin with them, it is better to start with the sub-network generation and the dynamical role selection process, *DRSP*.

Figure 4.6 in Chapter 4 shows the first process in *DARAL*, where a node $EN-A$ tries to connect to the network. First of all, $EN-A$ broadcasts an *ASSOCIATION_REQ* message. In Figure 4.6, the VCs from sub-networks 1 (VC-1), 2 (VC-2), and 3 (VC-3) are within its communication range, so they are receiving the *ASSOCIATION_REQ* message. After checking the message, they proceed to answer with an *ASSOCIATION REP* message. $EN-A$ receives those *ASSOCIATION REP* from VC-1, VC-2, and VC-3 and fires the *DRSP*, in order to choose the best sub-network to join, in terms of link quality indicator (LQI). Once

a sub-network is chosen and *EN-A* selects its role, *EN-A* sends a message to the corresponding VC, in Figure 4.6 VC-2, and from then on, it belongs to that sub-network, in the example, to sub-network 2.

DRSP: Dynamic Role Selection Process

As Figure 4.6 in Chapter 4 shows, a node selects the best link to connect with and, once it is chosen, the node evaluates the LQI to adopt one of two possible roles in the network, EN or VC. EN is basically limited to receiving and sending messages, reducing as much as possible its active time, thus saving energy. On the other hand, there is the VC. VCs are the key elements in *DARAL*, being used to widen the network coverage range and to route messages among sub-networks. VCs play both roles at the same time, as can be seen in Figure 4.8, as EN for the network they belong to and as VC to the sub-network that they own. In Figure 4.8, every VC shows its two vIDs, the *original vID* from the network where they belong and the *additional vID* from the sub-network that they manage.

DRSP is based on 2 user-defined parameters: $TH_{baselevel}$ and TH_{role} . $TH_{baselevel}$ is basically a threshold granting a minimum QoS for the link with the VC. Any link with a LQI below $TH_{baselevel}$ is not established. On the other hand, TH_{role} defines the threshold for the role. If the LQI is above TH_{role} , a node configures itself as EN, due to the good QoS of the link. But if the LQI is below TH_{role} (and above $TH_{baselevel}$), a node adopts the role of VC, considering that from that point onwards, the QoS will worsen even more. *DARAL* is based on the concept that a high LQI (above TH_{role}) means a better QoS and a nearest position, allowing nodes to be grouped and a traffic concentration, hence the adoption of an EN role. A worse LQI (below TH_{role}) represents a point where the QoS is not good enough, and if the distance continues to increase, that QoS will worsen. To reduce the impact of the deterioration in the QoS, a VC acts as a concentrator, which allows the network coverage to be expanded.

Figure 4.7 in Chapter 4 shows how *DRSP* is carried out. The first 5 lines initialize some variables. Then, $TH_{baselevel}$ is evaluated, if the LQI is below $TH_{baselevel}$, the process finishes. Otherwise, if it is above the $TH_{baselevel}$ value, it continues evaluating TH_{role} value. If the LQI is above TH_{role} , (lines 10-13), then the role is set to EN, the node status changes to connected and the node is marked as non VC. If the LQI is below TH_{role} (and above $TH_{baselevel}$), (lines 15-17), then the role is set as VC, the status is set to *awaiting* and the node is marked as VC. The *awaiting* status means that a VC is connected and can receive messages, but at the moment, it cannot route packets. This status changes to connected when that VC receives a vID for its new sub-network.

Finally the algorithm evaluates the node role adopted, (lines 21-25), to send an *ASSOCIATION_PAN_ID_REQ*, if the role is VC, or an *ASSO-*

CIATION REP ACK, if the role is an EN. At the end, different timers, like packet delivery time or connection status time are set.

Sub-network concept

DARAL routing is based on the use of sub-networks. Every sub-network possesses a sub-network ID (vID), which unequivocally identifies the sub-network in the whole network. Each sub-network groups a certain number of nodes, being that number of nodes user-limited by the parameter L_{nodes} , and hence, the load balance of *DARAL*. If the limit L_{nodes} is reached by a VC, that VC does not send any *ASSOCIATION REP* anymore, at least, as long as the number of connected nodes remains unchanged.

VCs manage the sub-networks, possessing two different routing tables, one for the local nodes in the sub-network and another for routing tasks. VCs know only the routes below their position, for example in Figure 4.8, the VC from sub-network 3 knows the nodes in its sub-network and the nodes in sub-network 4. Also, in Figure 4.8, the network root, (VC in sub-network 1) knows the complete routing table. The particularity of *DARAL* is that a message being routed between 2 different sub-networks is not completely processed. A VC in the route only needs to analyze the *Destination Virtual Network ID* and look for it in the routing table to know the next step in the route. Only messages that already reached their destination sub-network are completely processed. Then, the VC of that sub-network looks for the destination address in the local node table.

Sub-network IDs (vID) are only generated by the network root. Once a vID is generated, all the VCs in the route update their routing tables, adding a new entry to them. On the other hand, if a new EN connects to a sub-network, the VC of that sub-network informs every node above him, sending an *ASSOCIATION INFORM* message. Every VC that receives an *ASSOCIATION INFORM* updates its own routing table.

Due to this routing mechanism, it is necessary to check periodically the status of the different sub-networks, and also the connectivity of the ENs. This process is fired by the user-defined timer T_{alive} . When T_{alive} expires, a VC sends a *KEEP_ALIVE_REQ* message to its sub-network, checking the connectivity of every node. If a node does not respond, another user-defined timer T_{down} is fired. When T_{down} expires, a purging request is generated by the VC, asking the VCs above it to remove that node from the routing tables.

If the connectivity problem is detected by ENs and not by a VC, those ENs await until the user-defined timer $T_{reconnect}$ expires. If the timer expires and there is no answer from the VC, those ENs restart and begin with *DARAL* and the *DRSP*. After restarting, the nodes reconnect the network and the routing tables are updated with the new

information. If a keep alive process starts during the disconnection, the previously described purge process is fired.

1.4 EXPERIMENT SUMMARY

The simulations have been performed with *OMNeT++* [84], a well-known C++ discrete event simulator in the research community. As OMNeT++ is not focused on wireless networks, it is necessary the use of a framework, as *inetmanet* [22]. *DARP* was implemented for *OMNeT++* and it is publicly available in [38]. The network is configured following our Smart City scenario proposal, using randomly distributed static nodes. As *DARP* is an *IEEE 802.15.4*-based routing protocol, the communication band used is the ISM band of 2.4GHz using the beaconless mode.

1.4.1 *DARP Behavior analysis*

A large description of the configuration parameters is included in Chapter 5, defining in Tables 5.1 and 5.2 the configuration for the *MAC* and *PHY IEEE 802.15.4* layers.

A convergence study based on different iterations has been carried out to tune the different *DARP* configuration parameters T_{link} , T_{alive} , T_{down} , $T_{\text{reconnect}}$, L_{nodes} , $TH_{\text{baselevel}}$, and TH_{role} .

This chapter focuses on the analysis of this convergence study, summarizing the results with Table 5.4 as optimum configuration summary. The results are discussed in Chapter 10.

1.4.2 *Comparative study of protocol performance*

This chapter focuses on the different possibilities of a Smart City scenario. Thus, it is necessary to define different workloads in terms of traffic. There are different possibilities for that, and all of them involve the area size and the number of nodes, so the modification of these parameters makes possible to work with different network densities using the network degree as reference. For this work, 3 different node degrees (ND) are considered: 5, 10, and 15. The smallest scenario is built in a 145×145 m and the largest in a 500×500 m area. The use of this wide area and node degree combinations allows the simulation from sparse networks to highly dense networks. As a result, the simulation covers several network densities and sizes, resulting in a vast results comparison. The different combinations between network size, grade, and number of nodes are shown in Table 6.2 in Chapter 6. For

each experiment configuration, 10 randomly generated scenarios with different random seeds were evaluated.

The analysis is focused on the protocol dynamism, understanding dynamism as the combination of *convergence time* and *number of messages sent during the forming phase*. The results are discussed in Chapter 10.

1.4.3 Study of DARP additional features

This chapter studies some different additional features within DARP. These additional features are observed during the design of the protocol and are commented in Chapter 4. In order to validate them, this chapter analyzes them.

Different studies based on large-size scenarios under medium-density conditions are carried out, discussing the results in Chapter 10.

Parte II

INTRODUCCIÓN

2

INTRODUCCIÓN

ÍNDICE

2.1	Motivación	18
2.2	Campos de aplicación	19
2.2.1	Sistemas agrarios inteligentes	20
2.2.2	Ciudades inteligentes	20
2.2.3	Internet de las Cosas	21
2.3	Marco de trabajo	23
2.3.1	Caracterización de las comunicaciones en WSN	23
2.3.2	Densidad de nodos en WSN	24
2.3.3	Tiempo de convergencia en WSN	25
2.3.4	Número de Mensajes de control en WSN	26
2.4	Planteamiento	26
2.4.1	Problemática	26
2.4.2	Densidad de nodos asociada a los grandes escenarios	27
2.4.3	Objetivos científicos de la Tesis	30
2.5	Estructura de la Tesis	31

La Ciencia es una rama del conocimiento humano que se ha construido desde el albor de los tiempos. Desde los primeros hombres hasta la actualidad, las ideas y su puesta en práctica han permitido la evolución de nuestra especie. Los grandes descubrimientos y avances en diferentes áreas, nos permitieron utilizar un lenguaje común en la ciencia, las matemáticas. Mediante su uso, la humanidad pudo especializar la propia ciencia, diversificándola entre ciencia teórica y ciencia aplicada. Esta Tesis Doctoral se ha querido desarrollar como un trabajo de investigación científica aplicada.

Para poder desarrollar la presente Tesis Doctoral se ha partido de una primera motivación de tipo social, para la que diversos investigadores han propuesto algunas soluciones. Sin embargo, estas soluciones también presentan ciertas limitaciones. La búsqueda de respuestas a los interrogantes que surgen de estas limitaciones ha sido la principal motivación técnica que ha alentado este trabajo. Así pues, esta Tesis ha pretendido dar solución a un problema científico-técnico concreto dentro de un marco de aplicación y utilidad social.

La memoria de la presente Tesis Doctoral ha sido dividida en diferentes partes, que a su vez están subdivididas en capítulos. Esta parte se inicia describiendo una problemática que ha resultado desafiante para la comunidad científica. El desarrollo de arquitecturas de comunicaciones en el marco del desarrollo de la sociedad y su interacción con un

entorno cada vez más dinámico y cambiante, haciendo uso de dispositivos de comunicación inalámbrica autónomos.

Esta Tesis Doctoral utiliza las comunicaciones inalámbricas de bajo coste, para llevar a cabo la propuesta de nuevos mecanismos de enrutamiento orientados a proporcionar una arquitectura de comunicaciones inalámbricas en grandes escenarios, dado que los protocolos de enrutamiento actuales presentan claras deficiencias respecto al dinamismo exhibido cuando los escenarios tienen un gran tamaño.

2.1 MOTIVACIÓN SOCIAL Y PROBLEMÁTICA CIENTÍFICO-TÉCNICA

Teniendo en cuenta que en esta Tesis Doctoral se exponen técnicas de enrutamiento y formación de redes de sensores (**WSN**), resulta conveniente realizar una breve introducción, a modo de exposición acerca de este ámbito de trabajo e investigación.

La sociedad y la forma de vida actuales distan bastante de las de los inicios de las telecomunicaciones. A principios de siglo, las comunicaciones inalámbricas estaban empezando su expansión en forma de teléfonos móviles y las primeras líneas **GSM**. Internet vivía un proceso de expansión, pero todos esos cambios vivieron un cambio drástico con la llegada de los teléfonos inteligentes. Estos terminales revolucionaron la forma en que las personas interactúan entre ellas y cambió la forma de concebir la realidad. Los teléfonos inteligentes eran una ventana a un mundo de información e interacción en tiempo real y esto provocó el rápido auge de las tecnologías de comunicación inalámbrica y además como efecto colateral, el desarrollo de una necesidad de interacción con el entorno, nunca antes conocida por nuestra sociedad.

Esta nueva realidad presentó nuevos desafíos. La sociedad demandaba información y control sobre muchos elementos de la vida cotidiana, y así apareció el gran reto de los sistemas empotrados. A estos sistemas, largamente utilizados en multitud de productos cotidianos, se les dotó de otra función fundamental, la monitorización. Entiéndase esta monitorización como la recogida de datos. Desde entonces se han desarrollado nuevos sistemas empotrados con diferentes características, como el bajo consumo de energía o la posibilidad de comunicarse de forma inalámbrica, permitiendo su instalación en lugares remotos. Con este amplio desarrollo, se ha alcanzado un punto en el que iniciativas como *Sistemas Agrarios Inteligentes*, *Ciudades Inteligentes* o el mismo *Internet de las Cosas* comienzan a ser o son ya una realidad viable.

Proyectos como las *Ciudades Inteligentes* o el *Internet de las Cosas*, no solo se centran en la monitorización o interacción con sistemas empotrados, sino que también ofrecen nuevos servicios a la sociedad, alimentando esa nueva realidad de interacción en tiempo real. Hogares inteligentes en los que se conoce con exactitud el consumo de recursos como la electricidad, el agua o el gas. Sistemas de monitorización de tráfico,

WSN: Del inglés,
WIRELESS SENSOR
NETWORK.

GSM: Del inglés,
GLOBAL SYSTEM FOR
MOBILE
COMMUNICATIONS.

de aparcamiento, de residuos o medioambientales que proporcionan información actualizada a los ciudadanos. Para articular todo este magnum de datos e información demandada por los ciudadanos, se necesitan WSNs capaces de articular una arquitecturas de comunicaciones que cumpla con todas las necesidades actuales.

Las WSNs son, básicamente, pequeños y sencillos dispositivos con capacidades de comunicación inalámbricas, los cuales se comunican entre sí formando una red de comunicaciones. Esta red permite el envío de información y datos de un extremo a otro y mediante su utilización se podrán convertir en realidad todas estas nuevas aplicaciones que la sociedad demanda.

2.2 CAMPOS DE APLICACIÓN

Tal y como se ha comentado anteriormente, la motivación principal de esta Tesis Doctoral es proporcionar una aplicación real de la investigación, tanto a nivel científico como técnico, que sea útil a los intereses de la sociedad. Hoy en día las WSN son algo común y de uso cotidiano, pero solo a pequeña escala, con unos pocos nodos dispersos y con tareas muy concretas. Es por este hecho, que una vez se plantea ampliar su uso a grandes escenarios como las *Ciudades Inteligentes* o el *Internet de las Cosas*, aparecen problemas que deben ser resueltos. Con esta idea, se presentan a continuación algunos de los posibles campos de aplicación en los que la presente Tesis Doctoral podría significar un avance. Tras el análisis de estos campos de aplicación se procede a delimitar el marco de trabajo.



Figura 2.1
Cabezas de ganado en un latifundio uruguayo. Fuente: Agencia de Noticias de la Universidad Nacional de Colombia

2.2.1 Sistemas agrarios inteligentes

En un mundo en el que se busca digitalizar la mayoría de tareas, así como obtener datos en tiempo real para optimizar desde las cosechas al control del ganado, no pueden pasar desapercibidos los sistemas basados en redes de sensores. Estos sistemas ya son ampliamente utilizados hoy por hoy, pero su aplicación a latifundios o grandes extensiones, presenta un reto a la hora de proporcionar un soporte fiable y que ofrezca un amplio tiempo de uso, lo cual repercute directamente en los costes de mantenimiento.

Mediante el uso de una infraestructura adecuada, una gran plantación que discorra por diferentes micro-climas o diferentes tipos de suelo, permite un control en tiempo real de las necesidades de las plantas, maximizando su producción.

En la misma línea se mueve la aplicación al control de las cabezas de ganado. En amplias zonas de América, Asia o Australia, las cabezas de ganado gozan de una libertad controlada, como las mostradas en la Figura 2.1. Mediante el uso de una infraestructura adecuada, es posible conocer los patrones de movimiento del ganado, así como su localización e identificación, o incluso poseer información en tiempo real del estado de salud del animal, mejorando de esta forma sus cuidados y optimizando los resultados.

Actualmente existen soluciones mixtas basadas en GPS y GSM que requieren de costosos gastos de mantenimiento. También existen soluciones basadas en WSN, pero en los casos más extendidos como por ejemplo RPL-6LoWPAN, el envío de cada trama IPv6, conlleva enviar varios mensajes en la WSN, lo que condiciona en este tipo de entornos, la fiabilidad de la red. Es por ello que un nuevo protocolo de enrutamiento que resulte eficiente y que permita una rápida formación de las redes, posibilitando que los mensajes lleguen del emisor a un receptor, resulta de alto interés.

2.2.2 Ciudades inteligentes

Las Ciudades Inteligentes son otro de los elementos claves en la evolución de la sociedad. A través de ellas se articulan servicios para la ciudadanía, además de nuevas aplicaciones que permiten la ya comentada interacción con el entorno. A través de las Ciudades Inteligentes los ciudadanos pueden acceder a servicios como el *gobierno digital*, la *respuesta inmediata ante emergencias*, el *empleo sostenible de recursos*, la *información en tiempo real del estado del tráfico o del transporte público*, la *seguridad ciudadana*, etc.

Como se puede observar en la Figura 2.2, la Ciudad Inteligente es la evolución lógica de la relación entre la ciudad como ente y los propios

ciudadanos. Esta relación permite ahora una interacción mayor y más rápida, la cual repercutirá en el futuro, cambiando completamente el marco de aplicación de los servicios a disposición de los ciudadanos. Todas estas ventajas representan a su vez un reto para la infraestructura que debe dar el soporte. Dicha infraestructura no solo debe ser segura, sino que además debe resultar escalable, debe maximizar la cobertura y por supuesto debe adaptarse a los cambios de forma rápida y con el menor número de mensajes posible.

En las Ciudades Inteligentes, aún cuando la red de servicio para las infraestructuras se encuentre separada de las redes de uso cotidiano de los ciudadanos, estas deben alcanzar todos los rincones de la propia ciudad, cubriendo un gran terreno, y no solo en extensión, sino también llegando bajo tierra y a monitorizar incluso elementos del espacio aéreo. Todos estos servicios, que pueden presentar requisitos muy puntuales, representan en sí mismos el propio dinamismo que caracteriza estos escenarios, obligando a que los protocolos de comunicaciones sean igual de dinámicos. Actualmente las comunicaciones en una Ciudad Inteligente pasan por varios niveles, haciendo uso de redes de sensores pequeñas que captan la información y la van enviando a entes superiores. Estas conversiones entre diferentes tecnologías requieren de costosos elementos que permitan la interconexión. Toda esta complejidad sigue creciendo a medida que las Ciudades Inteligentes y los servicios ofertados por estas aumentan. Por este motivo, resulta de interés un nuevo protocolo que permita escalar y unificar todas estas redes de sensores y que ofrezca un buen compromiso entre dinamismo y escalabilidad, lo que permita en última instancia, simplificar el funcionamiento de las Ciudades Inteligentes.

2.2.3 Internet de las Cosas

La Internet de las Cosas (IoT) consiste en la búsqueda de una realidad en la que todos los objetos cotidianos se encuentren conectados a Internet y permitan su acceso de forma remota, no solo a los servicios ofre-



Figura 2.2
Ejemplo de Ciudad Inteligente y sus diferentes servicios. Fuente: <http://blog.hemav.com/>

cidos por la Ciudad Inteligente, sino también a cualquier objeto que se encuentre en el ámbito privado de los ciudadanos, como pueden ser objetos de su hogar o trabajo, transportes personales, comunicaciones privadas, etc.

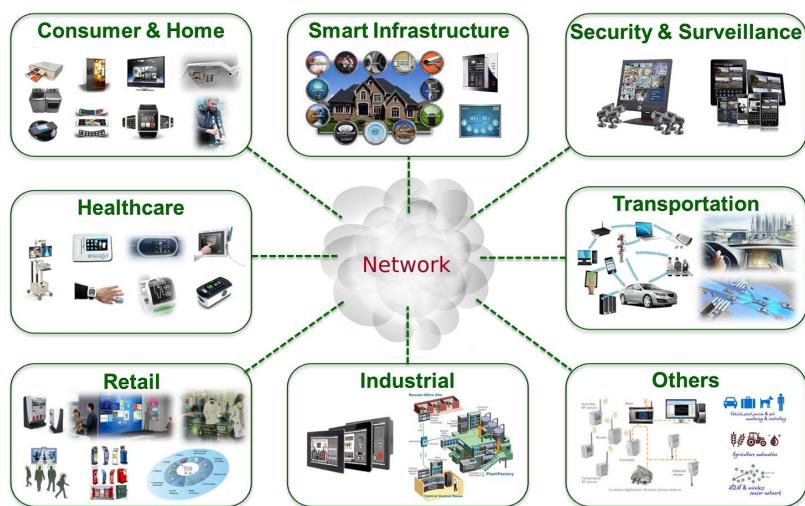
Como se puede observar en la Figura 2.3, el Internet de las Cosas, se puede definir como la unión de todos los diferentes escenarios, entre los que se encuentran:

- Sistemas Agrarios Inteligentes.
- Hogares Inteligentes.
- Ciudades Inteligentes.
- Transporte Inteligente.
- Industria Inteligente.
- Sistemas de Salud Inteligentes.
- Etc.

La conjunción de todos estos diferentes sistemas presenta un reto a nivel de escalabilidad del sistema, dado que el número de nodos presente en estos escenarios se incrementará enormemente. Los protocolos actuales presentan un problema dado que a nivel de dar soporte a la infraestructura y con tantos nodos involucrados, tienen un dinamismo muy reducido, lo que es difícilmente adaptable.

Otro problema de muchos de los protocolos actuales que se basan en sincronización de tiempo para asegurar una mayor fiabilidad en las comunicaciones, es el hecho de que mediante el uso de mecanismos de

Figura 2.3
Ejemplo de Internet de las Cosas y sus diferentes áreas de aplicación. Fuente: Vivante Corp.



Vivante and the Vivante logo are trademarks of Vivante Corporation. All other product, image or service names in this presentation are the property of their respective owners. © 2013 Vivante Corporation

sincronización, se limita el número de nodos que pueden pertenecer a la red. Adicionalmente, estos protocolos presentan otro problema, y es que a pesar de aumentar la fiabilidad, limitan también la comunicación en paralelo que permiten otras soluciones.

La aproximación más adecuada de las actuales es *RPL-6LoWPAN*, pero como se explicó en la Sección 2.2.1, este protocolo conlleva asociado una elevado número de mensajes para poder realizar la comunicación a nivel de IPv6. Como consecuencia, el dinamismo y la capacidad de adaptarse a eventuales problemas en la red se ve mermada. Por los motivos expuestos, resulta de interés el desarrollo de un nuevo protocolo de radio que optimice el dinamismo, el número de mensajes enviados y permita la comunicación en paralelo en diferentes áreas de la red.

2.3 MARCO DE TRABAJO

Tras plantear la motivación y diferentes ámbitos de aplicación de esta Tesis Doctoral, se acota y define a continuación el marco de trabajo en el que se ha desarrollado la presente investigación. El uso de WSN como elemento articulador de la sociedad y de las Ciudades Inteligentes, requiere unos sistemas de comunicación ágiles y dinámicos que aporten una nueva perspectiva en cuanto a la formación y estructuración de las redes, alcanzándose tiempos reducidos en grandes escenarios y permitiendo cambios de la topología de red de forma rápida.

Los escenarios ágiles deben caracterizarse por un tiempo de convergencia bajo y un número bajo y constante de mensajes que le permitan a un nodo conectarse a dicha red. Estos elementos contribuyen de forma directa, no solo a una rápida respuesta en caso de que se produzca la desconexión de uno o varios nodos, sino un ahorro en el consumo de energía, tal y como demostraron Soro y Heizelman en su trabajo [75].

2.3.1 Caracterización de las comunicaciones en WSN

Las redes de sensores se han desarrollado enormemente en los últimos años. La evolución lógica de este campo ha incluido áreas de investigación como la eficiencia energética, el ancho de banda, la seguridad, el alcance de las redes o la adaptación a cambios del entorno.

La *eficiencia energética* es, quizás, el mayor objetivo a alcanzar por cualquier WSN. Hasta la fecha, muchas son las propuestas lanzadas por fabricantes e investigadores para mejorar el tiempo de vida de todos y cada uno de los nodos que forman la red [2, 17, 20, 25, 53]. Hoy en día se ha avanzando tanto en el campo de la eficiencia energética, que ya se plantean sistemas WSN alimentados por el propio cuerpo humano

[71], dando una idea de la extrema eficiencia que se está llegando a alcanzar.

La *seguridad* es otro de los elementos clave en una WSN. El desarrollo de estos sistemas y su implantación en entornos cada vez más críticos, conlleva asociado un desarrollo en las técnicas de seguridad que, en todo caso, deben evitar el robo de información. Este campo también ha llevado una amplia investigación, y de ello dan cuenta autores como Agah et al. [24], Deng et al. [33] y Khalil et al. [54] entre otros.

La *cobertura* es otro elemento que ha supuesto numerosas investigaciones dentro de las WSN. La cobertura de una red no es solamente alcanzar determinados nodos con una calidad en la señal, sino ampliar el alcance de la propia red permitiendo que todos los nodos, independientemente de su localización puedan conectarse a la red. En este sentido, estudios como los de Arivudainambi et al. [26], Sakkari et al. [36] y Alam et al. [79] entre otros, dan cuenta del interés en esta problemática en los últimos años.

El *dinamismo* es otra de las características deseables en una WSN hoy en día. Este dinamismo abarca desde la respuesta rápida ante problemas, así como a la adaptación al entorno. En este área es donde se pueden encontrar los estudios más recientes dentro de las WSN. Los estudios de Mercadal et al. [62] y Fernandez-Marquez et al. [40] entre otros así lo corroboran.

Todas las características aquí citadas forman parte de lo que se considera el conjunto de características ideales que debe optimizar una arquitectura de comunicaciones. El diseño de una arquitectura de comunicaciones es aquella disciplina que se centra en el diseño de las propias redes, proveyendo de un soporte a las comunicaciones, con independencia de la aplicación, pero estrechamente ligado al escenario donde se producirá el despliegue de dicha red. En este sentido, el desarrollo y estudio de arquitecturas de comunicaciones que potencien la *seguridad*, *cobertura* y *dinamismo* de las redes, es altamente deseable. Sin una arquitectura de comunicaciones adaptada a las necesidades de la sociedad, será imposible continuar con la evolución y el ritmo de desarrollo actual y eso es algo que desde la comunidad científica debe ser resuelto. Por este motivo, se considera que el desarrollo de un protocolo de comunicaciones inalámbricas que proporcione una arquitectura de comunicaciones segura, de largo alcance y dinámica, es necesario para la consecución de objetivos como la *Ciudad Inteligente* o el *Internet de las Cosas*.

2.3.2 Densidad de nodos en WSN

La densidad de nodos es una métrica utilizada en redes y que se deriva de las matemáticas. En esta métrica se evalúan el número de enlaces

que un nodo puede crear en base a la potencia de emisión y a la distancia entre nodos.

Mediante este métrica es posible calcular la cantidad de tráfico potencial que existirá en una red, así como calcular la interferencia que los nodos provocarán unos con otros, resultando a su vez en más colisiones.

La densidad es una métrica que resulta de interés en grandes escenarios como las *Ciudades Inteligentes* o el *Internet de las Cosas* dado que permite obtener una imagen representativa de la distribución de los nodos y a su vez calcular potenciales requisitos.

Por otro lado, se deben analizar los servicios que la WSN debe satisfacer (p.e. Ciudades inteligentes, servicios de seguridad, etc.), de forma que se considere también la densidad aplicable a los mismos. Esta Tesis Doctoral está especialmente centrada en infraestructuras como las que componen la *Ciudad Inteligente*, las cuales debido a la distancia entre los diferentes elementos que componen la infraestructura y por características como la regulación en materia de potencia de emisión o el uso de baterías que limiten el uso de la radio, se basan en una densidad baja o media.

2.3.3 *Tiempo de convergencia en WSN*

Toda red WSN, tiene asociados una serie de tiempos de procesamiento, de vuelo de paquetes, de formación, etc. El *tiempo de convergencia* es una medida del tiempo necesario para que una serie de nodos se conecten a una misma red. Este valor hace medible un concepto como el dinamismo, dando cuenta de la medida de tiempo que una red necesitará para modificar su topología y para reaccionar ante eventuales desconexiones de algunos de sus elementos.

Durante el diseño y la planificación de un despliegue WSN, el *tiempo de convergencia* debe ser considerado como una parámetro clave, dado que en una red autónoma, un determinado tiempo sin conexión puede conllevar consecuencias potencialmente catastróficas. Por ello el tiempo de convergencia nos mostrará la respuesta ante fenómenos inesperados que podrá tener la red.

Por otro lado, el *tiempo de convergencia* es un indicador que permite estimar parcialmente, el *dinamismo* al que se enfrenta un protocolo de comunicaciones, ya que permite conocer dos elementos claves, el tiempo necesario para formar la red en sí misma y la rapidez con que reaccionará frente a cambios en el entorno.

2.3.4 Número de Mensajes de control en WSN

Todos los algoritmos de enrutamiento WSN, por su propia naturaleza inalámbrica, necesitan enviar mensajes para conectarse, sincronizarse, enviarse información, actualizar el estado de la propia red, etc. Por este motivo, una medida basada en el número de los mensajes de control, también resulta de utilidad a la hora de valorar la capacidad de reacción de la red. En tanto una red necesite un elevado número de mensajes para tareas como la conexión, esto conllevará asociada una menor capacidad de reacción ante fenómenos del entorno o ante problemas derivados de la propia red, (p.e. agotamiento de baterías).

El *número de mensajes de control* es otro de los parámetros que permiten estimar parcialmente el *dinamismo*, y en conjunción con el *tiempo de convergencia*, permite obtener una buena medida del *dinamismo* de un algoritmo o protocolo.

2.4 PLANTEAMIENTO

A continuación se expone el planteamiento inicial del trabajo de la presente Tesis Doctoral. Esta sección sirve como resumen de lo expuesto, y muestra las primeras decisiones de diseño, que han servido como punto de inicio de esta investigación.

2.4.1 Problemática

La problemática que se desea resolver es la siguiente:

Normalmente los nodos de un escenario grande que forman parte de una red WSN, y que por las características del escenario formen una gran red, se basan en algoritmos de enrutamiento que no pueden ofrecer un compromiso adecuado entre escalabilidad y dinamismo. Esto conlleva dos problemas asociados, el primero es que no todos los nodos existentes en ese gran escenario llegan a conectarse y el segundo, que no se pueden articular respuestas dinámicas a cambios externos o surgidos en el seno de la propia red.

Como se ha comentado previamente, las grandes infraestructuras presentan una densidad media o baja. Estos escenarios presentan propiedades muy específicas de dispersión de los nodos de una red. En este tipo de escenarios aún encontrándose los nodos en rango de alcance, presentan dificultades para encontrar diferentes nodos a los que conec-

tarse. Este hecho resulta en un caso de especial interés para los algoritmos de enrutamiento más utilizados hoy en día. Como se describirá en las Secciones 3.2.1 y 3.2.1, tanto *RPL* como *AODV* resultan en una pérdida mayúscula de dinamismo en términos de formación de la red.

2.4.2 Densidad de nodos asociada a los grandes escenarios

Las comunicaciones en grandes escenarios como las *Ciudades Inteligentes* o el *Internet de las Cosas* presentan un denominador común, la infraestructura de comunicaciones.

Esta infraestructura se compone, por lo general, de una combinación de elementos fijos y móviles, los cuáles se ven sometidos a variaciones en sus conexiones debido al movimiento y la interacción con el entorno.

Debido a que la infraestructura debe proveer el soporte para diferentes servicios y para otros nodos, los dispositivos que la componen deben esparcirse de forma que proporcionen la máxima cobertura posible. Un buen ejemplo de estas necesidades son los trabajos desarrollados por Calderoni et al. [55], Lécué et al. [58] o Sanchez et al. [76], en los que se presentan estudios basados en *Ciudades Inteligentes* con estudios específicos sobre la distribución ideal de los nodos. Otros estudios como los de Kamouskos et al. [73] o Morvaj et al. [65] plantean diferentes propuestas de simulación de sistemas basados en *Smart Grid*, en las que los nodos se posicionan de forma lo suficientemente espaciada.

$$ND = \frac{(x - 1) \times Range^2 \times \pi}{Size_x \times Size_y} \quad (2.1)$$

Tomando como base estos estudios, se propone en esta Tesis Doctoral, la utilización de una densidad de nodos para grandes escenarios de nivel medio o bajo, dependiendo de la cantidad de servicios disponibles en el escenario y basada en la ecuación 2.1. La definición de densidad alta, media o baja, se puede encontrar en el trabajo de H. Kermajani et al. [41], el cual define la densidad alta como 15, la media como 10 y la baja como 5. Estos valores de densidad serán utilizados en el presente trabajo, sentando la base para el diseño de las propuestas.

La aplicación de una densidad media o baja en los grandes escenarios, conlleva asociados problemas adicionales a la hora de localizar y formar rutas en muchos de los algoritmos de enrutamiento actuales. Esto se debe a que gran parte de las métricas utilizadas se basan en la eliminación de redundancias lo cual lleva a evitar formar rutas que pasen por un mismo nodo en varias ocasiones. La utilización de este tipo de métricas provoca que en condiciones de media o baja densidad, donde la distribución de los nodos tiende a ser muy dispersa, se produzcan redundancias que no pueden ser evitadas, lo que afecta a los

Figura 2.4
Evolución del tiempo de convergencia respecto a la densidad para *RPL*.

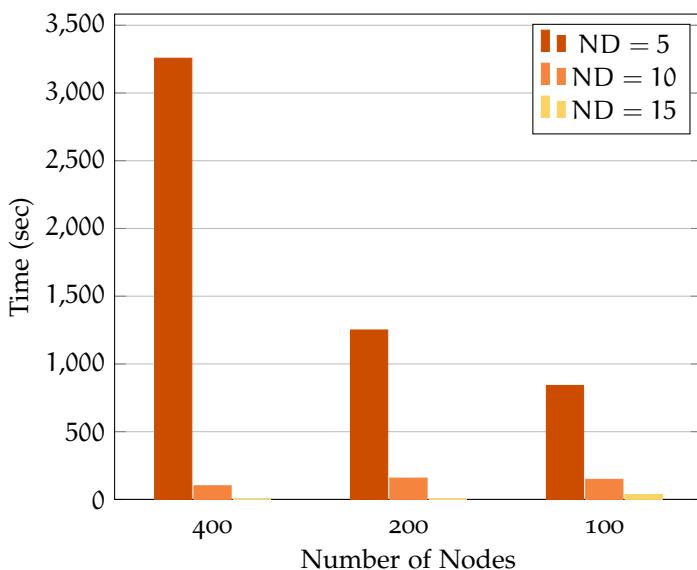
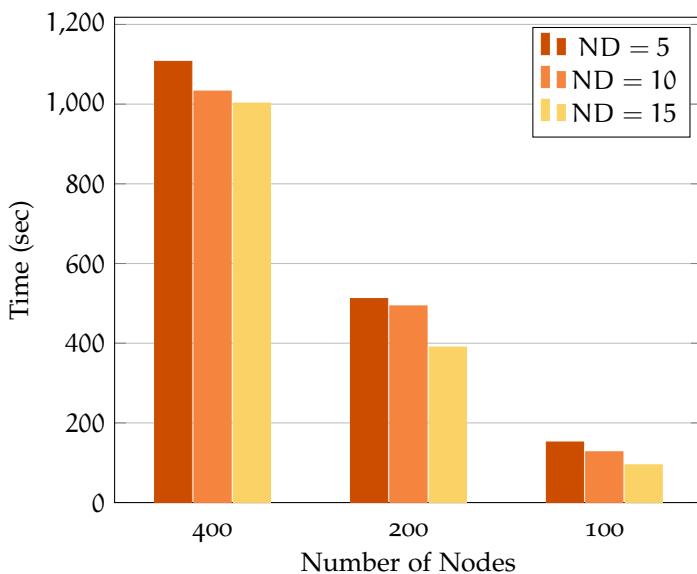


Figura 2.5
Evolución del tiempo de convergencia respecto a la densidad para *AODV*.



algoritmos que buscan y construyen las rutas de una red. Todos estos elementos tienen una importante repercusión en cómo se comportan los algoritmos de enrutamiento, tal y como se analiza en las siguientes sub-secciones.

Relación entre la densidad de la red y el tiempo de convergencia

Las Figuras 2.4 y 2.5 presentan la evolución del TIEMPO DE CONVERGENCIA con respecto a la densidad para *RPL* y *AODV* respectivamente. En ambas figuras se puede apreciar:

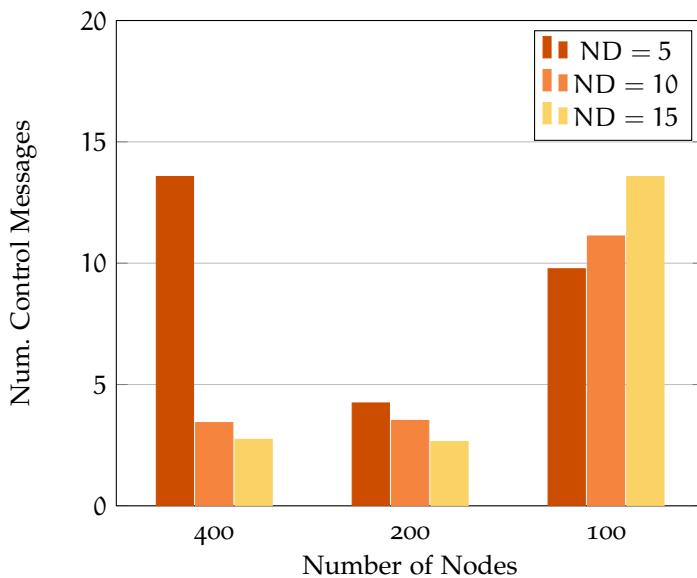


Figura 2.6
Evolución del número de mensajes de control enviados respecto a la densidad para *RPL*.

- Valores altos de tiempo de convergencia por nodo en escenarios de media o baja densidad.
- Un progresivo descenso del *tiempo de convergencia* respecto al incremento de la densidad de nodos.

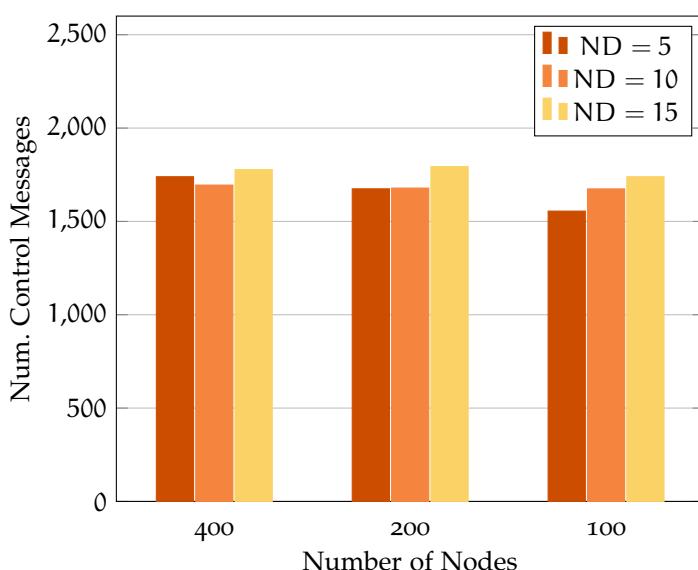
Estos resultados permiten ratificar la problemática existente para los escenarios de media y baja densidad mediante el uso de estos algoritmos y es por ello que resulta de interés la mejora de estos resultados para obtener una mejora del dinamismo de la red puesto que las grandes infraestructuras presentan usualmente una densidad de nodos media o baja, se espera que la mejora de estos resultados implique una mejora en la gestión de este tipo de infraestructuras.

Relación entre la densidad de la red y el número de mensajes de control enviados

En esta ocasión, el NÚMERO DE MENSAJES DE CONTROL ENVIADOS se muestra en las Figuras 2.6 y 2.7. Lo normal sería mantener un número de envíos estable o con pequeñas fluctuaciones, con independencia del escenario y la densidad. Sin embargo:

- Aunque *RPL* presenta una cierta independencia de la densidad, se aprecia cierta relación entre el número de nodos y el número de mensajes enviados.
- *AODV* muestra una relación directa entre la densidad de la red y el número de mensajes enviados. A medida que aumenta la densidad, mayor es el *número de mensajes de control enviados*.

Figura 2.7
Evolución del número de mensajes de control enviados respecto a la densidad para AODV.



Estos resultados presentan la estrecha relación existente entre el *número de mensajes de control enviados* y la densidad o el número de nodos presentes en los escenarios.

Los resultados del presente análisis, junto con los obtenidos a partir del *tiempo de convergencia* para algunos de los algoritmos de enrutamiento más utilizados hoy en día, permiten observar un problema con el dinamismo de la red y la capacidad limitada de dichos algoritmos de ofrecer respuesta a los retos que plantean los grandes escenarios en condiciones de media o baja densidad. Este hecho justifica el desarrollo de un nuevo algoritmo de enrutamiento enfocado tanto en la minimización del *tiempo de convergencia* y del *número de mensajes de control enviados* para grandes escenarios en condiciones de media o baja densidad como las que típicamente presentan la gestión de grandes infraestructuras.

2.4.3 *Objetivos científicos de la Tesis*

La presente Tesis Doctoral presenta varios objetivos, los cuales marcan las directrices generales de la investigación y pretenden dar respuesta a preguntas como: *¿Qué debe realizar el protocolo?* y *¿Cómo de buenos son los resultados que proporciona?*.

OBJETIVOS CIENTÍFICOS

Como OBJETIVOS CIENTÍFICOS se desea:

- A. Desarrollo de un algoritmo de búsqueda y creación de rutas en grandes escenarios de media o baja densidad.
- B. Desarrollo de un algoritmo de enrutamiento que escale correctamente, a la vez que minimiza el tiempo de convergencia y el número de mensajes de control enviados.
- C. Integración de un protocolo de enrutamiento a partir del algoritmo desarrollado en el punto anterior.

2.5 ESTRUCTURA DE LA TESIS

Esta Tesis Doctoral se compone de cinco partes, que a su vez se han dividido en capítulos. Las partes abordadas en este Tesis Doctoral son las descritas a continuación:

INTRODUCCIÓN La Parte I ha expuesto la problemática a la que se enfrenta esta Tesis. A lo largo de esta Parte, se han presentado las diferentes alternativas, así como el escenario de aplicación de esta tecnología. Al final del capítulo se han fijado los **OBJETIVOS** y el plan de actuación inicial a desarrollar para resolver la problemática a la que hace frente esta Tesis Doctoral.

REVISIÓN BIBLIOGRÁFICA La Parte II de la Tesis Doctoral abarca un capítulos. En él se estudia la literatura actual relacionada con ciudades inteligentes y el estudio de la densidad en WSN, así como el **ESTADO DEL ARTE** de los **OBJETIVOS** propuestos en la Parte I. En esta parte se analizan además los principales protocolos de comunicación inalámbrica existentes en WSN, prestando especial atención a la literatura relacionada con la problemática considerada.

MÉTODOS Y MATERIALES La Parte III de esta Tesis Doctoral consta de un capítulo en él se presentan los dos algoritmos y el protocolo que se proponen en esta tesis. Según la **HIPÓTESIS DE TRABAJO**, estos métodos deben mejorar el tiempo de convergencia y mantener un bajo y constante número de mensajes de control enviados en WSNs aplicadas a grandes escenarios con una media o baja densidad.

RESULTADOS EXPERIMENTALES La Parte IV presenta todas las pruebas experimentales realizadas en este trabajo. A fin de resultar sencillos a ojos del lector, se han articulado tres capítulos. El primero contempla la búsqueda de una configuración óptima de *DARP* para grandes escenarios, así como el *Análisis de Comportamiento de DARP*.

El segundo capítulo contiene la *Configuración de la comparativa entre algoritmos* y los resultados de dicha *Comparativa de algoritmos*.

Finalmente, el tercer capítulo presenta una *Evaluación de las características adicionales de DARP*, en forma de análisis de las características adicionales que este protocolo posee.

DISCUSIÓN Y CONCLUSIONES Esta parte está compuesta por dos capítulos, en los que se exponen las discusiones de los resultados finales, las conclusiones y las principales aportaciones científicas que proporciona esta Tesis Doctoral. Adicionalmente se presentan las vías de investigación futura basadas en la presente Tesis Doctoral.

BIBLIOGRAFÍA Y ANEXOS Esta última Parte incluye todas las referencias utilizadas en la Tesis. Para una correcta comprensión de la temática y la propuesta de esta Tesis Doctoral, se añaden una serie de anexos: *Plataformas actuales en WSN*, *OMNeT++ Tutorial* y *DARP Technical Manual*.

Parte III

REVISIÓN BIBLIOGRÁFICA

3

COMUNICACIONES INALÁMBRICAS

ÍNDICE

3.1	Estándares de comunicaciones inalámbricas	37
3.1.1	ISA100.11a	37
3.1.2	ISO 18000-7	37
3.1.3	IEEE 802.11	38
3.1.4	IEEE 802.15.1	39
3.1.5	IEEE 802.15.3	39
3.1.6	IEEE 802.15.4	40
3.2	Algoritmos de capa de red	43
3.2.1	Algoritmos de Enrutamiento	43
3.2.2	Algoritmos de Enrutamiento basados en Clustering	47
3.3	Dinamismo en una WSN	48
3.4	Conclusiones	49

Como se comentó previamente, las WSN se encuentran a la vanguardia del cambio en la sociedad, revolucionando la relación entre el entorno y las personas, así como cambiando a la propia sociedad. A lo largo de este capítulo se desgrana el concepto de WSN, abordando qué es y qué la forma.

Analizando la cuestión del qué y tomando una definición ampliamente aceptada y utilizada, se puede definir una WSN como "*una red de ordenadores muy pequeños (nodos), equipados con sensores, que colaboran en una tarea común*"[86].

En la propia definición podemos encontrar referencia al qué la forma. Esos *ordenadores muy pequeños o nodos, equipados con sensores* son esencialmente sistemas empotrados basados en microcontroladores con capacidad de comunicación inalámbrica y equipada con diferentes sensores, de entre una amplia variedad, como por ejemplo: ruido, humedad, luz, contaminación, peso, consumo de energía, presión, etc. Más información acerca de los componentes hardware de una WSN puede ser encontrada en el Anexo A.

Por supuesto un nodo no se compone solo de hardware físico, sino que también tiene una parte software, más importante aún si cabe. Los sistemas empotrados modernos se basan en un software relativamente complejo, el cuál variará dependiendo de la aplicación en la que será empleado. Hoy existen desde aplicaciones específicas a sistemas operativos que pueden ser utilizados en estos sistemas. En el Anexo A

se realiza un análisis más detallado de estos sistemas, ya que no forman parte del objetivo principal de esta tesis, aunque se consideran un elemento de relevante importancia en lo que es un sistema empotrado.

Con independencia del soporte software, ya sea un sistema operativo o un software específico, los sistemas empotrados necesitan una pila de comunicaciones inalámbricas. Esta pila de comunicaciones se compone de diversas capas, como puede verse en la figura 3.1. Existen una serie de capas alineadas, así como unos planos transversales a todas ellas, que son los que interactúan estrechamente con el software del propio sistema empotrado. Las capas de las que se compone una pila de comunicaciones WSN son cinco:

APL: Del inglés,
APPLICATION LAYER.

TL: Del inglés,
TRANSPORT LAYER.

NWK: Del inglés,
NETWORK.

MAC: Del inglés,
MEDIUM ACCESS
CONTROL.

PHY: Del inglés,
PHYSICAL.

- CAPA DE APLICACIÓN (APL), con el objetivo de ejecutar la aplicación específica de un determinado nodo de la red. Estas aplicaciones son generalmente las que se encargan de la tarea de recopilar o analizar datos.
- CAPA DE TRANSPORTE (TL), con el fin de proporcionar un transporte de datos confiable entre un origen y un destino. Su tarea es la transferencia libre de errores de los datos recibidos por la capa de aplicación entre el nodo emisor y el nodo receptor.
- CAPA DE RED (NWK), con la tarea de proporcionar conectividad entre dos nodos. Su misión principal es conseguir que los datos lleguen desde el origen al destino, aún cuando no exista conectividad directa entre ellos, mediante *Técnicas de Enrutamiento*.
- CAPA DE ACCESO AL MEDIO (MAC) o de *Enlace de Datos* es una capa que se encarga de la organización de los datos en bloques de información (llamados tramas), proporciona una dirección de capa de enlace (dirección MAC), gestiona el control de flujo y desarrolla tareas de corrección de errores.
- CAPA FÍSICA (PHY), con el objetivo de transformar los datos en secuencias de bits que puedan ser transmitidos de un punto a otro.

Asociados a las comunicaciones inalámbricas existen diversos estándares que son más o menos conocidos por el gran público, pero ampliamente utilizados. A continuación, se presenta una visión general sobre dichos estándares.



Figura 3.1
Pila de Comunicaciones para WSN.
Fuente: Overview of Wireless Sensor Network [60].

3.1 ESTÁNDARES DE COMUNICACIONES INALÁMBRICAS

Esta sección presenta los estándares de comunicaciones inalámbricas más utilizados actualmente. Cabe destacar que la mayoría de estándares definen únicamente las capas física y de enlace.

3.1.1 ISA100.11a

ISA 100.11a [47] es un estándar de la sociedad internacional de automatización específicamente diseñado para el desarrollo de tareas de automatización y monitorización inalámbrica de procesos industriales. Define las capas de bajo nivel (PHY y MAC) y funciona en la banda ISM de los 2,4GHz, con un esquema de salto de frecuencia, incrementando la fiabilidad y reduciendo las interferencias producidas por otros canales.

ISM: Del inglés,
INDUSTRIAL,
SCIENTIFIC AND
MEDICAL

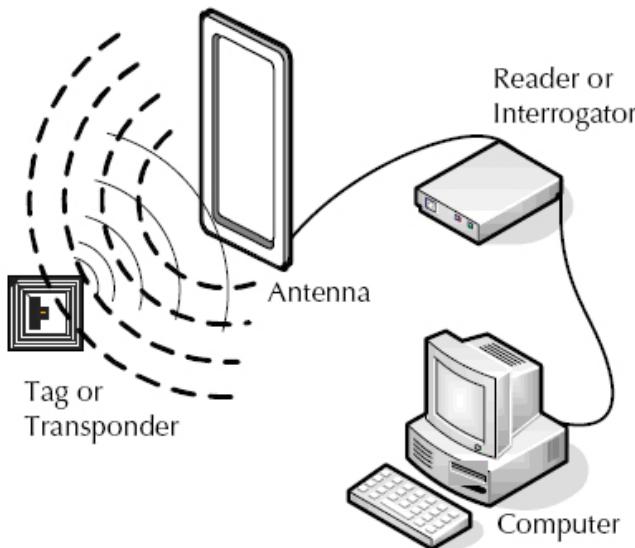
El objetivo principal de *ISA 100.11a* es el desarrollo de dispositivos con un bajo consumo energético, robustez en las comunicaciones, escalabilidad e interoperabilidad con otros sistemas inalámbricos.

3.1.2 ISO 18000-7

La norma *ISO 18000-7* [48] define las capas PHY y MAC para comunicaciones de identificación por radio frecuencia (**RFID**) en la banda de

RFID: Del inglés,
RADIO FREQUENCY
IDENTIFICATION

Figura 3.2
Ejemplo de sistema RFID. Fuente:
<http://www.alexan-tech.com>



frecuencia de los 433MHz.

RFID se basa en el uso de etiquetas (*o tags*) pasivas que no necesitan de ningún tipo de batería. Como se puede observar en la Figura 3.2, estas etiquetas se activan al recibir la señal de un lector que se encuentre a una distancia muy corta, el cuál con la propia energía emitida, alimenta la etiqueta y puede leer su contenido. Una vez obtenido el contenido, este se envía a un ordenador u otro sistema para su procesamiento. Debido a estas características, esta tecnología se caracteriza por una muy limitada transmisión de datos, un bajo costo y un reducido consumo energético.

3.1.3 IEEE 802.11

Este estándar [43] define la norma para las capas física y de enlace de las redes inalámbricas de área local (**WLANs**). Es el estándar de referencia en el que se basa la tecnología *Wireless Fidelity* (Wi-Fi) [21]. Wi-Fi soporta conexiones de alta velocidad (mayores a 11Mbps) a corta/media distancia.

Wi-Fi es una tecnología ampliamente conocida y utilizada hoy en día, que se encuentra presente en una infinidad de dispositivos, desde ordenadores a teléfonos inteligentes, pasando por aparatos multimedia, periféricos, transportes, etc. Esta tecnología es muy utilizada en dispositivos multimedia debido a las altas tasas de transferencia de datos, a su masiva integración en numerosos dispositivos y a la posibilidad de conexión directa entre dispositivos.

3.1.4 IEEE 802.15.1

El estándar *IEEE 802.15.1* [44] define las capas de bajo nivel (PHY y MAC) de las redes inalámbricas de área personal (**WPANs**). Este estándar se centra en la definición de redes con un número reducido de nodos y a corta distancia. Una de las claves de este estándar es que las redes necesitan, dependiendo del escenario, una baja infraestructura o incluso pueden no llegar a requerirla. El objetivo de estas redes es la transmisión a alta velocidad de datos entre grupos privados, limitando el alcance fuera de estos grupos.

WPANs: Del inglés,
WIRELESS PERSONAL
AREA NETWORKS

Este estándar es la base de una tecnología muy conocida y utilizada hoy en día, *Bluetooth* [5], presente en la vida diaria a través de los teléfonos inteligentes, tabletas, ordenadores, dispositivos multimedia y dispositivos médicos [4]. *Bluetooth*, al igual que *Wi-Fi* es ampliamente utilizado en dispositivos multimedia debido a las altas tasas de transferencia de datos y a la posibilidad de realizar conexiones directas entre dispositivos sin infraestructura.

El nombre de *Bluetooth* procede del nombre del rey danés y noruego *Harald Bluetooth*, conocido por unificar a las tribus noruegas y suecas-danesas y por convertirlas al cristianismo. Su nombre, se utiliza de forma representativa de la unificación entre las comunicaciones inalámbricas de ordenadores y dispositivos móviles.

3.1.5 IEEE 802.15.3

El estándar *IEEE 802.15.3* [45] o *Ultra Wide Band* (UWB) define las redes inalámbricas de área personal (WPANs) de alta velocidad. Funciona en la banda de 2,4GHz, llegando a ofrecer tasas de envío entre 11 y 55 Mbps. Hace uso de un mecanismo de acceso al medio por división de tiempo (**TDMA**) para garantizar un acceso seguro y sin colisiones al medio. Ofrece la posibilidad de realizar transferencias síncronas y asíncronas. Además permite la gestión del consumo de potencia, de la tasa de envío de datos y de la frecuencia de transmisión.

TDMA: Del inglés,
TIME DIVISION
MULTIPLE ACCESS

Este estándar es muy utilizado en aplicaciones de seguridad y localización. Sus características lo hacen muy atractivo para la utilización en transmisiones de audio y vídeo en tiempo real. Actualmente ha centrado mucha atención de la comunidad investigadora y está concentrando mucho esfuerzo en su desarrollo.

3.1.6 IEEE 802.15.4

El estándar *IEEE 802.15.4* [46], en sus diferentes versiones desde la original de 2003, especifica las capas PHY y MAC de una Red de Área Personal de baja velocidad (**LR-WPAN**). Como en el caso de otros estándares, el *IEEE 802.15.4* sienta las bases para el desarrollo de algoritmos de capa de red que habiliten la formación de protocolos de comunicaciones, como por ejemplo ZigBee [25], RPL-6LoWPAN [87] o WirelessHART [70].

El estándar *IEEE 802.15.4* se ha consolidado como uno de los referentes en sistemas para *Hogares Inteligentes* o *Sistemas Agrarios Inteligentes* porque ofrece comunicaciones inalámbricas de *bajo coste* y *bajo consumo energético* en entornos en los que no son necesarias grandes transferencias de datos, permitiendo de esta forma la producción de sistemas baratos y que pueden ser alimentados mediante baterías.

Debido a que el estándar *IEEE 802.15.4* es la base del protocolo de comunicaciones propuesto en esta Tesis Doctoral, resulta de interés presentarlo de forma más detallada.

Capa Física (PHY)

El estándar ofrece cinco bandas de frecuencia para operar, aunque estas se encuentran limitadas geográficamente. La banda ISM a 2.4GHz se puede utilizar a nivel mundial con 16 canales disponibles, la de 915MHz en Norte América con 10 canales, la de 868Mhz en Europa con 3 canales, la de 950Mhz en Japón con 8 canales y la de 780Mhz en China también con 8 canales.

La Figura 3.3 muestra una perspectiva general de las diferentes bandas de frecuencia junto con los distintos tipos de modulación disponibles. Respecto a la modulación, existen diferentes técnicas, aunque este trabajo no profundiza en su análisis, resulta de interés comentar las más relevantes [31], por ello se presenta la Tabla 3.1 a modo de resumen.

Capa de Acceso al Medio (MAC)

El estándar *IEEE 802.15.4* define no solo la capa PHY, sino también la capa MAC. En esta capa la información se organiza en tramas compuestas de tres campos: una *Cabecera*, una *Carga útil* y una *Secuencia de comprobación de la trama (FCS)*, como se puede apreciar en la figura 3.4. La *Cabecera* tiene un tamaño variable, entre 3 y 23 Bytes, e incluye

802.15.4 PHY Overview (data rate vs frequency)

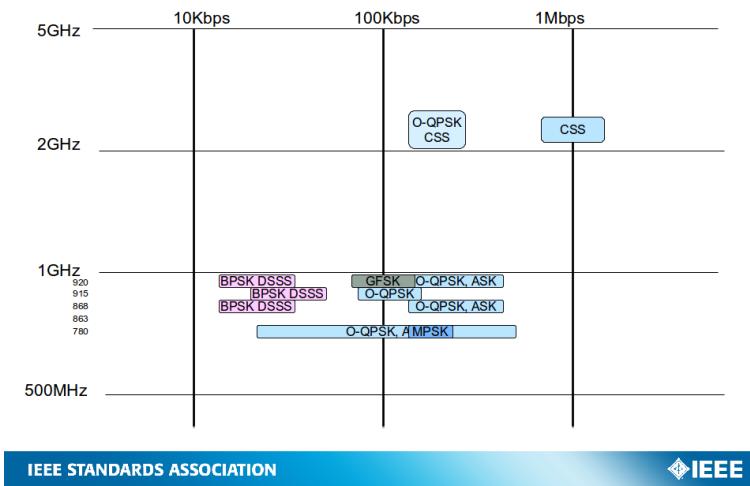
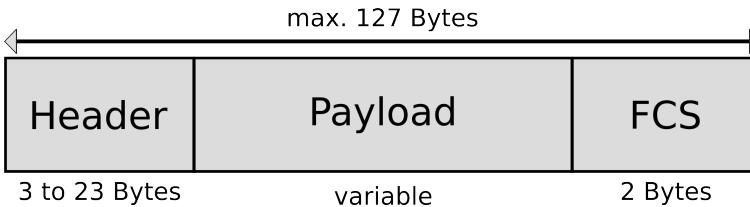


Figura 3.3
Bandas de Frecuencias y tipos de modulaciones en IEEE 802.15.4. Fuente: Tim Godfrey for the 802.24.1 Smart Grid TAG Consolidated White Paper Presentation (17.07.2015)

IEEE STANDARDS ASSOCIATION IEEE

Figura 3.4
Trama 802.15.4



información de control del estándar, además de las direcciones de los nodos origen y destino. La *Carga útil* tiene un tamaño variable e incluye la información a trasmisir. La *Secuencia de comprobación de la trama* tiene un tamaño de 2 Bytes y se utiliza para la corrección de errores.

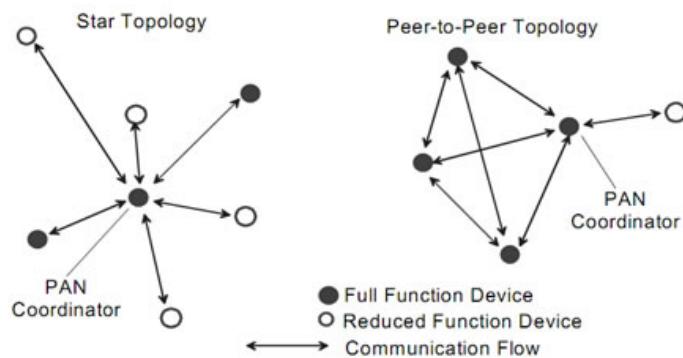
La capa MAC ofrece dos modos de operación diferentes:

Referencia	Frecuencia (MHz)	Modulación	Transferencia (Kbps)
CSS	2400–2483.5	D-QPSK	250
CSS	2400–2483.5	8-/64-ary*	1000
950 MHz	950	BPSK	20
950 MHz	950	GFSK	100
915 MHz	902–928	O-QPSK	250
915 MHz	902–928	ASK	250
868 MHz	868	O-QPSK	100
868 MHz	868	ASK	250
780 MHz	780	O-QPSK	250
780 MHz	780	MPSK	250

Tabla 3.1
Resumen general de frecuencias y tipos de modulación.

8-/64-ary* makes reference to 8-/64-ary orthogonal codes.

Figura 3.5
Posibles Topologías
basadas en IEEE
802.15.4. Fuente:
<http://ecee.colorado.edu/>



CSMA/CA: del inglés,
CARRIER SENSE
MULTIPLE ACCESS
WITH COLLISION
AVOIDANCE

CSMA/CD: del inglés,
CARRIER SENSE
MULTIPLE ACCESS
WITH COLLISION
DETECTION

RFD: Del inglés,
REDUCED-FUNCTION
DEVICE

FFD: Del inglés,
FULL-FUNCTION
DEVICE

PAN: Del inglés,
PERSONAL AREA
NETWORK

TOPOLOGÍAS: Son las
formas en que están
diseñadas las redes,
sea en el plano físico o
lógico.

**TOPOLOGÍA EN
ESTRELLA:** Del inglés,
STAR TOPOLOGY

**TOPOLOGÍA
PUNTO-A-PUNTO:** Del
inglés, PEER-TO-PEER
TOPOLOGY

- Modo Non-Beacon, donde la capa MAC hace uso del método **CSMA/CA** [82] no ranurado para acceder a los canales de forma seguro y evitar colisiones.
- Modo Beacon, donde las TRAMAS BEACON son periódicamente enviadas por el Coordinador de la red para sincronizar a todos los nodos asociados a él. En este modo, la capa MAC basa su funcionamiento en un **CSMA/CD** [82] ranurado. En cada TRAMA BEACON, se delimita el principio de la SUPERTRAMA, definiendo un intervalo de tiempo en el que las tramas se intercambian entre los nodos de la red. Este modo ofrece cierta garantía de servicio para aquellos nodos que necesiten un mayor tiempo de envío de información, a costa de limitar el número máximo de nodos que podrán pertenecer a la red, presentando además problemas de retardo y difusión en las TRAMAS BEACON.

Tipos de Nodos y de Redes

El estándar define dos tipos de dispositivos: de funcionalidad reducida **RFD** y de funcionalidad completa **FFD**. Los RFDs son dispositivos muy sencillos con recursos muy limitados, por ello solo pueden comunicarse con FFDs y no entre ellos. Por otra parte, los FFDs son dispositivos que tienen la capacidad de funcionar como *Coordinador* de una **PAN** o como nodo normal. Los FFDs pueden comunicarse con cualquier otro nodo, lo cual les permite también enrutar paquetes.

Basándose en estos tipos de dispositivos, las redes basadas en el estándar **IEEE 802.15.4** pueden adoptar dos formas o **TOPOLOGÍAS** diferentes: la **TOPOLOGÍA EN ESTRELLA** y la **TOPOLOGÍA PUNTO-A-PUNTO**, tal y como se puede observar en la Figura 3.5.

3.2 ALGORITMOS DE CAPA DE RED

Esta sección presenta una amplia revisión de los diferentes tipos de algoritmos de enrutamiento existentes a nivel de capa de red (NWK). Tal y como se puede apreciar en la Figura 3.1, esta capa se sitúa inmediatamente por encima de la capa MAC y provee de una infraestructura de red a las capas superiores de la pila de comunicaciones. Esta infraestructura se basa, generalmente en algoritmos de enrutamiento, los cuales expanden las características definidas por los estándares a nivel PHY y MAC, habilitando la construcción de topologías y métodos de envío más complejos.

Se llama *Protocolo de enrutamiento* al conjunto formado por al menos las capas física, de enlace y de red, aunque puede incorporar a otras capas de nivel superior. Las capas PHY y MAC son definidas por los estándares, como el IEEE 802.15.4, mientras que las capas de red (NWK) se basan en algoritmos de enrutamiento. Este conjunto que incluye a las tres capas de menor nivel, es el conjunto mínimo que permite configurar un protocolo de enrutamiento.

Debido al importante papel que juegan los algoritmos de enrutamiento, existen diferentes metodologías como el multicast [23] [76], las ad-hoc [32] [52], las de malla [83] [35], las basadas en grafos [64] [66] o las basadas en clustering [85] [42]. A continuación se hace un análisis detallado de los principales algoritmos de enrutamiento existentes hoy en día.

3.2.1 Algoritmos de Enrutamiento

A día de hoy existen una amplia variedad de algoritmos de enruteamiento, cada uno ligado a unas necesidades muy específicas. Los algoritmos de enruteamiento de capa de red se han desarrollado de una forma amplia, y durante años los investigadores se han dedicado a desarrollar nuevos y diferentes algoritmos que se centraban en la mejora de algún parámetro específico. Por ejemplo el trabajo de J. H. Chang and L. Tassiulas [50] en el que se explora una mejora en la duración global de la red, a través de la mejora del tiempo de vida medio de los nodos. Otro trabajo es el C. F. Huang and Y. C. Tseng [29], el cual se centra en el análisis del problema de la cobertura de la red y en como maximizarla, asumiendo que todos los nodos en una red se encuentran bajo un determinado grado de cobertura.

Dentro de la gran variedad de algoritmos de enruteamiento existentes, los hay que centran sus esfuerzos en encontrar formas alternativas de búsqueda de rutas y que son conocidos como algoritmos basados en

ACO: Del inglés, ANT COLONY OPTIMIZATION ALGORITHM

AODV: Del inglés, AD-HOC ON DEMAND DISTANCE VECTOR

RPL: Del inglés, IPv6 ROUTING PROTOCOL FOR LOW-POWER AND LOSSY NETWORKS

RREQ: Del inglés, ROUTE REQUEST
RREP: Del inglés, ROUTE REPLY

técnicas de enrutamiento inteligentes, como por ejemplo el algoritmo de enrutamiento BAR, *Basic ant routing* de Camilo et al. [28], donde se presenta un algoritmo de enrutamiento basado en el algoritmo de optimización de colonia de hormigas (**ACO**) o el algoritmo de enrutamiento FMO, *Fuzzy multi-objective routing algorithm* propuesto por Minhas et al. en [63]. En este área hay más algoritmos de enrutamiento como GA-Routing propuesto por Islam and Hussain en [68] y que se basa en algoritmos genéticos o *Sensor Intelligence Routing* (SIR) de Barbancho et al. [27] basado en redes neuronales.

Como se ha comentado, existen una amplia variedad de algoritmos de enrutamiento, basados en múltiples estrategias, pero a nivel comercial y a nivel científico, dos algoritmos destacan por encima del resto, estos han sido utilizados además como base de muchos otros, hablamos de **AODV** y **RPL**. Además de su extendido uso, estos algoritmos resultan de especial interés en escenarios como las *Ciudades Inteligentes* o el *Internet de las Cosas*. Como K. Machado et al. describen en su trabajo [61], los algoritmos de enrutamiento para las *Ciudades Inteligentes* o el *Internet de las Cosas* deben considerar diferentes esquemas de tráfico como el envío de *un nodo a varios*, de *varios a uno* o de *varios a varios*. Estos esquemas de tráfico provocan que los escenarios ya comentados sean muy dinámicos, obligando a que los algoritmos de enrutamiento utilizados también lo sean. En este sentido, *AODV* y *RPL* presentan un buen desempeño en términos de dinamismo y tolerancia a los cambios, es por ello y por su enorme cuota de uso, que serán tomados como referencia en lo que respecta a los análisis realizados en esta Tesis Doctoral. A continuación se detallan ambos algoritmos en profundidad.

AODV

El algoritmo *AODV* presenta un esquema de creación de las rutas basado en grafos de distancia, buscando obtener una ruta hacia un nodo objetivo, requiriendo un mínimo de saltos, lo cual no asegura ni la eficiencia energética, ni el tiempo de convergencia ni el mantenimiento de un reducido número de mensajes de control enviados.

AODV es un algoritmo de enrutamiento que se basa en el descubrimiento de las rutas a la vez que envía mensajes, asegurando que no existan bucles y al mismo tiempo, tratando de encontrar la ruta más corta en base al número de saltos. *AODV* hace uso de dos tipos de mensajes especiales para la creación de las rutas, la petición de ruta o **RREQ** y la respuesta de ruta **RREP**. En *AODV* las rutas se forman a partir de un nodo tomado como origen, el cual emite de forma global un mensaje RREQ buscando un nodo destino específico, como se puede ver en la Figura 3.6.

Sus nodos cercanos escucharan el mensaje RREQ y buscarán en sus tablas de enrutamiento a dicho nodo destino, si no lo encuentran continuarán reenviando el mensaje RREQ original hasta que algún nodo tenga en su tabla de enrutamiento al nodo destino. Una vez el nodo

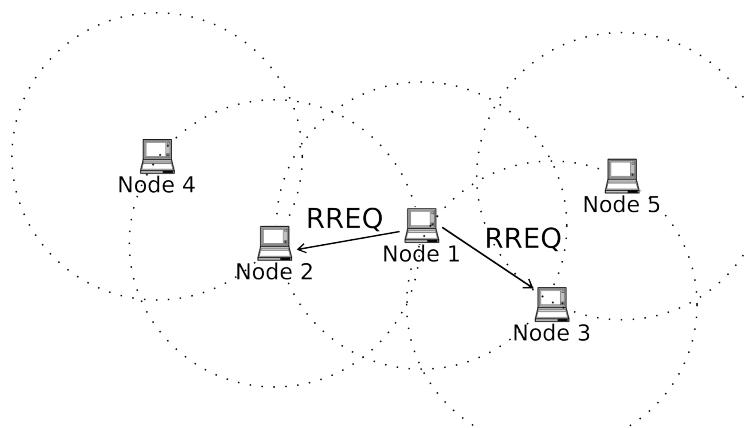


Figura 3.6
Ejemplo de envío de mensajes RREQ en AODV.

destino ha sido encontrado, el nodo que lo encontró le reenviará el RREQ en exclusiva a él. El nodo destino, procesará el mensaje RREQ y responderá con un mensaje RREP que será enruteado hasta el nodo origen, como muestra la Figura 3.7, dando por terminado el proceso de creación de rutas. Como se puede apreciar, el método para crear rutas usado por AODV es un método ejecutado bajo demanda y debe contar con un mínimo número de saltos, lo cual no asegura ni la eficiencia energética, ni otros aspectos de interés en términos de dinamismo de una red. A priori puede parecer un esquema eficiente en términos de mensajes de control enviados, pero como se podrá apreciar en la sección 6, presenta ciertas deficiencias que lo hacen poco eficiente.

Otro punto interesante de AODV es el hecho de que solo almacena una ruta para un nodo destino, lo cual conlleva asociados penalizaciones de tiempo y envío de mensajes en caso de error o de que la ruta quede desactualizada, por el hecho de tener que descubrir una nueva ruta para ese destino. La inestabilidad de las rutas, especialmente si son de gran tamaño, es otro punto importante a considerar en AODV, como Cheng R. H. et al comentan en su trabajo [30], esto se produce debido a que las rutas establecidas por AODV permanecen estáticas y sin cambios a menos que los enlaces fallen. Debido a la alta frecuencia con

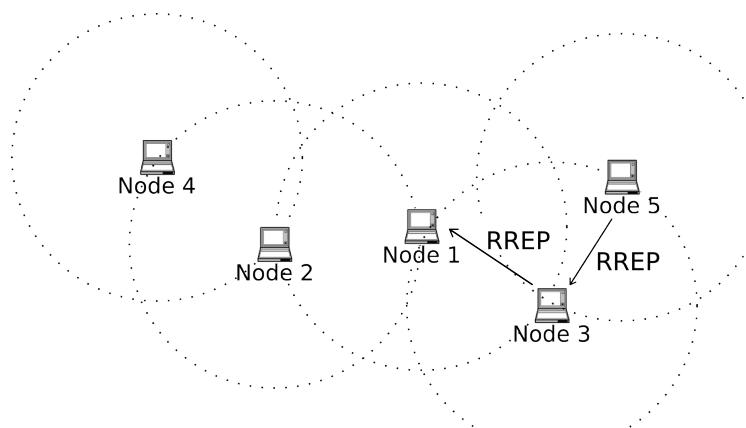


Figura 3.7
[Ejemplo de envío de mensajes de respuesta RREP en AODV.]

que las topologías y el entorno cambian en escenario dinámicos, como por ejemplo *Ciudades Inteligentes*, pueden ocurrir dos alternativas. La primera que algunos nodos intermedios establezcan una nueva ruta de forma rápida, resultando en un mínimo contratiempo. La segunda alternativa es que la nueva ruta dependa de muchos saltos y sea inestable, desembocando nuevamente en una pérdida de la ruta debido a fallos en los enlaces entre los nodos intermedios. Por ello, ante la falta de un mecanismo que asegure la eficiencia energética y la eficiencia en términos de tiempo de convergencia y número de mensajes de control enviados, AODV presenta un exceso en el consumo de recursos, lo cual no es recomendable en el área de los sistemas empotrados alimentados por batería y aplicados a escenarios dinámicos.

RPL

El segundo algoritmo comentado en esta sección es RPL. En este algoritmo las rutas también se basan en grafos acíclicos dirigidos o **DODAGs**. Estos grafos toman en consideración no sólo a los nodos, sino también a los enlaces entre los nodos, lo cual permite formar la ruta desde el nodo coordinador de la red, básicamente el nodo sumidero o el nodo coordinador de la red. Aunque su amplio estudio ha demostrado un buen desempeño en diferentes esquemas de transmisión, RPL fue diseñado y optimizado para asegurar las transmisiones de datos entre los nodos y el coordinador de la red, siguiente un esquema *varios a uno*.

Para la construcción y el mantenimiento de un grafo DODAG, los nodos RPL realizan un **ENVÍO MÚLTIPLE** del mensaje **DIO** periódicamente y de forma local. Este mensaje contiene información que permite a los propios nodos descubrir la existencia de grafos DODAG en su área geográfica, junto con su información de configuración. Para que un nodo pueda unirse a un determinado grafo DODAG, este debe recibir al menos un mensaje **DIO** por parte de uno de sus nodos vecinos. Para realizar la conexión a un grafo, y con ello la conexión a la red, RPL ofrece dos métodos diferentes. Un nodo puede esperar la recepción de un mensaje **DIO**, como ya se ha comentado previamente, o puede enviar un mensaje **DIS** para requerir la transmisión inmediata de un mensaje **DIO**. Una vez que un nodo ha enviado un mensaje **DIS**, continuará enviando mensajes **DIS** hasta que reciba un mensaje **DIO** como respuesta. Si no se recibe ningún mensaje **DIO** después de un determinado tiempo, el nodo puede decidir convertirse en el coordinador de un nuevo grafo DODAG. RPL soporta también la creación de rutas descendentes, es decir, de un nodo hacia el coordinador, mediante el uso de mensajes **DAO**, los cuales son generados por nodos que no son coordinador y enviados hacia el coordinador de un grafo DODAG, para anunciarse como posibles nodos destino.

Como se puede observar, RPL presenta como en el caso de AODV un excesivo uso de recursos además de, a priori, una baja eficiencia en términos de tiempo de convergencia y número de mensajes de control

DODAGs: Del inglés,
DESTINATION
ORIENTED DIRECTED
ACYCLIC GRAPHS

ENVÍO MÚLTIPLE: Del
inglés, MULTICAST
DIO: Del inglés,
DODAG INFORMATION
OBJECT

DIS: Del inglés,
DODAG INFORMATION
SOLICITATION

DAO: Del inglés,
DESTINATION
ADVERTISEMENT
OBJECT

enviados. En este sentido H. Kermajani y C. Gomez en [41] analizaron el rendimiento de *RPL*, optimizando el método de formación de los grafos *DODAG* a través del uso de mecanismos de redundancia, lo cual les permitió optimizar *RPL*, de ahora en adelante *O-RPL*, y mejorar el tiempo de convergencia de las redes en escenarios de alta densidad.

3.2.2 Algoritmos de Enrutamiento basados en Clustering

Los algoritmos de enrutamiento basados en clustering [74] [77], se basan generalmente en la optimización de diferentes parámetros como pueden ser la calidad del servicio *QoS*, la minimización del consumo energético, la reducción del tráfico o la maximización del rango de alcance [88]. En base a estos objetivos de optimización tan específicos, se ha considerado que el algoritmo de enrutamiento propuesto en esta Tesis Doctoral, haga uso de técnicas de clustering. Dentro de esta categoría de algoritmos de enrutamiento hay principalmente una topología, debido a su uso y configuración, el **ÁRBOL JERÁRQUICO BASADO EN CLÚSTERES**. Por supuesto, además de esta topología existen otras opciones dentro de los algoritmos de clustering, como son las propuestas basadas en algoritmos de negociación como la propuesta de Shun-Sheng W. et al [78] o los árboles redundantes de Saravanan M. et al en [59].

QoS: Del inglés,
QUALITY OF SERVICE

**ÁRBOL JERÁRQUICO
BASADO EN
CLÚSTERES:** Del
inglés, *HIERARCHICAL
CLUSTER-TREE*

Habitualmente los algoritmos basados en árboles jerárquicos tienen nodos con diferentes roles o funcionalidades, conocidos como cabezas de clúster y no-cabezas de clúster. Un clúster debe estar formado por al menos una cabeza de clúster y uno o más nodos no-cabeza de clúster. Los nodos del clúster se comunican entre ellos (comunicación sensor-to-sensor) y principalmente con la cabeza del clúster (comunicación intra-clúster), el cual se encarga de realizar las comunicaciones entre distintos clústeres (comunicación inter-clúster). En base a estos roles y a la necesidad que tienen los algoritmos basados en clustering de elegir uno o más nodos cabeza de clúster, se realiza un análisis a continuación de los algoritmos para la elección de una cabeza de clúster más utilizados.

Algoritmos de elección de la cabeza de un clúster

Los algoritmos de elección de la cabeza de un clúster son un área de investigación por sí mismos en el que destaca el hecho de que siempre utilizan al menos un parámetro como estimador base. Hay algoritmos que utilizan la localización geográfica [67], la energía restante en la batería [56] [89] o la calidad de los enlaces inalámbricos entre los nodos [49]. Un ejemplo es el trabajo de M. Jiasong et al. [51], donde describen una optimización de enrutamiento adaptativa basada en un algoritmo de balanceo de la energía para redes de clustering jerárquico en *ZigBee*. En ese trabajo, M. Jiasong et al. limitan el número de saltos dependiendo de la energía disponible en la batería, limitando el rango de alcance

de un nodo, de forma que los clústeres se configuran dinámicamente. Otro ejemplo es el algoritmo *MutihopLQI* [34], el cual crea un árbol con múltiples saltos de forma dinámica, considerando dos parámetros, *MinLQI* y *MaxLQI*, cuyos valores representan la calidad de los enlaces de radio y mediante los cuales analiza el impacto del *LQI* en la formación de las rutas, eligiendo de esta forma las cabezas de los clústeres. Mittal N. et al [67] también proponen un algoritmo basado en la distancia y la energía residual en los nodos, centrándose en la maximización del tiempo de vida de la red para seleccionar las cabezas de los clústeres. El trabajo de Sundareswaran P. et al [80] también se basa en la selección de la cabeza de clúster, aunque en este caso consiste en un algoritmo para una distribución equitativa de las cabezas de los clústeres. Este algoritmo se basa en el coste de los enlaces para realizar una distribución pseudo-aleatoria de las cabezas de clúster.

A partir de la revisión de la literatura hecha, se puede concluir que los algoritmos de elección de la cabeza del clúster presentan un claro problema. A mayor complejidad, mayor número de mensajes de control para intercambiar información, dependiendo del estimador que utilicen. Al aumentar el número de mensajes de control enviados y la necesidad de análisis, y en muchos casos también aumentará la necesidad de sincronizarse. Este problema conlleva que todos los algoritmos de enrutamiento que hacen uso de complejos algoritmos de elección de la cabeza del clúster, conlleven asociados un elevado tiempo de convergencia y un alto número de mensajes de control enviados. Esto resulta además en algoritmos estáticos y rígidos, que no presentan un buen desempeño en escenarios grandes dinámicos como las *Ciudades Inteligentes* o el *Internet de las Cosas*.

3.3 DINAMISMO EN UNA WSN

A lo largo de esta Tesis Doctoral se han ido presentando diversos estudios científicos que ponen de manifiesto la necesidad de un nuevo enfoque a la hora de analizar el dinamismo en las WSN. Dicho enfoque debe basarse en el análisis de la capacidad de adaptación de una red ante cambios en el entorno, lo cuál puede ser alcanzado mediante la optimización tanto del tiempo de convergencia como del número de mensajes de control enviados. Para dar un buen ejemplo de la importancia que tiene la adaptabilidad en las WSN modernas, hagamos un ejercicio práctico y supongamos una serie de nodos encargados de la gestión del alumbrado público en cualquier calle del centro de una ciudad. En el momento en que el tráfico haga acto de presencia las conexiones empezarán a tener interferencias o su calidad se degradará rápidamente, volviendo a mejorar con la ausencia de tráfico. Sería inconcebible un alumbrado público que no funcionara en presencia de tráfico rodado, al igual que resulta inconcebible que una determinada aplicación deje de funcionar por los problemas derivados de sus comunicaciones. Es por este motivo, que la red debe ser capaz de adaptarse

de forma ágil y rápida ante cualquier eventualidad ya sea provocada por factores internos o externos a la red.

El tiempo de convergencia puede ser utilizado con el fin de minimizar el tiempo de respuesta ante problemas que impliquen desconexión parcial o total de nodos de la red. Para ello, es posible el desarrollo de un algoritmo de enrutamiento que permita la organización de las redes en grupos, los cuales acepten la conexión de otros nodos de forma desatendida, dando lugar a múltiples puntos de conexión en paralelo, lo cual mejorará no solo el tiempo de convergencia sino también el consumo energético.

El número de mensajes de control enviados está estrechamente relacionado con el propio tiempo de convergencia y con el consumo energético. A un menor número de mensajes de control, menor tiempo de convergencia y menor consumo energético. Si un protocolo permite actuar frente a eventualidades con un mínimo número de mensajes enviados, esto repercute en un menor tiempo de uso del transmisor de radio, y por extensión, en un menor tiempo de actividad, lo cual permite mantener al microcontrolador principal y a la radio en un estado de bajo consumo energético.

Los algoritmos de enrutamiento de tipo *clustering*, son ampliamente utilizados ya que permiten la conexión de diferentes nodos a la vez en diferentes clústeres de la red, dando lugar a un menor tiempo de convergencia. Sin embargo, los algoritmos de clustering suelen requerir de una infraestructura y un soporte a la misma, lo cual repercute directamente en un mayor número de mensajes de control enviados.

Tal y como se explicó en la Sección 2.4.2, las WSN que forman parte de los proyectos de *Ciudades Inteligentes* o *Internet de las Cosas*, son redes grandes de media o baja densidad que están sujetas a entornos cambiantes en los que será necesarios adaptarse de forma ágil y rápida, buscando por ello mantener un bajo tiempo de convergencia, así como un bajo y constante número de mensajes de control enviados.

3.4 CONCLUSIONES

Una WSN, está formada por una serie de nodos que interactúan con el mundo físico y se comunican entre ellos mediante comunicaciones inalámbricas. El principal estándar de comunicaciones para WSN es IEEE 802.15.4, el cual ofrece comunicaciones inalámbricas de bajo consumo y bajo coste. El estándar define dos capas, la física (PHY) y la de acceso al medio (MAC). Usando como base este estándar, numerosos algoritmos de enrutamiento a nivel de capa de red se han desarrollado.

Basándose en la revisión acerca de los algoritmos de enrutamiento realizada en este capítulo, es posible identificar en los algoritmos de enrutamiento actuales una carencia en términos de eficiencia del *Tiempo de*

Convergencia y minimización del Número de mensajes de control enviados en la fase de formación de la red. Esta propuesta se centra en escenarios grandes del tipo *Ciudad Inteligente* o *Internet de las Cosas*, considerando estos grandes escenarios en lo relativo a su infraestructura de soporte como escenarios de densidad media/baja.

Con este objetivo en mente y en base a los conocimientos presentados en este capítulo, se llevará a cabo la demostración de la HIPÓTESIS DE TRABAJO de esta Tesis Doctoral sobre un modelo basado en simulaciones, tal y como se recoge en el Capítulo 4.

Parte IV

MÉTODO Y MATERIALES

4

PROTOCOLO DARP

ÍNDICE

4.1	DARP: Dynamic and Adaptive Routing Protocol	53
4.1.1	DARAL: Dynamic and Adaptive Routing ALgorithm	54
4.1.2	Cabeceras y tipos de mensajes	56
4.1.3	DRSP: Dynamic Role Selection Process	59
4.1.4	Sub-redes Virtuales	62
4.2	Conclusiones	64

Las WSN se caracterizan por ser redes basadas en hardware económico, pero esta idea es difícil de combinar cuando, por muy económico que sea el hardware, se requieren miles o millones de dispositivos. Es por ello que los simuladores de WSN están convirtiéndose en un elemento de gran interés en el estudio y la utilización de las redes. Estos simuladores permiten tanto la simulación del comportamiento de las redes en entornos dinámicos o estáticos, así como la simulación de los despliegues, validando topologías, arquitecturas y el impacto que en estas puedan suponer los eventos adversos.

En este capítulo, se llevará a cabo una profunda descripción de las propuestas de protocolo y de algoritmos que se presentan en esta Tesis Doctoral. También se presenta la plataforma de simulación donde se han desarrollado y evaluado los modelos de algoritmo y protocolo desarrollados, así como todos los parámetros de configuración que puedan resultar necesarios.

4.1 DARP: DYNAMIC AND ADAPTIVE ROUTING PROTOCOL

El protocolo de enrutamiento dinámico y adaptativo **DARP** [39], es una propuesta de protocolo de enrutamiento multi-capa resultante de la combinación de las diferentes capas que componen una pila de comunicaciones. En la Figura 4.1 se ilustra el caso de **DARP**, el cual se basa en las capas PHY y MAC del estándar IEEE 802.15.4 y utiliza a nivel de capa de red el algoritmo de enrutamiento dinámico y adaptativo **DARAL**, el cual se basa en clustering, y que a su vez hace uso de un proceso de selección de rol dinámico **DRSP**. Ambos algoritmos se describen en profundidad a lo largo de esta sección.

DARP: Del inglés,
DYNAMIC AND
ADAPTIVE ROUTING
PROTOCOL

DARAL: Del inglés,
DYNAMIC AND
ADAPTIVE ROUTING
ALGORITHM

DRSP: Del inglés,
DYNAMIC ROLE
SELECTION PROCESS

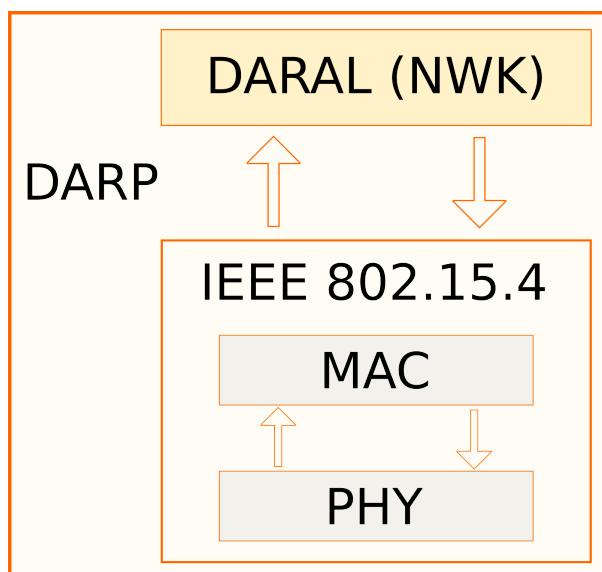
4.1.1 DARAL: Dynamic and Adaptive Routing ALgorithm

DARAL es un algoritmo de enrutamiento basado en redes IEEE 802.15.4 de tipo *non-beaconing*, que ofrece un clustering dinámico en base al algoritmo DRSP, el cual hace uso de la calidad del enlace entre nodos para llevar a cabo las tareas de selección de clúster. DARAL se ha diseñado con el objetivo de dar soporte a grandes escenarios donde la arquitectura de comunicaciones tenga una densidad media o incluso baja, es decir, grandes escenarios en los que la formación de las redes resulta tediosa. Es evidente que en el caso de las *Ciudades Inteligentes*, más que dificultades por distancia o localización, aparecerán también en forma de tráfico. Por ello se ha diseñado una propuesta basada en clustering, la cual permite concentrar las comunicaciones que se produzcan en determinadas áreas. Estas técnicas hacen uso del *Principio de Localidad Espacial*, el cual se basa en concentrar elementos, en este caso nodos de la red, que se encuentran en un área cercana, geográficamente hablando. Gracias al uso del *Principio de Localidad Espacial*, DARAL permite la construcción de redes basadas en clústeres en entornos complejos.

Antes de continuar con una descripción más detallada conviene aclarar algunos conceptos generales acerca de DARAL, como es el propio concepto de sub-red virtual, así como los diferentes tipos de roles disponibles en el algoritmo. DARAL no se limita a ofrecer únicamente técnicas de *clustering*, sino que las lleva un paso más allá, dotando a estos clústeres de ciertas propiedades que les permiten actuar como sub-redes virtuales, lo cual permite a su vez, que estas sub-redes trabajen de forma paralela y autónoma. Este algoritmo trata de ser eficiente en términos de tiempo de convergencia y número de mensajes de control enviados, además de maximizar la cobertura e intentar minimizar el consumo de energía de las redes en grandes escenarios de baja o media densidad.

DRSP: SE DESCRIBE EN LA SECCIÓN 4.1.3

Figura 4.1
Pila de Comunicaciones del protocolo DARP.



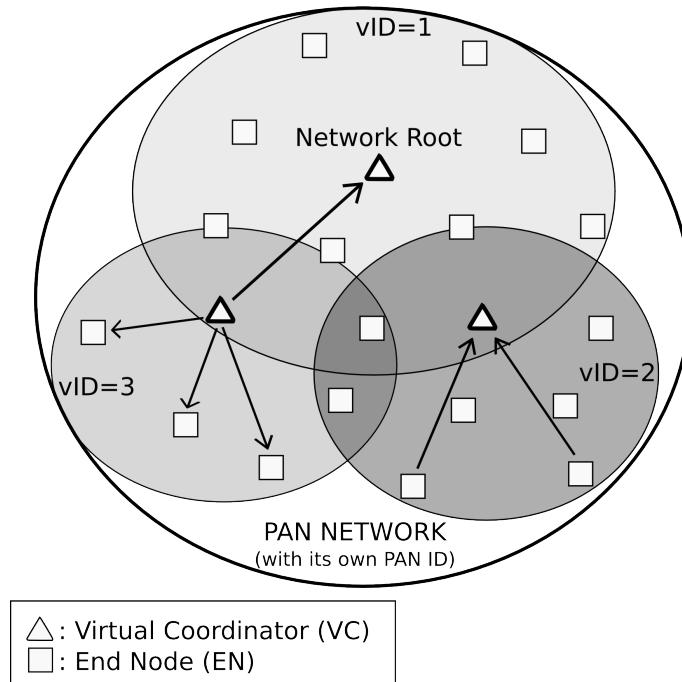


Figura 4.2
Ejemplo de red PAN basada en DARAL

Basándose en todas estas características se considera que el *clustering* es la opción más adecuada, debido a que permite una rápida respuesta por parte de los clústeres a los nodos que necesiten conectarse a la red, además de minimizar el tráfico debido a la concentración en dicho clústeres. La aplicación de las sub-redes virtuales se hace en base a un identificador de sub-red virtual (vID), el cual se utiliza en las tareas de enrutamiento y permite un eficaz procesamiento del paquete. La Figura 4.2 muestra un ejemplo de red PAN IEEE 802.15.4 con su propio *Coordinador* e identificador. A su vez, la red PAN está dividida en tres sub-redes virtuales cada una con su propia cabeza de clúster y su propio identificador.

La Figura 4.2 muestra otro elemento clave en las redes basadas en DARAL, los roles de los nodos. Los nodos deben ser iguales en términos de hardware, de forma que sea el propio algoritmo de enrutamiento el que pueda seleccionar el rol de los nodos, por medio de la ejecución de DRSP al iniciar el nodo. DARAL sigue la filosofía del estándar IEEE 802.15.4 definiendo dos roles. El primero es el conocido como *Nodo Final (EN)*, este nodo coincide con la especificación RFD del estándar, tanto en cuanto da al nodo la capacidad de comunicarse exclusivamente con la cabeza de su clúster. El segundo rol es la propia cabeza del clúster, llamado en DARAL como *Coordinador Virtual (VC)*. Los VCs siguen la especificación FFD del estándar y se encargan de las tareas de creación y gestión de las sub-redes virtuales, así como de las tareas de enrutamiento entre sub-redes.

EN: Del inglés, END NODE

VC: Del inglés, VIRTUAL COORDINATOR

Tabla 4.1
Cabecera de Control
en DARP.

Offset	Campo	Tamaño
0	OPERATION CODE	1B
1	PACKET LENGTH	1B
2	ROUTING TYPE	1B
3	HOP LIMIT	1B
4	CHECKSUM	2B
6	MESSAGE IDENTIFIER	1B
7	SOURCE VIRTUAL NETWORK IDENTIFIER	2B
9	DESTINATION VIRTUAL NETWORK IDENTIFIER	2B
11	SOURCE ADDRESS	8B
19	DESTINATION ADDRESS	8B

4.1.2 Cabeceras y tipos de mensajes

Resulta de interés describir cómo funciona DARAL, presentando los diferentes tipos de mensajes y describiendo las cabeceras de control que utiliza el algoritmo de enrutamiento para direccionar e interpretar los paquetes. Los paquetes de datos, también conocidos como tramas, son generados en la capa más alta (APL), y van descendiendo a lo largo de la pila de comunicaciones. En cada capa se encapsula el paquete de la capa superior como **CARGA ÚTIL** y se añaden las cabeceras correspondientes y necesarias de esa capa. La Tabla 4.1 muestra la posición de la cabecera DARAL, la descripción y el tamaño asociado a cada campo en Bytes.

CARGA ÚTIL: *Del inglés, PAYLOAD*

La cabecera DARAL se ubica justo a continuación de la cabecera MAC y tiene un tamaño de 27 Bytes distribuidos como sigue:

- Código de operación (Operation Code): Este campo se utiliza para identificar el tipo de mensaje de control enviado.
- Longitud del paquete (Packet Length): Indica la longitud del paquete a nivel de capa de red.
- Tipo de enrutamiento (Routing Type): Identifica el tipo de enruteamiento usado.
- Límite de saltos (Hop Limit): Un límite de saltos, que por ahora no se utiliza.
- Suma de comprobación (Checksum): Suma de control de errores.

- Identificador del mensaje (Message ID): Este identificador se usa para prevenir problemas de control de flujo, así como la recepción de mensajes duplicados.
- Identificador de sub-red virtual origen (Source Virtual Network ID): Este identificador se utiliza en las tareas de enrutamiento para conocer la sub-red origen de la que vino el mensaje de forma que pueda ser respondido.
- Identificador de sub-red virtual destino (Destination Virtual Network ID): Este identificador se utiliza para las tareas de enrutamiento de forma que permita direccionar el mensaje hacia la sub-red adecuada.
- Dirección Origen (Source Address): Dirección de red del nodo que generó el mensaje.
- Dirección Destino (Destination Address): Dirección de red del nodo hacia el que se destina el mensaje.

Dentro de estos campos destacan dos, el *Código de operación* y el *Tipo de Enrutamiento*. El *Código de Operación* permite seleccionar entre los quince tipos de mensajes disponibles en *DARAL*, entre los que se encuentran los asociados a conexión o borrado de nodos o tareas de enrutamiento. A continuación se presentan los diferentes mensajes:

- ASSOCIATION_REQ: Cada nodo difunde este mensaje a todos los nodos a los que tenga en su radio de alcance, indicando que requiere una conexión.
- ASSOCIATION REP: Un VC que se reciba un ASSOCIATION_REQ y tenga espacio disponible en su sub-red, responderá con este mensaje, indicándole al nodo que puede establecer conexión con él.
- ASSOCIATION REP ACK: Si el resultado de ejecutar *DRSP* es que el nodo se configure a sí mismo como EN, entonces enviará este mensaje al VC, cerrando la conexión y pasando a estar conectado.
- ASSOCIATION PAN_ID REQ: Si el resultado de ejecutar *DRSP* es que el nodo se configure a sí mismo como VC, entonces enviará este mensaje al VC de la sub-red solicitando un nuevo identificador de sub-red. El nodo pasará a un estado de espera donde aún no podrá enrutar paquetes, pero sí que podrá recibirlas y enviarlas.
- ASSOCIATION PAN_ID REQ ACK: Este mensaje es la respuesta desde el *Coordinador de la Red* a la petición de un nuevo vID (ASSOCIATION PAN_ID REQ) y contiene la información del nodo objetivo actualizada, incluyendo el nuevo vID.

- ASSOCIATION_PAN_ID_ASSIGN: El VC asigna el nuevo vID al nodo VC que depende de él.
- ASSOCIATION_PAN_ID_ASSIGN_ACK: Este mensaje es la respuesta al mensaje *ASSOCIATION_PAN_ID_ASSIGN* y confirma el inicio de la nueva sub-red.
- ASSOCIATION_INFORM: Este mensaje se genera cuando un nuevo nodo VC o EN se conecta a una sub-red. El mensaje se dirige al *Coordinador de la Red* y sirve para que los nodos intermedios actualicen sus tablas de enrutamiento.
- ASSOCIATION_INFORM_ACK: Este mensaje es la respuesta del *Coordinador de la Red* al mensaje *ASSOCIATION_INFORM*.
- KEEP_ALIVE_REQ: Este mensaje es generado por un VC para controlar el estado de los nodos conectados a su sub-red.
- KEEP_ALIVE_REQ_ACK: Un mensaje de respuesta al mensaje *KEEP_ALIVE_REQ*.
- PURGE_REQ: Un VC genera este mensaje para el *Coordinador de la Red* cuando detecta que un nodo en su sub-red está caído.
- PURGE_REQ_ACK: Este mensaje es la respuesta del *Coordinador de la Red* a una petición del tipo *PURGE_REQ*. Este mensaje se utiliza para actualizar las tablas de enrutamiento una vez un nodo deja de estar disponible.
- DATA: Este mensaje se corresponde con el mensaje de datos y de ser necesario, contendrá la carga útil procedente de las capas superiores.
- DATA_ACK: Este mensaje confirma la recepción de un mensaje del tipo *DATA*.

El otro campo de interés en la cabecera de control de *DARAL* es el *Tipo de Enrutamiento*, donde se definen tres tipos de enrutamiento diferentes. El tipo de enrutamiento en combinación con los vIDs son los elementos que permiten enrutar mensajes en este algoritmo:

- Gateway: Representa el envío de un mensaje en dirección hacia el elemento de la red (VC) localizado justamente por encima en la jerarquía y que se puede alcanzar mediante un envío directo.
- Forwarding: Este enrutamiento representa un mensaje descendiendo en la jerarquía de la red, esto es, siendo enrutado hacia un destino. Este tipo de enrutamiento se utiliza hasta alcanzar la sub-red destino. Una vez allí se utiliza el enrutamiento de tipo *Parsing*.

- Parsing: Este tipo de enrutamiento se utiliza para el último envío. Este envío se produce desde el VC de la sub-red destino hasta el nodo destino. Para este último envío se utiliza ya la dirección destino del nodo.

Dado el interés que suponen los tipos de enrutamiento, por su importancia en *DARAL*, las Figuras 4.3 4.4 y 4.5 muestran tres ejemplo de su uso. En el caso 4.3 y 4.4 muestran ejemplos de comunicación inter-clúster donde el nodo EN_{2F} envía un paquete a EN_{3E} en 4.3 y a EN_{1B} en 4.4. EN_{2F} es un EN que solo puede comunicarse en su sub-red con su VC, por lo que no puede conocer la ruta hacia el destino, por ello usa el tipo de enrutamiento *Gateway*. El VC de la sub-red 2 tampoco conoce la ruta hacia EN_{3E}, ni hacia EN_{1B}, por lo que tiene que pre-guntar al VC de orden superior, así que reenvía el mensaje con el tipo de enrutamiento *Gateway*. En este paso ya ha alcanzado al *Coordinador de la Red* que es a su vez VC de la sub-red 1. En 4.4, EN_{1B} pertenece a esta sub-red, por lo que el *Coordinador de la Red*, busca en su tabla de enrutamiento a nivel local, localizando el nodo destino y cambian-do el tipo de enrutamiento a *Parsing*, enviándole el mensaje de forma directa y exclusiva a EN_{1B}. En el caso de 4.3, el *Coordinador de la Red*, encuentra la ruta hacia la sub-red destino, por lo que cambia el tipo de enrutamiento a *Forwarding* y reenvía el mensaje. Una vez que el men-saje alcanza la sub-red destino, el VC de esa sub-red volverá a cambiar el tipo de enrutamiento a *Parsing*, reenviándolo al nodo destino EN_{3E}. En el ejemplo 4.5 se muestra un ejemplo de comunicación intra-clúster, en la que EN_{2F} trata de enviar un mensaje a EN_{2C}. En este caso el VC de la sub-red 2 detecta que el mensaje procede y se dirige a su sub-red, busca en su tabla de enrutamiento local y una vez encuentra el nodo destino, procede a enviarle el mensaje usando el tipo *Parsing*.

4.1.3 DRSP: Dynamic Role Selection Process

Los nodos *DARAL* en su arranque, ejecutan *DRSP*, el proceso de selección de rol dinámico mediante el cual seleccionan su rol en la red. *DRSP* se basa en la calidad del enlace (LQI) entre dos nodos, un es-timador muy utilizado y analizado durante años. En este caso este algoritmo resulta innovador debido a que cambia de lugar la toma de decisión acerca del enlace, recayendo esta tarea en el propio nodo que desea conectarse. No solo el nodo debe decidir su propio rol en la red, sino que además es el propio nodo el que espera un determinado tiempo T_{link} , durante el que recogerá diferentes respuestas por parte de VCs que puedan integrarlo en sus redes. Una vez seleccionado el mejor de los enlaces de entre todos los recibidos, el nodo realizará un muestreo en base a una serie de umbrales que marcarán si se desecha el enlace seleccionado ($TH_{baselevel}$) o si el nodo se configura como VC o EN (TH_{role}). La Figura 4.6 ilustra cómo funciona el proceso. En ella se puede apreciar como el nodo EN – A recibe tres mensajes di-ferentes por parte de los VC de las sub-redes 1, 2 y 3. Cada mensaje recibido tiene un valor de LQI diferente L₁, L₂ y L₃. De esos valores,

Figura 4.3
Ejemplo A de enrutamiento en DARAL.

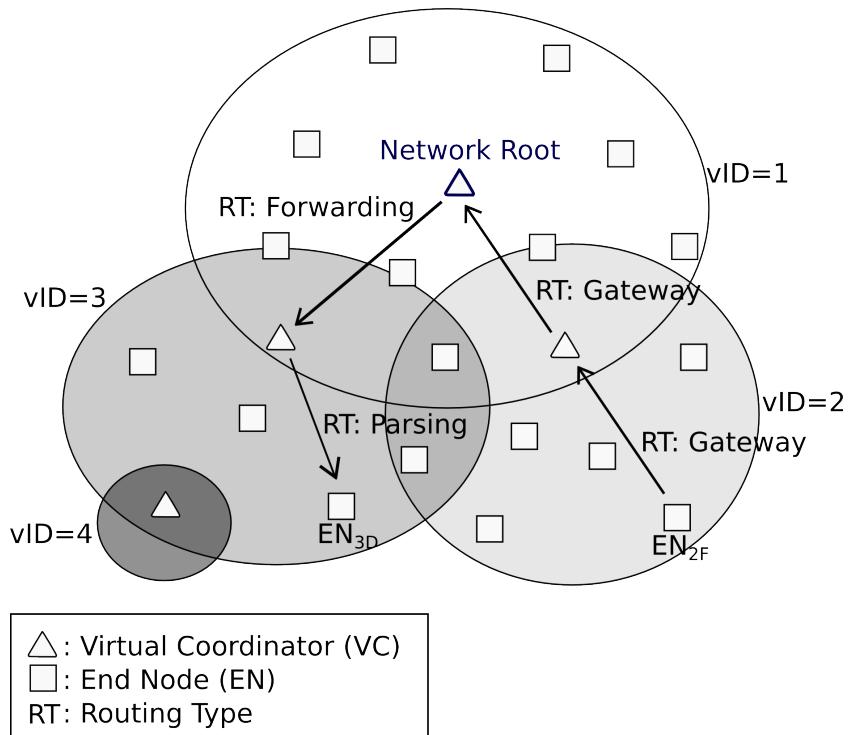
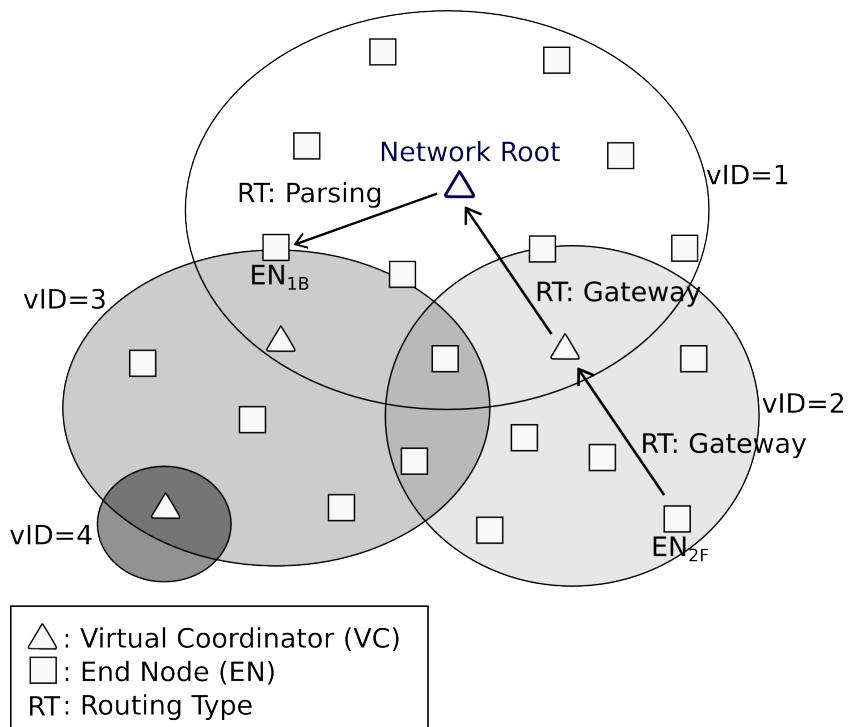
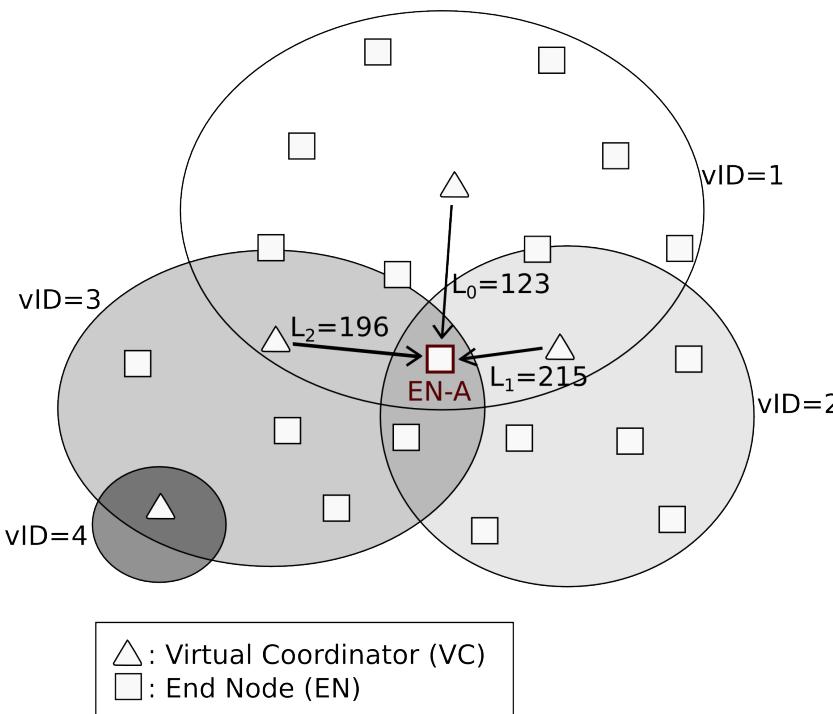
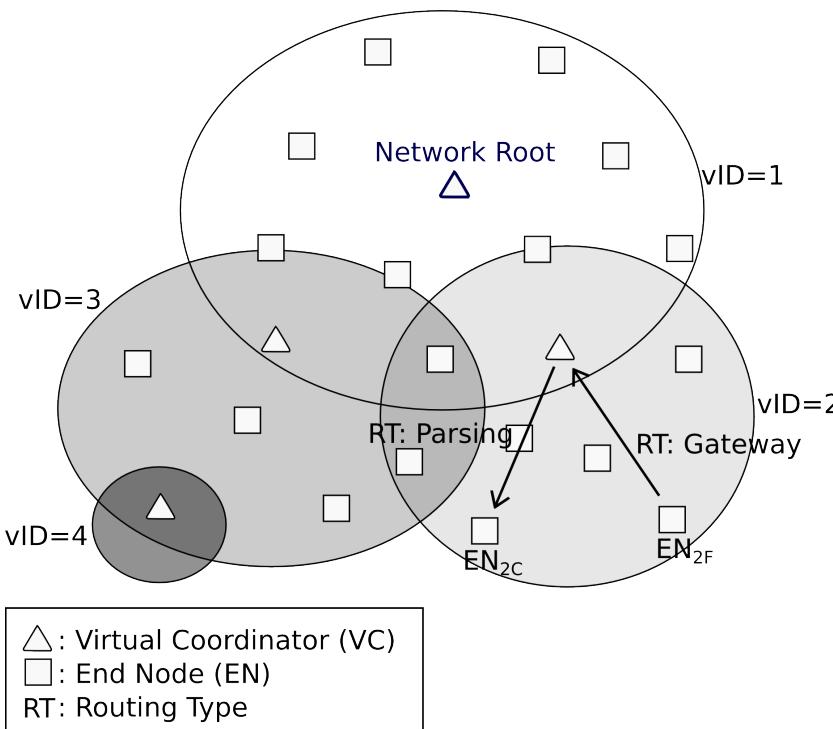


Figura 4.4
Ejemplo B de enrutamiento en DARAL.

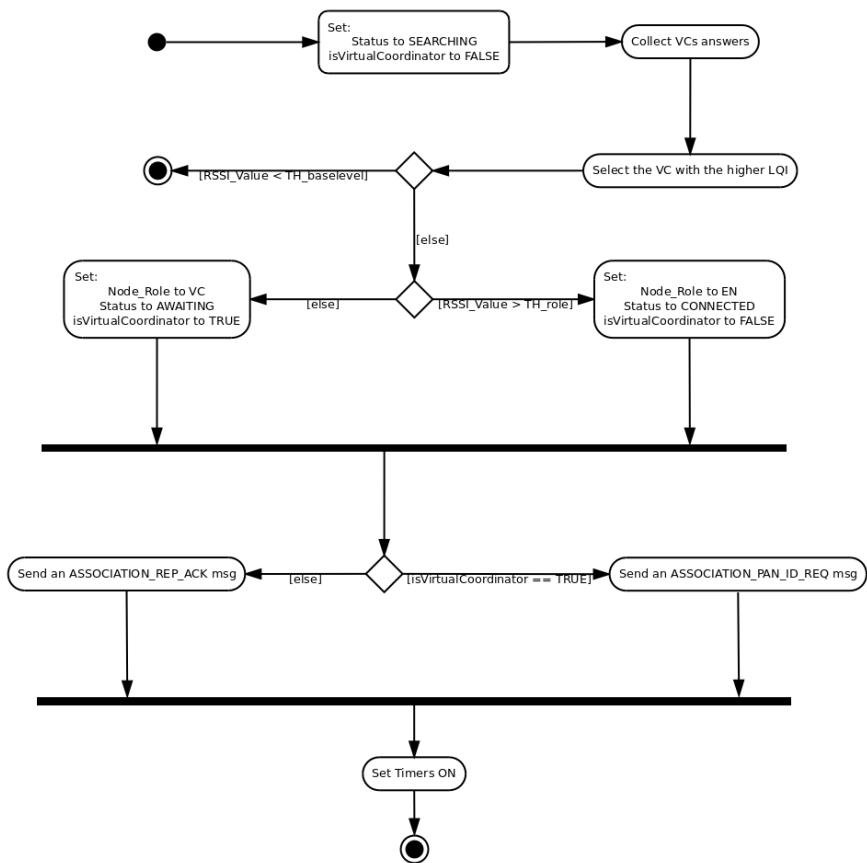


EN – A seleccionará L₁ y se conectará con la sub-red 2, y dependiendo de cómo este ajustado TH_{role} se conectará como EN o VC.



La particularidad de *DRSP* es que sigue el esquema descrito en la Ecuación 4.1, donde los ENs aprovechan la mejor calidad de los enlaces,

Figura 4.7
Representación del algoritmo DRSP en UML.

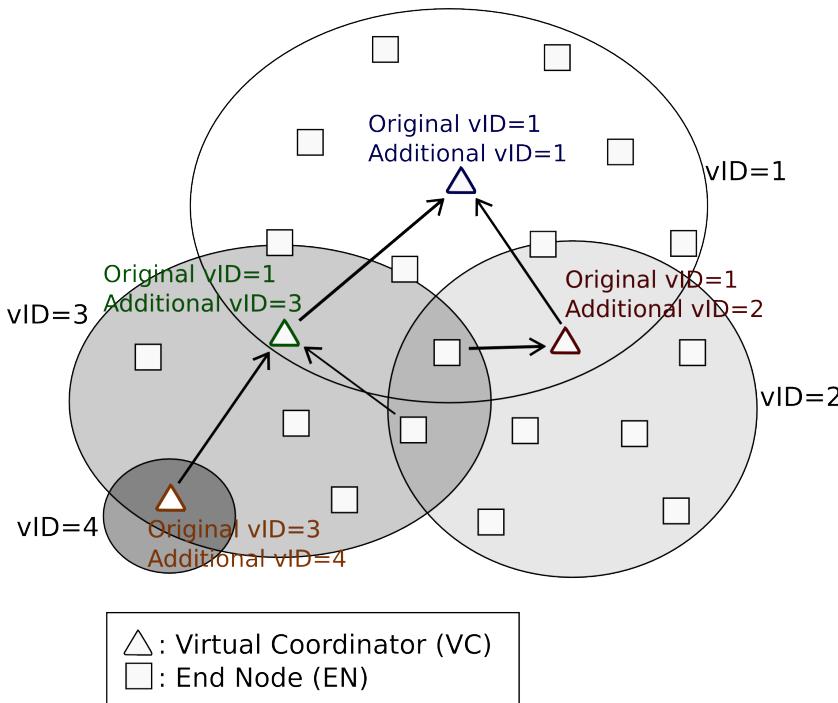


mientras que los VCs se encuentran en una franja periférica alrededor del nodo central. Este esquema permite, por un lado, concentrar los nodos espacialmente más cercanos y que tienden a comunicarse entre ellos, debido a la relación directa que existe entre calidad de enlace y distancia espacial. Por otro lado consigue ir expandiendo el alcance de la red de forma progresiva. Cada nuevo VC genera su propia sub-red, de forma que la red va aumentando sus límites, permitiendo alcanzar progresivamente más nodos.

$$\begin{aligned} \text{EN : } & \text{LQI} > \text{TH}_{\text{role}} \\ \text{VC : } & \text{TH}_{\text{baselevel}} < \text{LQI} < \text{TH}_{\text{role}} \end{aligned} \quad (4.1)$$

4.1.4 Sub-redes Virtuales

Como se ha comentado previamente, *DARAL* se basa en el uso de sub-redes virtuales o clústeres. Cada una de estas sub-redes se encuentra identificada por medio de un identificador de red virtual o vID. Cada



sub-red desarrolla su propio balanceo de carga al agrupar a un número determinado de nodos, el cual viene limitado por el parámetro L_{nodes} . En el momento en que un VC alcanza dicho límite deja de enviar respuestas a las peticiones de asociación de los nodos, negando a partir de este punto, el acceso de nuevos nodos a la sub-red. Dicha denegación de acceso permanecerá activa mientras el VC haya alcanzado o sobrepasado dicho límite.

Los VC juegan una doble función en *DARAL*, debido a la estructura jerarquizada, como muestra la Figura 4.8. Dicha doble función se basa en actuar como ENs en la sub-red origen a la que pertenecen y como VCs en la sub-red que crean y gestionan. En la Figura 4.8, los VCs muestran dos vIDs, *original vID* para representar la función de EN y *additional vID* para representar la vID de la red en la que son VCs. Es por este motivo, los VCs utilizan dos tablas de enrutamiento en *DARAL*, una para el enrutamiento a nivel local dentro de su sub-red y otra para el enrutamiento de paquetes entre diferentes sub-redes de la red. Un detalle importante es el conocimiento de la red que poseen los VCs, ya que estos solo conocen la estructura de la red que depende directamente de ellos, es decir, los niveles inferiores de la sub-red que puedan ser accedidos a partir de ellos. Para una mejor comprensión, la Figura 4.8 presenta las diferentes sub-redes, donde se puede apreciar como el VC de la sub-red 3 solo conocerá a los nodos de su sub-red y cómo enrutar paquetes hacia los nodos de la sub-red 4, que es la que depende de él. El *Coordinador de la Red*, en este caso el VC de la sub-red 1, conocerá toda la red y en el peor de los casos un paquete que vaya a otra rama, tendrá que transcurrir por él.

Centrándonos en los vID, estos solo pueden ser generados por el *Coordinador de la Red*. Una vez un vID es generado, todos los VC que se encuentren en la ruta hasta la vID origen actualizarán sus tablas de enrutamiento, incluyendo esa nueva sub-red. Por otro lado, las sub-redes deben informar a los VC que se encuentran por encima en la jerarquía sobre nuevas conexiones de nodos. Por ello, cuando un nuevo EN se conecta a una sub-red, el VC de la sub-red genera un mensaje del tipo *ASSOCIATION_INFORM*, que sirve para anunciar al EN conectado.

Debido a la estructura y organización de *DARAL*, resulta necesario mantener un control sobre la topología de la red. Como en muchos otros algoritmos de enrutamiento, esto se consigue mediante un mecanismo de control del estado de la red, el cual se realiza pasado un determinado tiempo (T_{alive}). Este proceso de control del estado de la sub-red, es activado por los VC una vez T_{alive} expira. De esa forma se genera una petición de respuesta a los nodos mediante un mensaje *KEEP_ALIVE_REQ*. Los ENs deben responder uno a uno de forma explícita mediante un *KEEP_ALIVE_REQ_ACK*. En el caso de que un nodo no responda durante un tiempo T_{down} , el VC iniciará un proceso de expulsión del nodo, informando hacia los nodos de niveles superiores de que dicho nodo se ha desconectado mediante un mensaje *PURGE_REQ*. Este mensaje conlleva la actualización de las tablas de enrutamiento de todos los nodos, eliminando las entradas correspondientes a ese EN. El control del estado de la red se ha optimizado de forma que cada vez que un EN responde a un mensaje de un VC, el temporizador asociado al control del estado de la conexión, es reiniiciado, requiriendo así un mínimo número de mensajes de control de estado.

Si el problema de conectividad es detectado por los ENs, en vez de por el VC, los ENs esperarán un tiempo $T_{reconnect}$ antes de reiniciarse a sí mismos y recomenzar el proceso de conexión a una sub-red de nuevo. Una vez que un nodo que antes pertenecía a otra sub-red se conecta nuevamente, pero a una sub-red diferente, el mensaje *ASSOCIATION_INFORM* permite a los VCs actualizar las entradas correspondientes a ese nodo en las tablas de enrutamiento, incluso si el nodo no había sido eliminado a través de un *PURGE_REQ*.

4.2 CONCLUSIONES

En este capítulo se han presentado los diferentes algoritmos que componen el protocolo de enrutamiento *DARP*, que constituye la base de esta Tesis Doctoral. Estos algoritmos permiten, de manera dinámica y configurable, adaptar el protocolo de enrutamiento a una gran cantidad de grandes escenarios, a pesar de que los escenarios objetivo son aquellos de gran tamaño con una media o baja densidad, como por ejemplo las *Ciudades Inteligentes*.

Como resumen se muestran las principales aportaciones científicas a partir de la investigación realizada en este capítulo:

1. Protocolo DARP: Se ha propuesto un nuevo protocolo de comunicaciones inalámbricas de sensores diseñado específicamente para proporcionar una respuesta ágil y dinámica en grandes escenarios de media o baja densidad. Este diseño se ha basado en dos algoritmos diferentes:
 1. DARAL: Este algoritmo es la clave del protocolo, proporcionando todos los elementos a nivel de red. *DARAL* se basa en una implementación de árbol jerárquico basado en clustering, implementando sus propias técnicas de balanceo de carga, auto-reparación, control de la topología y de reducción de tráfico. De manera novedosa en el estudio de las WSN se propone un algoritmo que se centre en la optimización del tiempo de convergencia de la red y del número de mensajes de control enviados en el proceso de formación de la red. A través del diseño de *DARAL* se ha observado la posibilidad de implementar mecanismos para faciliten la maximización de la cobertura y el enrutamiento eficiente.
 2. DRSP: Este algoritmo es utilizado por *DARAL* para la creación de las sub-redes y la selección de los nodos que las gestionarán.

Parte V

RESULTADOS EXPERIMENTALES

5

ANÁLISIS DE COMPORTAMIENTO DE DARP

ÍNDICE

5.1	Configuración de las capas PHY y MAC	70
5.1.1	Comunicaciones Inalámbricas	70
5.1.2	Simulador	70
5.1.3	Configuración del estándar IEEE 802.15.4	72
5.2	Metodología de ajuste de los parámetros de DARAL	73
5.3	Análisis de la configuración óptima propuesta	74
5.3.1	Umbral de mínima calidad: $TH_{baselevel}$	74
5.3.2	Umbral de decisión de rol: TH_{role}	77
5.3.3	Límite de nodos por sub-red: L_{nodes}	80
5.3.4	Límite de reintentos: $DARAL_MAX_RETRIES$	83
5.3.5	Temporizador de conexión T_{link}	86
5.3.6	Tiempo de vida de los enlaces T_{alive}	89
5.3.7	Temporizador de desconexión T_{down}	91
5.3.8	Temporizador de reconexión $T_{reconnect}$	94
5.3.9	Tiempo de espera para respuestas T_{ack}	97
5.4	Conclusiones	100

El algoritmo *DARAL* utiliza una serie de parámetros para determinar su funcionamiento. Estos parámetros regulan el comportamiento de *DARAL*, y deben ser optimizados para los escenarios de aplicación, proporcionando de esta manera los mejores resultados. Entre los parámetros más destacables y que más influyen en el comportamiento de la red, se encuentran el umbral de mínima calidad, el umbral de decisión de rol, el tiempo de enlace y el límite de nodos por sub-red.

Dado que el escenario objetivo de la presente Tesis Doctoral son los escenarios basados en grandes infraestructuras que presentan densidades medias o bajas, se hace necesario encontrar una configuración que optimice el comportamiento de *DARAL*.

El desarrollo de la búsqueda de los parámetros óptimos se presenta a lo largo de este capítulo y se basa en un proceso iterativo en el que se partirá de la selección de parámetros fijados a mano mostrada en la Tabla 5.3, hasta alcanzar la convergencia.

5.1 CONFIGURACIÓN DE LAS CAPAS PHY Y MAC

El desarrollo del estudio que permita demostrar la optimización del tiempo de convergencia y los mensajes de control enviados en el proceso de creación de las redes requerirá el uso de diversas tecnologías de las ya comentadas en la sección 4.1.1, así como de técnicas y elementos como los comentados en los Apéndices B y C. A continuación se detallan las decisiones que se han adoptado para tomar estas medidas.

5.1.1 Comunicaciones Inalámbricas

De entre los diferentes estándares de comunicaciones inalámbricas citados en el Capítulo 3, se ha hecho uso, como se ha comentado previamente, de las capas PHY y MAC definidas en el estándar IEEE 802.15.4. A nivel de análisis se han seleccionado *AODV* y *RPL* como algoritmos de enrutamiento con los que compararse, debido a su extendido uso y a su características ampliamente estudiadas. Es también destacable que existen actualmente modelos de estos algoritmos para diferentes simuladores, lo cual es un extra a la hora de analizar su comportamiento.

5.1.2 Simulador

Como se presentó en el Capítulo 3, una WSN es una red de nodos, equipados con sensores y, por lo general, con baterías, que colaboran en una tarea común. Debido a que los nodos se suelen utilizar para obtener información de lugares remotos o de difícil acceso, son equipados generalmente con radios inalámbricas, lo que permite que las redes que se forman sean de tipo inalámbrico.

De las alternativas descritas en el Apéndice A, se hará uso de OMNeT++. La configuración básica de OMNeT++ es relativamente sencilla y se apoya en varias librerías multi-plataforma como *Java* o *TCL*, lo cual permite su ejecución en sistemas operativos *Windows*, *GNU/Linux* u otros.

Este simulador de eventos está adaptado a la simulación de modelos de comunicaciones de todo tipo. A través de un complemento llamado *INETMANET* se amplían los modelos simulables a las WSN. Dentro de estos modelos destacan tanto los derivados de IEEE 802.11 como de IEEE 802.15.4 por su contrastada eficiencia en las comunicaciones. La utilización de IEEE 802.15.4, permite llevar a cabo simulaciones de capa PHY y MAC. OMNeT++ provee de un amplio soporte para el

desarrollo de modelos de capa de red, como los considerados en esta Tesis Doctoral.

Además de las características ya mencionadas, *OMNeT++* es un simulador muy extendido entre la comunidad investigadora, por lo que su eficacia queda fuera de toda duda. Gracias al apoyo ofrecido por la comunidad alrededor de *OMNeT++*, los resultados obtenidos en esta Tesis Doctoral podrán ser contrastados dentro de un marco común con otros trabajos.

OMNeT++ es un simulador basado en diferentes módulos como se detalla en el Anexo A. La Figura 5.1 muestra la representación gráfica de un nodo *DARP* como los descritos en el Apéndice B. Como se puede apreciar, cada caja representa un módulo diferente con su función específica dentro del simulador. El módulo coloreado es el módulo perteneciente a *DARAL*, mientras que los restantes módulos conforman los diferentes niveles de un nodo WSN. Estos módulos se corresponden con la estructura de capas presentada en la Figura 3.1.

- app: Hace referencia a la *Capa de aplicación*, donde se simula una aplicación que sería la generadora del tráfico, como por ejemplo el *Generador de tráfico* o (*TraffGen*).
- net: Es el módulo que representa a la *Capa de red*. En él se simulan los algoritmos de enrutamiento como *DARAL*.
- ifq: Este módulo hace referencia a un *Sistema de colas*, el cuál gestiona los paquetes que pasan de la capa de red a la capa de acceso al medio.
- mac: Hace referencia a la *Capa MAC* y simula el acceso al medio de un nodo. En este caso simula el estándar *IEEE 802.15.4*.

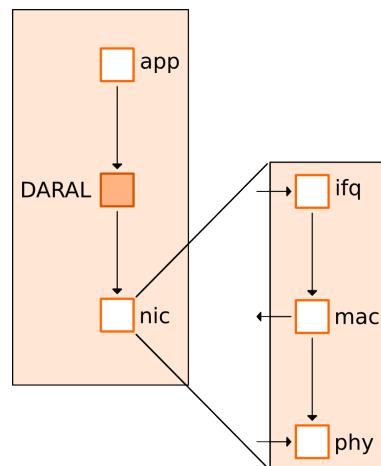


Figura 5.1
Estructura modular
de un nodo *DARP* en
OMNeT++.

- phy: Es el módulo que representa la *Capa PHY*, simulando las radios y los elementos físicos de la comunicación. Para la Figura 5.1 simula el estándar *IEEE 802.15.4*.

Estos módulos se conectan entre sí mediante puertas definidas en los archivos de configuración del simulador. Dichas puertas permiten configurar el flujo de mensajes de una capa superior a una inferior y viceversa. Para más información acerca de la configuración de un nodo en OMNeT++ se puede consultar el Apéndice B.

5.1.3 Configuración del estándar IEEE 802.15.4

En cualquier tipo de WSN, resulta necesario configurar los parámetros relativos a las capas PHY y MAC definidos en el estándar *IEEE 802.15.4*. Debido a que la presente Tesis Doctoral se ha validado en base a modelos de simulación para OMNeT++, resulta de interés presentar la configuración utilizada tanto para los parámetros que caracterizan el estándar *IEEE 802.15.4* como para los del medio inalámbrico por el que circulan los mensajes.

Las Tablas 5.1 y 5.2 presentan todos los parámetros de interés, así como los valores utilizados.

Tabla 5.1
Configuración del
medio de simulación
y de la capa PHY
del estándar *IEEE
802.15.4*.

Parámetro	Valor
FREQUENCY	2.4 GHz
CHANNEL	0
TRANSMITTER POWER	1.0 mW
SENSITIVITY	-85 dBm
RECEPTION THRESHOLD	-110 dBm
THERMAL NOISE	-110 dBm
PATH LOSS ALPHA	3
ANTENNA GAIN	0 dB
SYSTEM LOSS FACTOR	0 dB
LOG NORMAL SHADOWING SIGMA	1
SIGNAL-TO-INTERFERENCE-PLUS-NOISE RATIO	4 dB
RADIO POWER - IDLE STATE	1.38 mA
RADIO POWER - RECEPTION STATE	9.6 mA
RADIO POWER - SLEEP STATE	0.06 mA
Max. PSDU SIZE	127 Bytes

Parámetro	Valor
MAC QUEUE	100
SIMPLE INTERFRAME SPACE	0.000192 s
Max. CSMA BACKOFFS	5
MAC. FRAME RETRIES	3
ACK WAIT DURATION	0.00056 s
CCA DETECTION TIME	0.000128 s
RECEPTION SETUP TIME	0.00108 s
TURNAROUND TIME	0.000192 s
BACKOFF PERIOD PER UNIT	0.00032 s
ACK LENGTH	5 B
MAC HEADER LENGTH	14 B
CONTENTION WINDOW	2

Tabla 5.2
Configuración de la capa MAC del estándar IEEE 802.15.4.

Se simularán diferentes escenarios cada uno con sus propias características, haciendo uso de modelos aleatorios de posicionamiento los cuales se han desarrollado expresamente para la simulación de grandes escenarios con un elevado número de nodos. El área de simulación que contiene los nodos variará en función de las diferentes densidades de nodo utilizadas en cada configuración.

5.2 METODOLOGÍA DE AJUSTE DE LOS PARÁMETROS DE DARAL

El proceso iterativo se ha ido realizando a partir de la configuración inicial fijada en la Tabla 5.3, analizando cada parámetro de forma independiente. En cada iteración se ha realizado un análisis de cada parámetro como los llevados a cabo en la Sección 5.3, utilizando en cada nueva iteración los valores que resultaban más óptimos, y continuando hasta que un conjunto de valores ha dado un resultado estable.

Para el desarrollo de las simulaciones que permitirán analizar el comportamiento de los diferentes parámetros se ha utilizado un **escenario grande con 400 nodos** y una **densidad media ND = 10**, limitando las simulaciones a 3600 segundos, de forma que proporcione un resultado balanceado en las diferentes métricas analizadas.

La Tabla 5.3 presenta un resumen de los distintos valores adoptados para cada uno de los parámetros en cada una de las iteraciones. Como también se puede observar en la tabla, el proceso ha convergido tras 3 iteraciones, lo que nos ha proporcionado una configuración suficientemente buena para los grandes escenarios, objetivo de esta Tesis.

5.3 ANÁLISIS DE LA CONFIGURACIÓN ÓPTIMA PROPUESTA

Todos los parámetros que habilitan la configuración de *DARAL* son analizados en detalle en base al número de mensajes de control enviados durante una hora de simulación, el consumo energético, el número de nodos conectados a la red, el tiempo de convergencia y el número de mensajes de control enviados para la conexión. A lo largo del capítulo se estudia el comportamiento de los parámetros y se analiza una configuración óptima que proporcione buenos resultados en grandes escenarios, priorizando el *tiempo de convergencia* y el *número de mensajes de control enviados*.

El estudio de cada parámetro desarrollado a lo largo de esta sección, ha sido repetido en cada iteración hasta encontrar la configuración óptima para grandes escenarios, ya comentada. Por facilidad de comprensión, esta sección solo presenta el estudio en detalle de la última iteración, que es la que corresponde a los parámetros óptimos de configuración para grandes escenarios.

5.3.1 Umbral de mínima calidad: $TH_{baselevel}$

El umbral de mínima calidad ($TH_{baselevel}$) es utilizado por el algoritmo *DRSP*. Este parámetro permite establecer un mínimo de calidad del enlace por debajo del cual dichos enlaces potenciales no serán considerados en la selección del VC.

La medición del *LQI* es un parámetro de la capa física y se encuentra limitado a 1 Byte en el estándar *IEEE 802.15.4*. Esto se traduce en que el intervalo al que se encuentra sujeto el *LQI* va de 0 a 255. Considerando este intervalo se observa que el umbral base no puede ser excesivamente alto, o limitará en exceso el rango disponible dentro del intervalo para la selección de rol. Por este motivo se han seleccionado 4 valores para el presente estudio que permiten reducir el rango de

Tabla 5.3
Configuración original adoptada para la calibración de *DARP*.

Parámetro	Inicial	Iteración 1	Iteración 2	Iteración 3
$TH_{baselevel}$	25	5	45	45
TH_{role}	120	80	80	80
L_{nodes}	50	90	50	50
T_{link}	5.00 s	1.00 s	1.00 s	1.00 s
T_{alive}	1800.00 s	600.00 s	600.00 s	600.00 s
T_{down}	25.00 s	5.00 s	45.00 s	45.00 s
$T_{reconnect}$	6.00 s	2.00 s	2.00 s	2.00 s
T_{ack}	0.75 s	1.50 s	1.50 s	1.50 s
<i>MAX_RETRIES</i>	3	6	3	3

selección de los enlaces hasta en un 25 %. Dichos valores son 5, 25, 45 y 65.

Inicialmente se fijó el valor a mano en 25, y tras varias iteraciones su valor ha convergido a 45, manteniendo el resto de parámetros fijados a los valores óptimos mostrados en la columna *Iteración 3* de la Tabla 5.3, se ha analizado el comportamiento de $\text{TH}_{\text{baselevel}}$.

En la Figura 5.2 se puede apreciar como un valor de $\text{TH}_{\text{baselevel}} = 5$ muestra el mejor resultado medio en términos de nodos conectados, llegando a alcanzar prácticamente los 340 nodos en media, aunque $\text{TH}_{\text{baselevel}} = 65$ y $\text{TH}_{\text{baselevel}} = 45$ muestran valores cercanos también.

En la Figura 5.3 se analiza el comportamiento en términos de *tiempo de convergencia*. Como se puede observar $\text{TH}_{\text{baselevel}} = 45$ y 65 muestran un comportamiento equilibrado, obteniendo en general menores

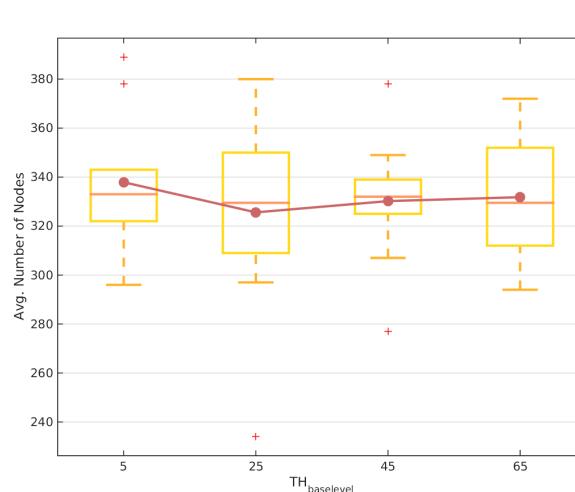


Figura 5.2
Número de nodos conectados para diferentes niveles de $\text{TH}_{\text{baselevel}}$.

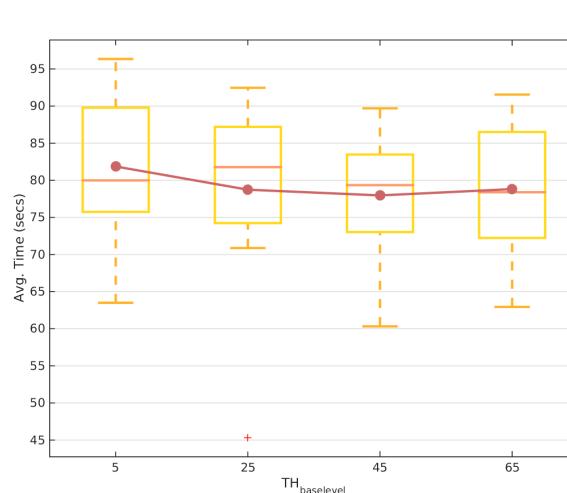


Figura 5.3
Análisis del Tiempo de Convergencia para diferentes niveles de $\text{TH}_{\text{baselevel}}$.

tiempos que el resto de configuraciones. En este análisis, el resultado de $TH_{baselevel} = 25$ presenta una menor dispersión de los valores. $TH_{baselevel} = 45$ presenta los máximos y mínimos más bajos.

La figura 5.4 muestra el *número de mensajes de control enviados durante la fase de conexión* para cada uno de los escenarios simulados.

En este análisis $TH_{baselevel} = 25$ muestra el mejor resultado en término de número de mensajes de control enviados durante la fase de conexión, obteniendo una media y una mediana inferiores al resto. A continuación $TH_{baselevel} = 65$ presenta un buen resultado, aunque $TH_{baselevel} = 45$ queda muy próximo a ambos.

La Figura 5.5 presenta el resultado de analizar todos los mensajes de control enviados durante el tiempo de simulación.

Figura 5.4
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de $TH_{baselevel}$.

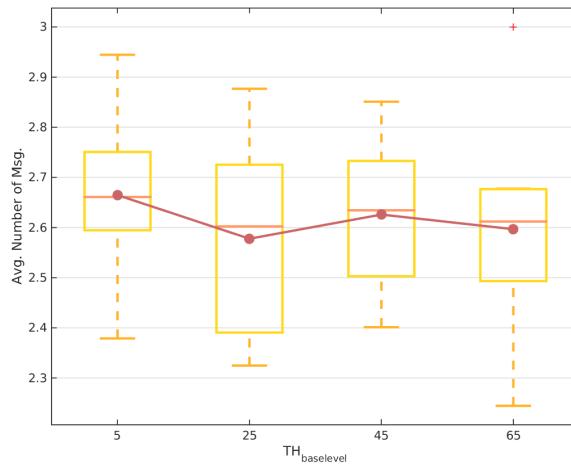
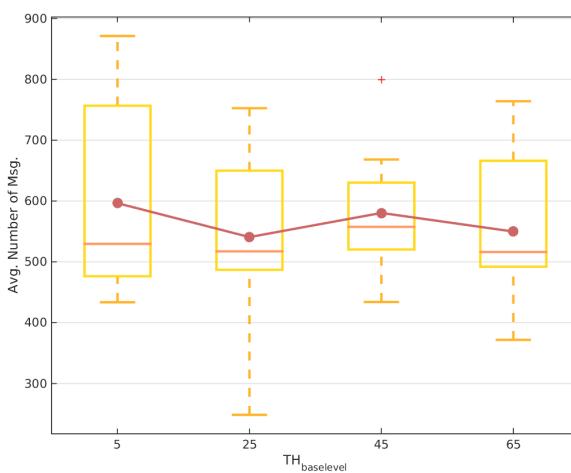


Figura 5.5
Análisis del Número de mensajes de control enviados durante todo la simulación para diferentes niveles de $TH_{baselevel}$.



En términos de número medio de mensajes de control enviados se puede observar como $TH_{baselevel} = 25$ presenta el mejor rendimiento, obteniendo una media y mediana inferior a las demás. $TH_{baselevel} = 65$ también muestra un buen desempeño.

Finalmente, la Figura 5.6 muestra la media de la evolución del consumo de energía por nodo. En este análisis todas las configuraciones presentan resultados muy similares, sin embargo $TH_{baselevel} = 65$, presenta un consumo de energía ligeramente inferior.

Como resultado se ha seleccionado $TH_{baselevel} = 45$, dado que cuenta con el menor *tiempo de convergencia* y aunque en términos absolutos de *número de mensajes de control enviados durante la fase de conexión* no es la mejor configuración, el empeoramiento es mínimo.

5.3.2 Umbral de decisión de rol: TH_{role}

Esta sección analiza el umbral de decisión TH_{role} que es utilizado por DRSP para la selección de los VCs. Al igual que $TH_{baselevel}$, este parámetro se basa en la medición del *LQI*, por lo que su rango potencial oscila entre el valor de $TH_{baselevel}$ y 255.

Para los siguientes análisis se han utilizado los valores de configuración 80, 120, 160 y 200.

La Figura 5.7 presenta el número medio de nodos conectados para cada uno de los diferentes valores de TH_{role} .

Este análisis muestra que el número medio de nodos conectados es similar para todos los valores de TH_{role} analizados, aunque $TH_{role} =$

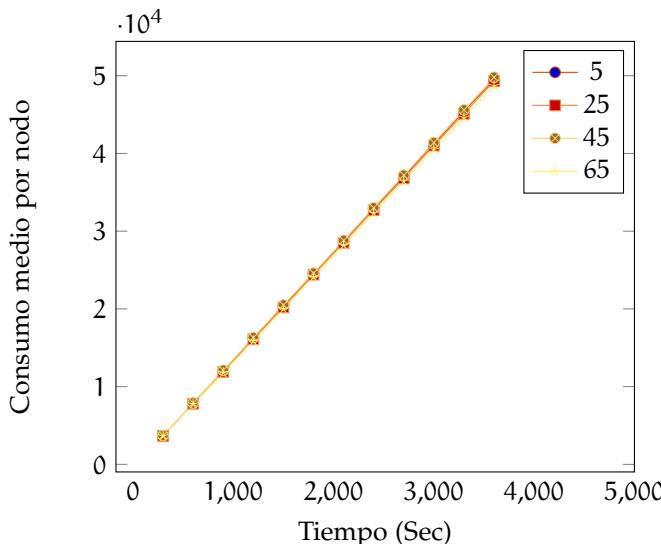


Figura 5.6
Análisis del Consumo de energía medio por nodo para diferentes niveles de $TH_{baselevel}$.

120 destaca ligeramente alcanzando hasta 330 nodos con una baja dispersión.

En la Figura 5.8 se presenta el *tiempo de convergencia* para cada uno de los diferentes escenarios analizados por separado.

A nivel de *tiempo de convergencia* se puede observar como $TH_{role} = 200$ muestra el menor tiempo de convergencia, aunque $TH_{role} = 80$ y $TH_{role} = 120$ muestran resultados cercanos presentando además una menor dispersión. $TH_{role} = 160$ presenta el peor rendimiento a nivel de media y mediana.

El siguiente análisis es el basado en el *número de mensajes de control durante la fase de conexión* y se muestra en la Figura 5.9.

En este análisis se puede observar como $TH_{role} = 200$ presenta el mejor comportamiento, ya que muestra las menores media y mediana.

Figura 5.7
Análisis del Número de nodos conectados para diferentes niveles de TH_{role} .

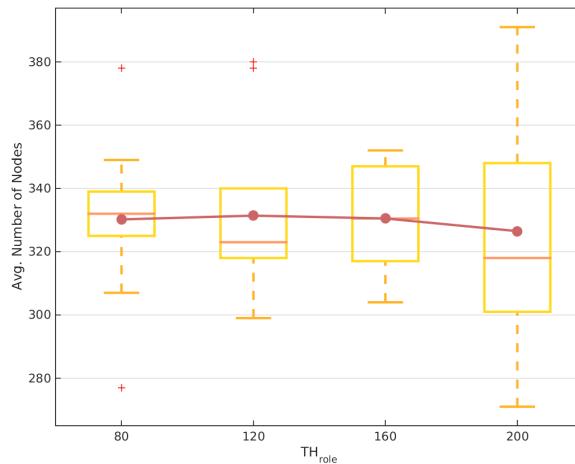
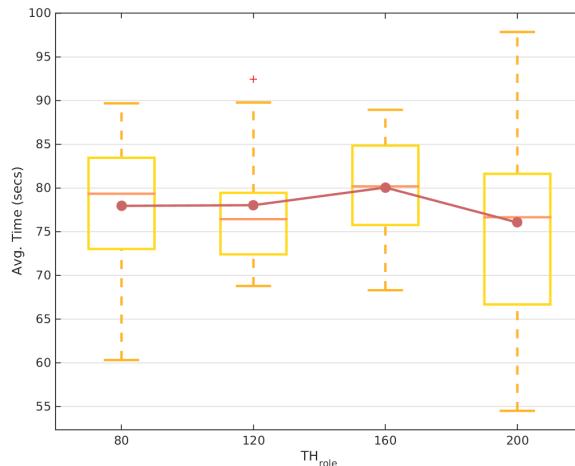


Figura 5.8
Análisis del Tiempo de Convergencia para diferentes niveles de TH_{role} .



$\text{TH}_{\text{role}} = 80$ se comporta de manera similar a $\text{TH}_{\text{role}} = 200$, aunque el máximo y mínimo son algo más elevados, también presenta una menor dispersión.

Continuando con el análisis de los mensajes de control enviados, la Figura 5.10 muestra los resultados para la medición durante toda la simulación.

En este análisis se puede observar como $\text{TH}_{\text{role}} = 120$ y $\text{TH}_{\text{role}} = 200$ presentan el mejor rendimiento, tienen los menores valores de media y mediana. $\text{TH}_{\text{role}} = 200$ cuenta con el mínimo más bajo aunque lo logra a costa de una gran dispersión de valores. Por otro lado que $\text{TH}_{\text{role}} = 80$ se presenta como una buena alternativa a $\text{TH}_{\text{role}} = 120$, dada su baja dispersión y su reducido máximo.

El análisis de consumo energético presentado en la Figura 5.11, presenta la evolución del consumo durante los 3600s de duración de las

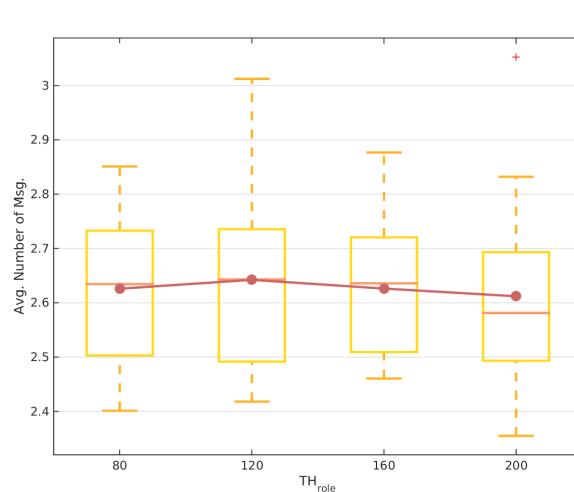


Figura 5.9
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de TH_{role} .

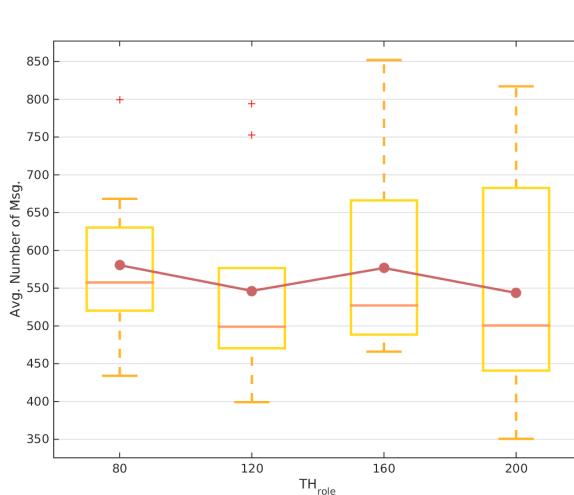


Figura 5.10
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de TH_{role} .

simulaciones. Para este análisis se utiliza la media de los diferentes escenarios simulados.

Este análisis muestra sin lugar a dudas que $\text{TH}_{\text{role}} = 200$ es el valor más eficiente en términos de consumo de energía, seguido de $\text{TH}_{\text{role}} = 120$. $\text{TH}_{\text{role}} = 160$ y $\text{TH}_{\text{role}} = 80$ presentan resultados muy similares, aunque son ligeramente peores que los cosechados por $\text{TH}_{\text{role}} = 200$.

Como resumen se puede apreciar que $\text{TH}_{\text{role}} = 80$ muestra el rendimiento más equilibrado entre *tiempo de convergencia* y *número de mensajes de control enviados durante la fase de conexión*, por ello se ha seleccionado esta configuración como la óptima y recomendable.

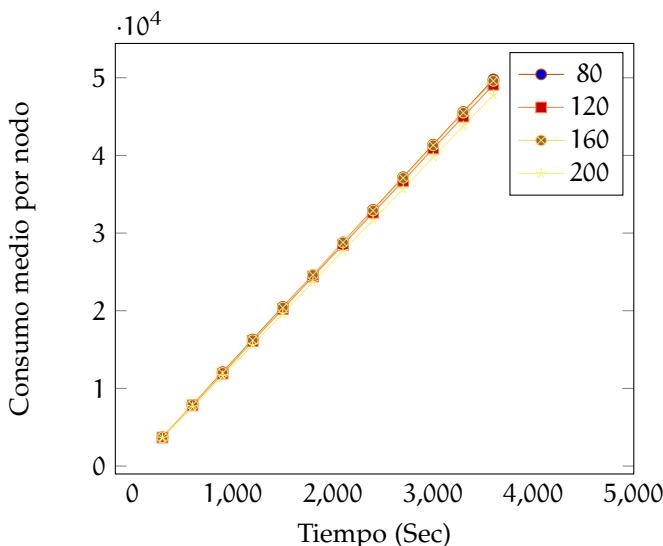
5.3.3 Límite de nodos por sub-red: L_{nodes}

La presente sección se centra en el análisis del *límite de nodos por sub-red*. Este parámetro es utilizado por el algoritmo de enrutamiento DARAL para limitar el número de nodos que puede contener un clúster. Se han realizado varias iteraciones en las que se han desarrollado los mismos análisis, como el aquí presentado.

A lo largo de la sección se muestran 5 estudios diferentes, como en las secciones previas. El estudio comienza con un análisis del número medio de nodos conectados como muestra la Figura 5.12.

Se puede observar como todas las configuraciones presentan resultados muy similares. $L_{\text{nodes}} = 10$ presenta unos resultados ligeramente superiores en términos de media, aunque a nivel de mediana $L_{\text{nodes}} = 90$ obtiene el mejor resultado. $L_{\text{nodes}} = 50$ presenta la menor dispersión y unos valores de mediana y media muy cercanos.

Figura 5.11
Análisis del Consumo de energía medio por nodo para diferentes niveles de TH_{role} .



La Figura 5.13 muestra el comportamiento de las diferentes configuraciones de L_{nodes} . $L_{nodes} = 50$ presenta el mejor rendimiento con una mediana considerablemente inferior a las configuraciones de 10 y 90. El máximo es también inferior, así como la media, presentando una dispersión inferior a $L_{nodes} = 90$.

A continuación, la Figura 5.14 muestra los resultados en término de *número de mensajes de control enviados durante la fase de formación* para L_{nodes} .

En este análisis se observa un mejor comportamiento por parte de $L_{nodes} = 90$, obteniendo una ligera mejoría a nivel de media con respecto a $L_{nodes} = 50$. Por otro lado, la dispersión de $L_{nodes} = 90$ es la menor contando además con el menor mínimo.

En el análisis de la Figura 5.15 se puede observar claramente cómo el hecho de tener los nodos más concentrados en un clúster, agiliza

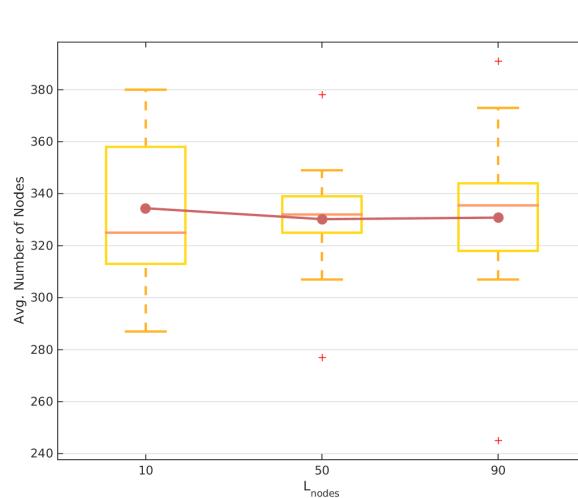


Figura 5.12
Número de nodos conectados para diferentes niveles de L_{nodes} .

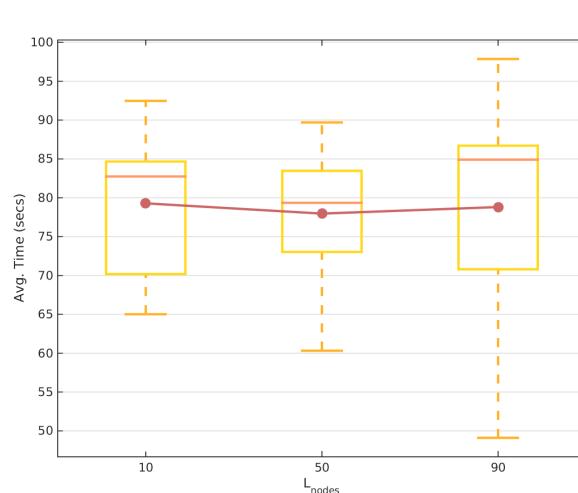


Figura 5.13
Análisis del Tiempo de Convergencia para diferentes niveles de L_{nodes} .

las comunicaciones, es por ello que $L_{nodes} = 90$ obtiene los mejores resultados, con una amplia diferencia.

Por último y para concluir con el análisis de comportamiento de L_{nodes} , se presenta en la Figura 5.16 el estudio de consumo energético medio por nodo.

En este estudio se observa el mejor rendimiento en términos de consumo por parte de $L_{nodes} = 10$, seguido muy de cerca de $L_{nodes} = 90$. Aunque en las diferentes configuraciones muestran un consumo parejo en las primeras etapas de la red.

A nivel global, se observa que $L_{nodes} = 50$ presenta una ligera mejoría en el *tiempo de convergencia* con respecto a las restantes configuraciones, mientras que en *número de mensajes de control enviados en la fase de formación* es ligeramente mayor que para $L_{nodes} = 90$, aunque a nivel de

Figura 5.14
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de L_{nodes} .

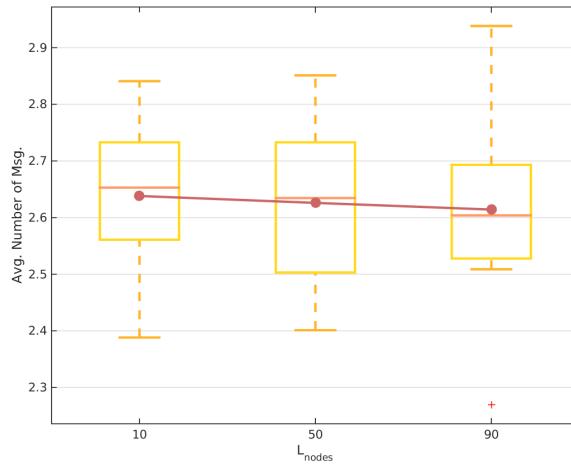
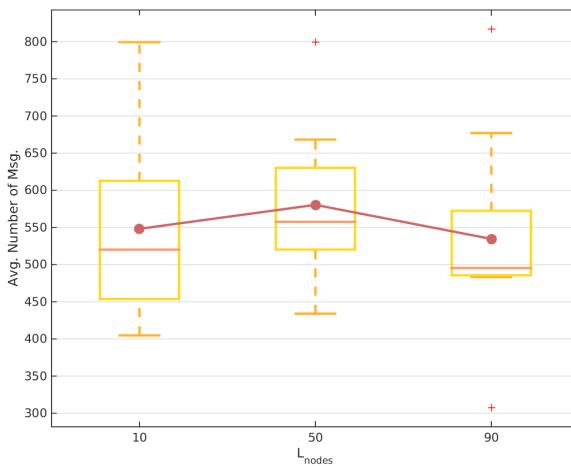


Figura 5.15
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de L_{nodes} .



número de nodos $L_{nodes} = 50$, también resulta mejor, por ellos se ha seleccionado la configuración de 50 es la seleccionada.

5.3.4 Límite de reintentos: DARAL_MAX_RETRIES

A lo largo de esta sección se presenta un estudio del comportamiento de *DARAL_MAX_RETRIES*, como el que se ha realizado para todas las iteraciones. Se han analizado como en las secciones anteriores 5 aspectos clave como el número de nodos conectados, el tiempo de convergencia, el número de mensajes de control enviados durante la fase de conexión y de forma global a la simulación, así como el consumo energético.

La Figura 5.17 presenta los resultados del análisis acerca del número de nodos conectados.

DARAL_MAX_RETRIES = 1 presenta el mejor rendimiento en general con una media superior a la de *DARAL_MAX_RETRIES* = 6 y *DARAL_MAX_RETRIES* = 3. La menor dispersión es mostrada por *DARAL_MAX_RETRIES* = 3.

En esta ocasión y como muestra la Figura 5.18, una política de reenvíos más permisiva parece tener un mejor resultado en términos de *tiempo de convergencia*, dado que *DARAL_MAX_RETRIES* = 3 muestra el mejor resultado, aunque también presenta una gran dispersión. *DARAL_MAX_RETRIES* = 1 presenta la menor dispersión.

A nivel de *número de mensajes de control enviados durante la fase de conexión*, la Figura 5.19 muestra los resultados medios.

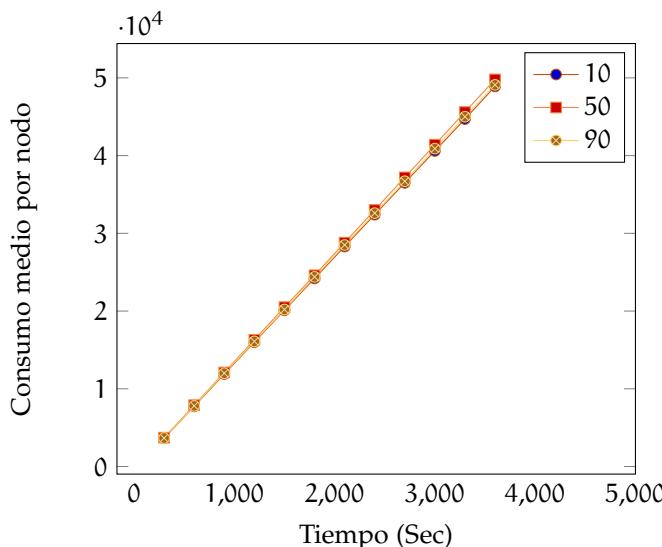
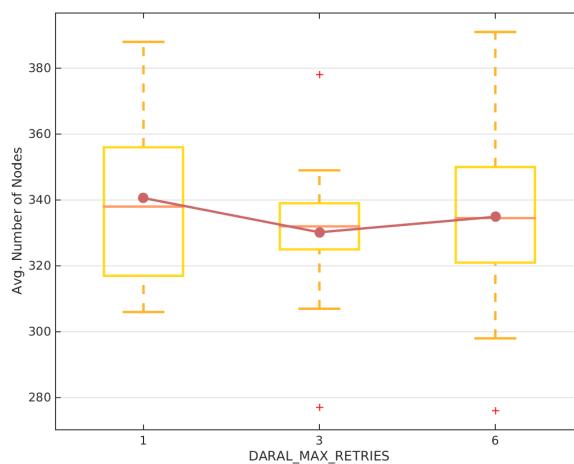


Figura 5.16
Análisis del Consumo de energía medio por nodo para diferentes niveles de L_{nodes} .

Figura 5.17
Número de nodos conectados para diferentes niveles de DARAL_MAX_RETRIES.



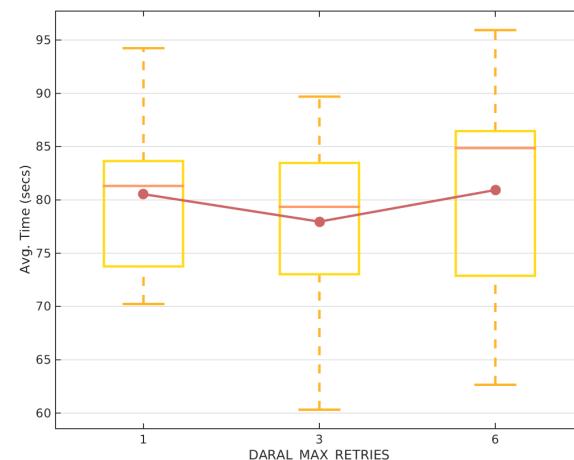
En este análisis se puede observar como en el caso anterior que una política de reenvíos equilibrada mejora el *número de mensajes de control enviados durante la fase de conexión*, ya que $\text{DARAL_MAX_RETRIES} = 3$ presenta los mejores resultados.

La Figura 5.20 presenta los valores resultantes de extender el análisis de los mensajes de control enviados a toda la simulación.

En esta ocasión, una política de reenvíos más agresiva resulta en una mejor tasa de mensajes de control enviados. $\text{DARAL_MAX_RETRIES} = 1$ ofrece el mejor rendimiento. $\text{DARAL_MAX_RETRIES} = 3$ es la peor configuración a utilizar, aunque el resultado es muy similar al de 6.

Para finalizar con el análisis de DARAL_MAX_RETRIES, se presenta en la Figura 5.21 los resultados referentes al consumo energético.

Figura 5.18
Análisis del Tiempo de Convergencia para diferentes niveles de DARAL_MAX_RETRIES.



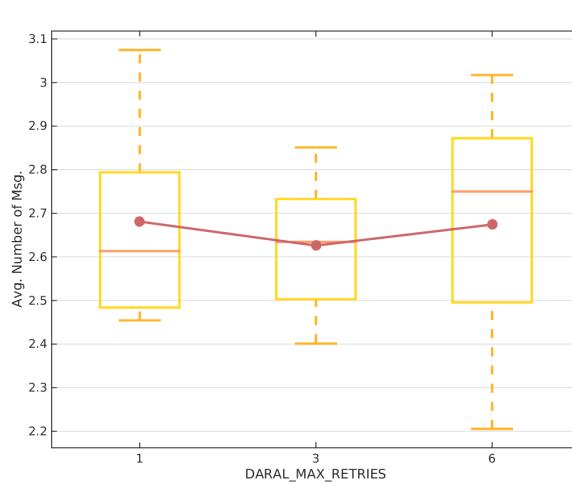


Figura 5.19
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de DARAL_MAX_RETRIES.

Este análisis de consumo resulta muy interesante ya que se puede observar a lo largo de la evolución de la simulación, como ha ido cambiando el consumo con respecto a cada configuración. Cabe recordar que la Figura 5.21 presenta los valores medios de consumo para todos los escenarios simulados. Como se puede apreciar $\text{DARAL_MAX_RETRIES} = 1$ presenta el menor consumo. Tanto $\text{DARAL_MAX_RETRIES} = 6$ como $\text{DARAL_MAX_RETRIES} = 3$ obtienen resultados muy similares.

Finalmente, se ha seleccionado $\text{DARAL_MAX_RETRIES} = 3$ como parámetro óptimo debido a que presenta tanto el *tiempo de convergencia* como el *número de mensajes de control durante la fase de conexión* más bajos, siendo estos los principales parámetros a optimizar.

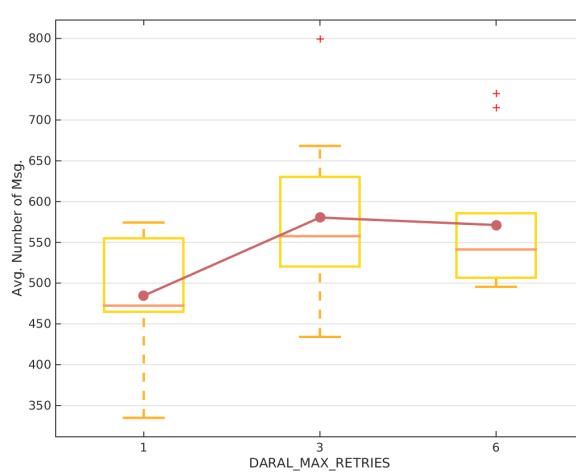


Figura 5.20
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de DARAL_MAX_RETRIES.

Figura 5.21
Análisis del Consumo de energía medio por nodo para diferentes niveles de DARP_MAX_RETRIES.

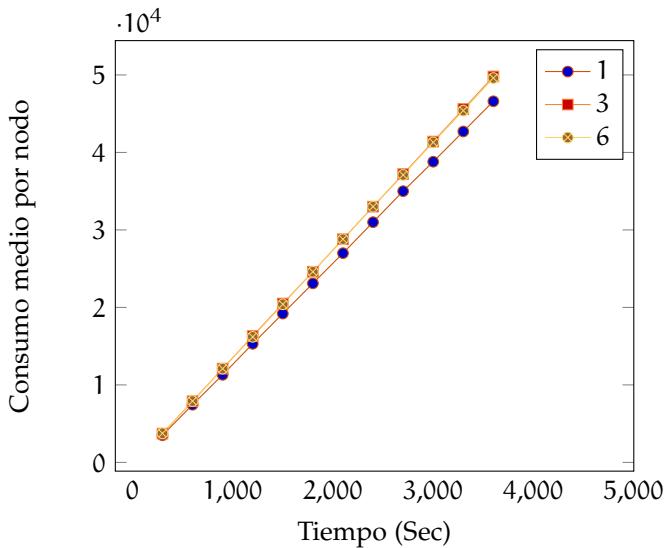
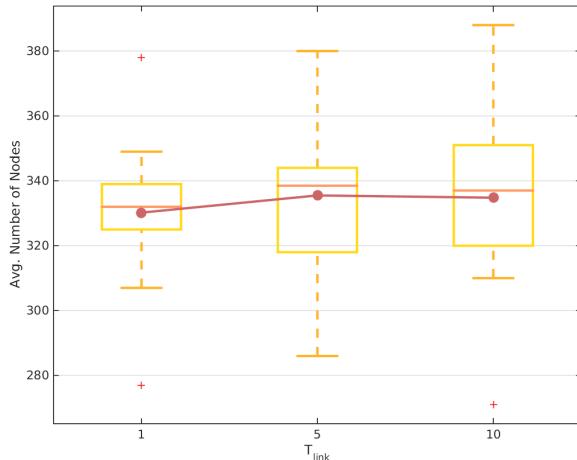


Figura 5.22
Número de nodos conectados para diferentes niveles de T_{link}.



5.3.5 Temporizador de conexión T_{link}

Este temporizador es utilizado para fijar el tiempo durante el que un nodo determinado recogerá las diferentes respuestas a su petición de conexión. La configuración aplicada en los diferentes estudios tanto para las diferentes iteraciones como para los siguientes experimentos, se encuentra recogida en la Tabla 5.3.

En la Figura 5.22 se han presentado los resultados referentes al número medio de nodos conectados. En esta gráfica se puede observar que 5s presenta la mejor tasa de conexión, alcanzando hasta 340 nodos, aunque las otras configuraciones presentan solo una tasa de conexión un 2% inferior.

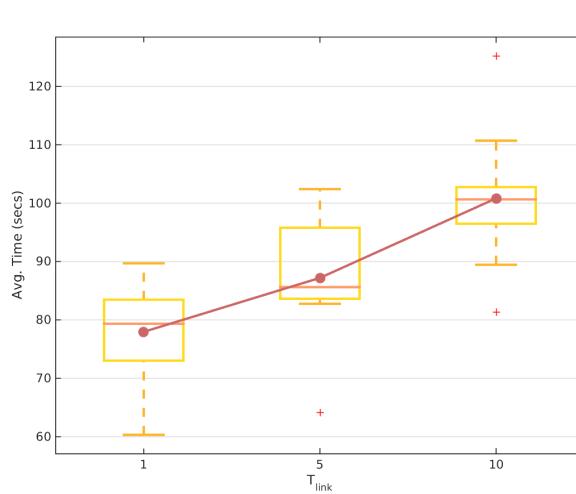


Figura 5.23
Análisis del Tiempo de Convergencia para diferentes niveles de T_{link}.

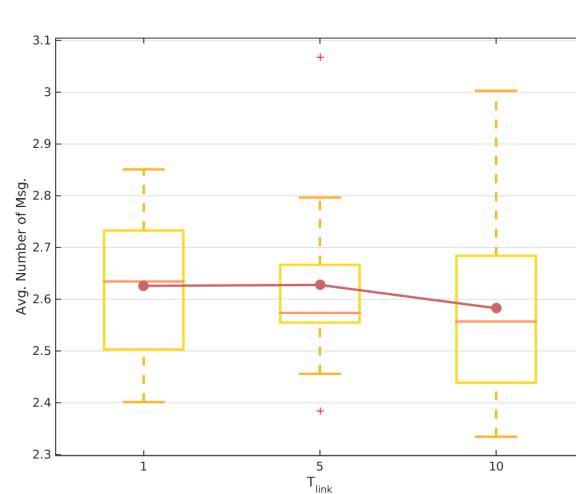


Figura 5.24
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{link}.

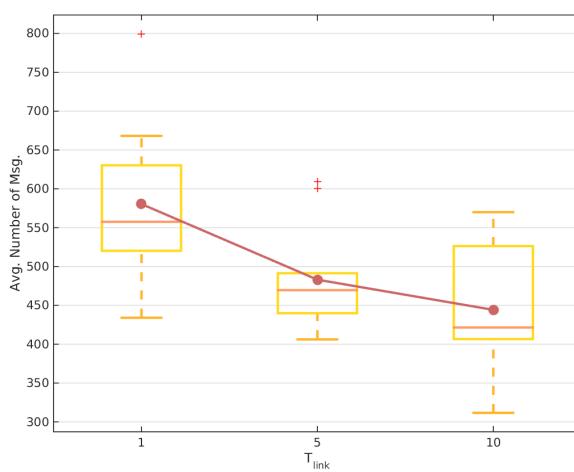
La Figura 5.23 presenta el *tiempo de convergencia* asociado a cada escenario de forma independiente. En esta figura se puede observar, como era de esperar, que $T_{\text{link}} = 1$ obtiene los mejores resultados, con una baja dispersión y una media y medianas inferiores al resto.

A nivel de *número de mensajes de control enviados durante la fase de conexión*, la Figura 5.24 muestra la configuración de 10s como la más constante y la que alcanza un mejor compromiso a lo largo de los diferentes escenarios. $T_{\text{link}} = 1$ presenta en esta ocasión el peor rendimiento, a pesar de tener una media similar a $T_{\text{link}} = 5$.

La Figura 5.25 muestra el resultado del análisis del envío de mensajes de control a lo largo de toda la simulación.

El análisis del *número de mensajes de control enviados durante toda la simulación* muestra que $T_{\text{link}} = 10$ tiene el rendimiento más equilibrado en términos de menor número de mensajes de control enviados. La

Figura 5.25
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{link} .

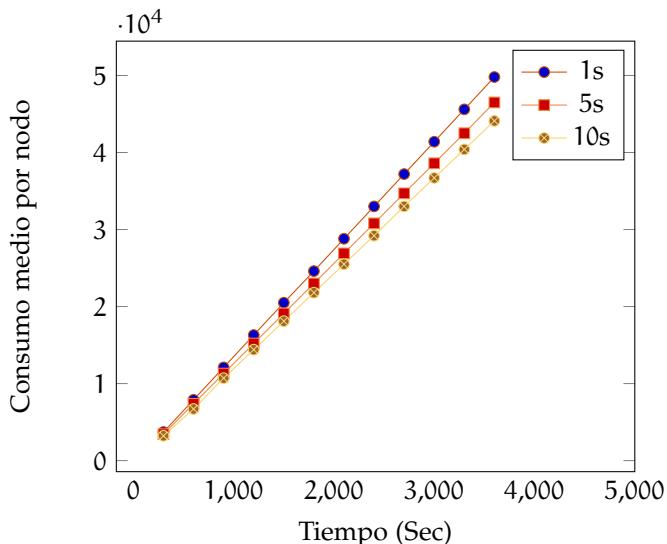


menor dispersión a lo largo de las diferentes simulaciones la presenta en esta ocasión la configuración de 5s.

El último análisis se resume en la Figura 5.26 presenta el consumo energético medio asociado a cada nodo para los diferentes valores de T_{link} . En esta figura se puede apreciar como $T_{link} = 10$ presenta el mejor consumo, seguido de $T_{link} = 5$, obteniendo $T_{link} = 1$ el peor dato de consumo.

Como resumen de este estudio se puede observar que $T_{link} = 1$ obtiene ligeras desventajas en la mayoría de los análisis, aunque obtiene una enorme mejora en el análisis del *tiempo de convergencia*, lo que resulta en su selección final.

Figura 5.26
Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{link} .



5.3.6 Tiempo de vida de los enlaces T_{alive}

El temporizador T_{alive} fija el tiempo necesario para realizar una revisión del estado de la red. Es utilizado por los VCs para controlar el estado de los ENs asociados a su sub-red. El presente estudio de rendimiento se ha realizado para cada una de las iteraciones, seleccionando en cada una el mejor resultado. El estudio aquí presentado se basa en los mismos análisis que se han ido realizando a lo largo de este Capítulo. Los valores utilizados son los presentados en la Tabla 5.3.

En la Figura 5.27 se observa el resultado medio obtenido para las diferentes configuraciones de T_{alive} estudiadas. Todas las configuraciones obtiene resultados similares, aunque $T_{alive} = 1200$ obtiene ligeramente un mejor rendimiento. Por otro lado, $T_{alive} = 600$ presenta la menor dispersión.

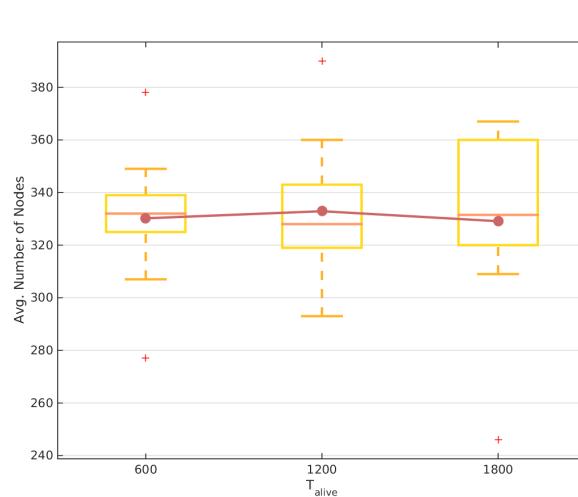


Figura 5.27
Número de nodos conectados para diferentes niveles de T_{alive} .

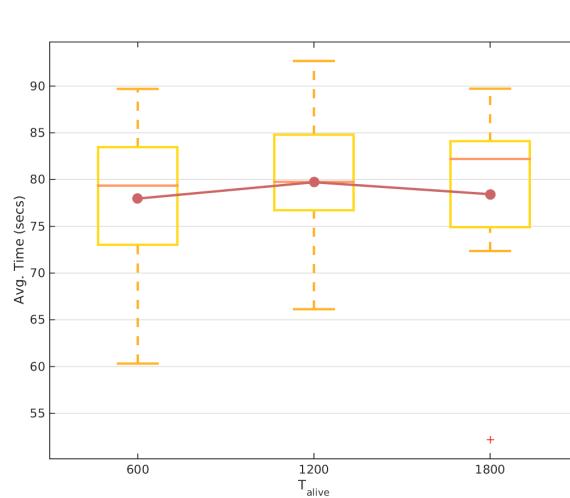


Figura 5.28
Análisis del Tiempo de Convergencia para diferentes niveles de T_{alive} .

La Figura 5.28 muestra los resultados en términos de *tiempo de convergencia*. Como se puede apreciar $T_{alive} = 600$ muestra los valores medios más bajos, seguido de cerca por $T_{alive} = 1800$, que además cuenta con la menor dispersión. La configuración de 1200s presenta los peores resultados en términos generales.

La Figura 5.29 presenta los resultados del análisis del *número de mensajes de control enviados durante la fase de conexión*. Estos resultados muestran como $T_{alive} = 600$ presenta un mejor rendimiento. En este análisis $T_{alive} = 1800$ mejora a $T_{alive} = 1200$, presentando un máximo y mínimo inferiores, así como una menor media.

Extendiendo el análisis del número de mensajes de control enviados a toda la duración de la simulación, la Figura 5.30 presenta los resultados detallados. En esta ocasión la configuración de 1200s presenta el mejor rendimiento en todas las facetas. Presentando además la media, la mediana y el mínimo más bajos, manteniéndose por debajo de los

Figura 5.29
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{alive} .

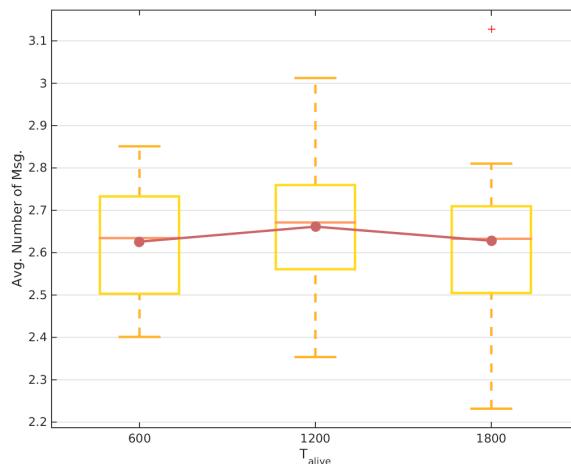
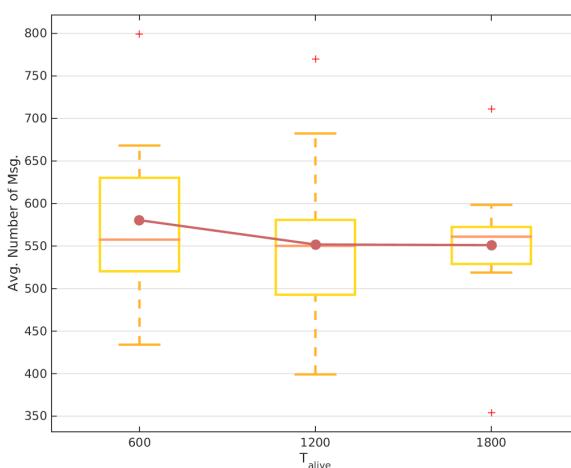


Figura 5.30
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{alive} .



700 mensajes en todos los casos. Las otras configuraciones presentan resultados dispares, con mayores medias y medianas.

Finalmente la Figura 5.31 muestra los resultados de la evolución del consumo energético medio durante la simulación. Todas las configuraciones obtienen unos resultados muy similares, aunque $T_{alive} = 1200$ presenta unos resultados ligeramente más eficientes a lo largo de toda la simulación.

Como resumen se observa que $T_{alive} = 600$ presenta unos resultados ligeramente mejores tanto a nivel de número de nodos conectados, de *tiempo de convergencia* y de *número de mensajes de control enviados durante la fase de conexión*. En los restantes análisis, el rendimiento es ligeramente inferior, pero dado el objetivo de DARAL, se selecciona este parámetro como óptimo.

5.3.7 Temporizador de desconexión T_{down}

T_{down} es un parámetro utilizado tanto por los VCs como por los ENs para controlar el tiempo que un nodo permanece desconectado, una vez detectado un problema con la comunicación. Como en los análisis previos, se han desarrollado estos estudios para cada una de las iteraciones y a continuación se presenta el estudio de la última iteración, donde se hace uso de los valores de configuración recogidos en la Tabla 5.3.

La Figura 5.32 presenta los resultados medios en función del número de nodos conectados. Como se puede apreciar la configuración de 25s presenta los mejores resultados, superando los 340 nodos conectados. La media de $T_{down} = 5$ alcanza los 340 nodos, mientras que la de

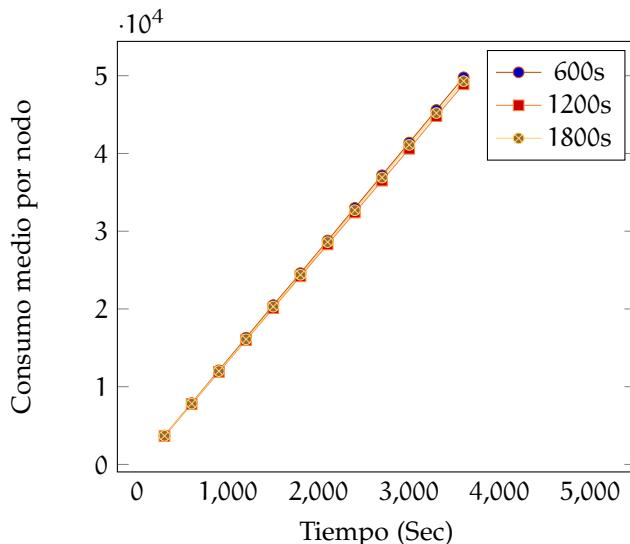
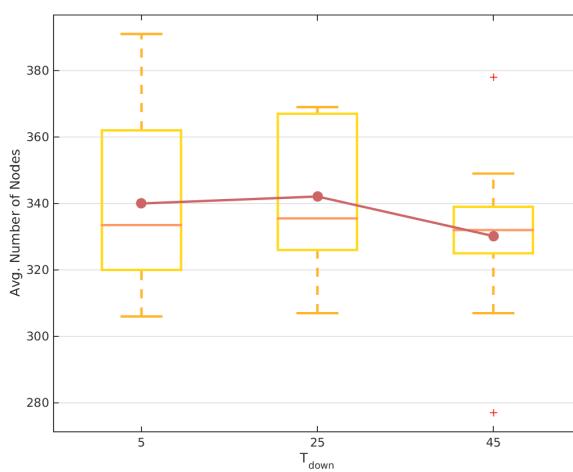


Figura 5.31
Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{alive} .

Figura 5.32
Número de nodos conectados para diferentes niveles de T_{down} .

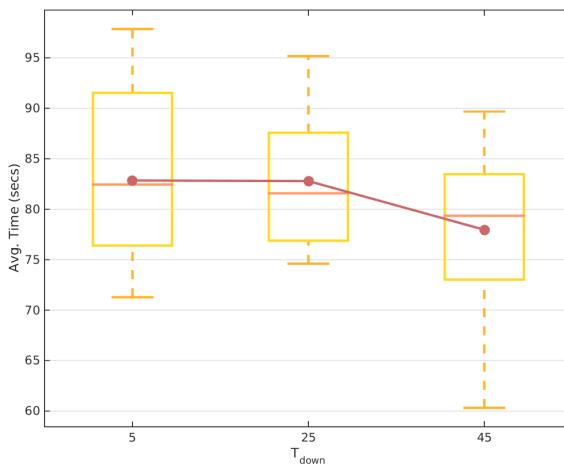


$T_{down} = 45$ es un 3 % más baja. La dispersión mostrada por $T_{down} = 45$ es la menor de todas.

La Figura 5.33 muestra los resultados del análisis del *tiempo de convergencia* y como se puede observar $T_{down} = 45$ muestra los mejores resultados aún a pesar de mostrar una amplia dispersión. Las configuración de 25s también presenta un muy buen resultado, con una media similar a la de $T_{down} = 5$, pero con una mediana y una dispersión menores. $T_{down} = 25$ a pesar de mostrar la menor dispersión de entre todos los resultados, presenta unos valores de media y mediana más elevados que los de $T_{down} = 45$.

En la Figura 5.34 y a nivel de *número de mensajes de control enviados durante la fase de conexión*, $T_{down} = 45$ presenta una media ligeramente superior a 2.6 mensajes por nodo, aunque en este análisis $T_{down} = 5$ presenta también un buen desempeño. Las medianas en ambas confi-

Figura 5.33
Análisis del Tiempo de Convergencia para diferentes niveles de T_{down} .



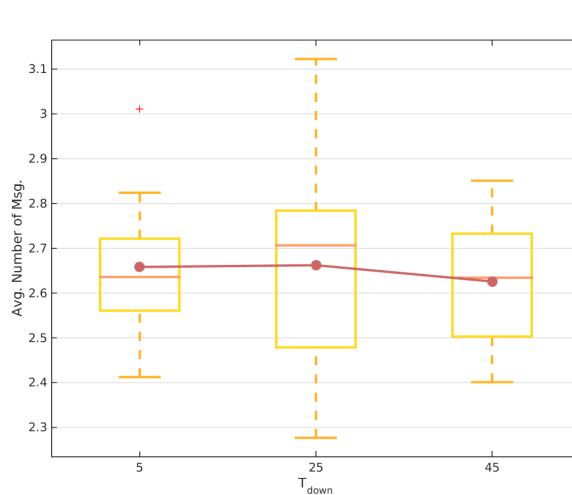


Figura 5.34
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{down} .

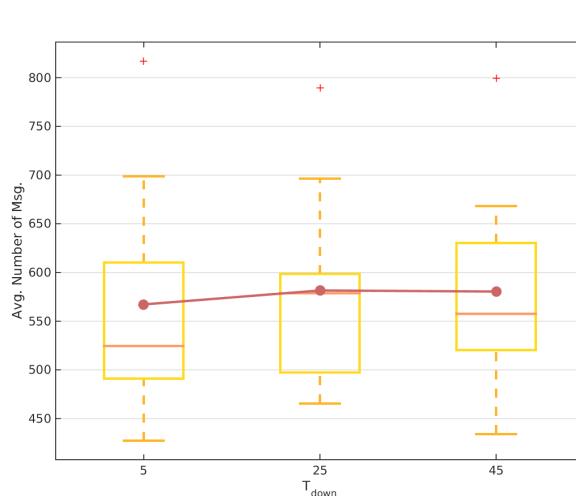


Figura 5.35
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{down} .

guraciones presentan similares resultados, aunque $T_{down} = 5$ obtiene los mejores resultados a nivel de máximo y en términos de dispersión. $T_{down} = 45$, por su parte, presenta el mejor máximo y una dispersión muy similar a la de $T_{down} = 5$.

El análisis del número de mensajes de control a nivel global mostrado en la Figura 5.35 presentan a $T_{down} = 5$ como la mejor opción. Esta configuración obtiene los mejores resultados a nivel de media, mediana y mínimo, aunque resulta en una mayor dispersión que las demás configuraciones.

A nivel de consumo energético, la Figura 5.36 presenta los resultados en función del tiempo de simulación, y presentan la evolución del consumo para cada una de las configuraciones. Se puede observar como $T_{down} = 5$ presenta el menor consumo desde el inicio, mientras que $T_{down} = 25$ y $T_{down} = 45$ muestran prácticamente el mismo consumo, mayor en ambos casos que el de $T_{down} = 5$.

En este estudio se observa como $T_{down} = 45$ obtiene el mejor rendimiento tanto en términos de *tiempo de convergencia* como de *número de mensajes de control enviados durante la fase de conexión*. Aún cuando el rendimiento decae ligeramente en el resto de estudios, su mejoría en los principales parámetros a considerar por DARAL, hacen que 45 sea el valor seleccionado finalmente.

5.3.8 Temporizador de reconexión $T_{reconnect}$

El temporizador de reconexión o $T_{reconnect}$, es utilizado de forma interna por un nodo cuando pierde la conectividad con el VC. Una vez este temporizador expira, el nodo reinicia la pila y comienza de nuevo el proceso de conexión a la red, buscando un nuevo VC válido.

Figura 5.36
Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{down} .

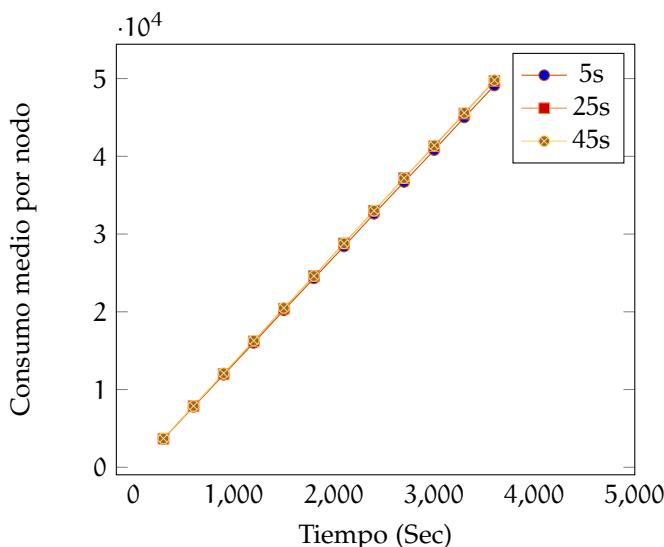
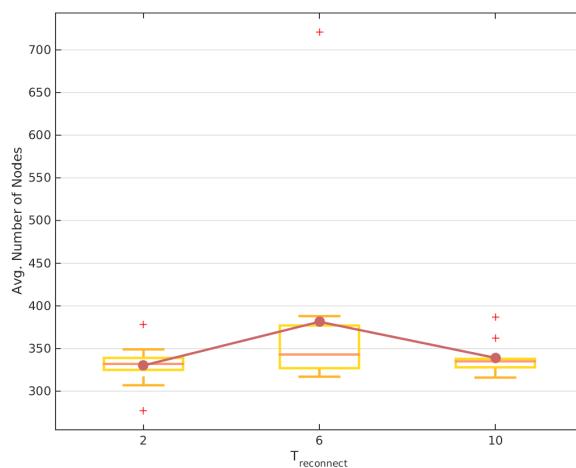


Figura 5.37
Número de nodos conectados para diferentes niveles de $T_{reconnect}$.



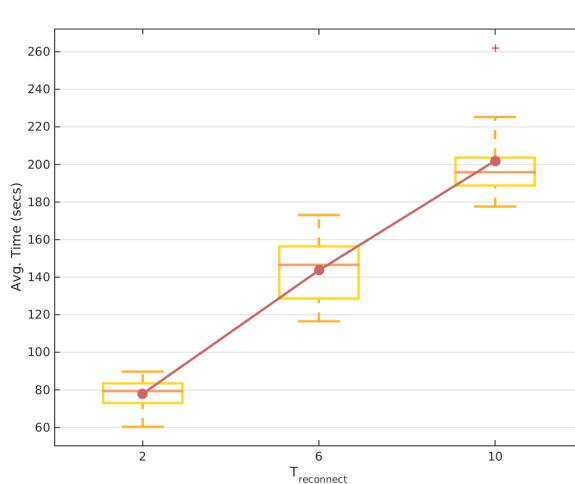


Figura 5.38
Análisis del Tiempo de Convergencia para diferentes niveles de $T_{reconnect}$.

Como en las iteraciones previas, en estos estudios se busca el parámetro óptimo a utilizar. A lo largo de esta sub-sección se analizan las 5 características previamente analizadas en las que se presentan detalladamente los resultados del estudio de la última iteración y cuya configuración se encuentra descrita en la Tabla 5.3. Para comenzar, la Figura 5.37 se observa el análisis del número medio de nodos conectados.

$T_{reconnect} = 6$ presenta las mejores media y mediana, alcanzando la media hasta aproximadamente, los 380 nodos conectados, aunque cuenta con la mayor dispersión. $T_{reconnect} = 2$ presenta la peor media, alcanzando solo 340 nodos, a pesar de mostrar una baja dispersión de los resultados.

La Figura 5.38 presenta los resultados del análisis del *tiempo de convergencia*, en el que destaca con una amplia diferencia $T_{reconnect} = 2$. Esta configuración obtiene unas media y mediana cercanas a los 80s con una baja dispersión. El siguiente resultado es $T_{reconnect} = 6$ que alcanza en media alrededor de los 140s. $T_{reconnect} = 10$ se va hasta los 200s aproximadamente. En general, la dispersión mostrada por todas las configuraciones es baja.

El resultado del análisis del *número de mensajes de control enviados durante la fase de conexión* se muestra en la Figura 5.39. Nuevamente, $T_{reconnect} = 2$ presenta la menor media y una baja mediana, con una dispersión contenida y los menores máximos y mínimos. $T_{reconnect} = 6$ y $T_{reconnect} = 10$ presentan unos valores de media y mediana similares.

A nivel de número de mensajes de control enviados de forma global a toda la duración de la simulación, los resultados mostrados en la Figura 5.40 muestran un importante cambio, donde $T_{reconnect} = 10$ muestra los mejores resultados con una amplia diferencia. $T_{reconnect} = 10$ alcanza los 380 mensajes por nodo en media, obteniendo los mejores máximo y mínimo y una dispersión ligeramente superior a la de

$T_{reconnect} = 6$. $T_{reconnect} = 2$ obtiene los peores resultados alcanzando la media los 580 mensajes por nodo. El máximo y mínimo también son los más elevados, así como la dispersión que resulta en la mayor de todas las analizadas.

El consumo de energía en base a las diferentes configuraciones de $T_{reconnect}$ mostrado en la Figura 5.41 resulta en un menor consumo de $T_{reconnect} = 10$, seguido de $T_{reconnect} = 6$ y por último, como peor resultado $T_{reconnect} = 2$.

A pesar del mayor consumo de energía, $T_{reconnect} = 2$ muestra el mejor rendimiento en términos de *tiempo de convergencia* y *número de mensajes de control durante la fase de conexión*, por lo que se ha seleccionado como parámetro óptimo para la simulación de grandes escenarios con media o baja densidad.

Figura 5.39
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de $T_{reconnect}$.

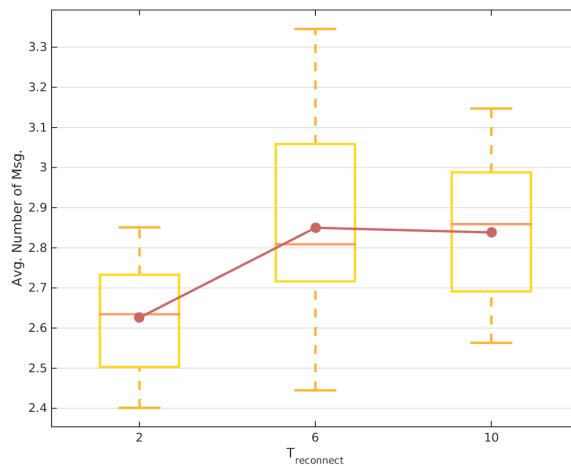
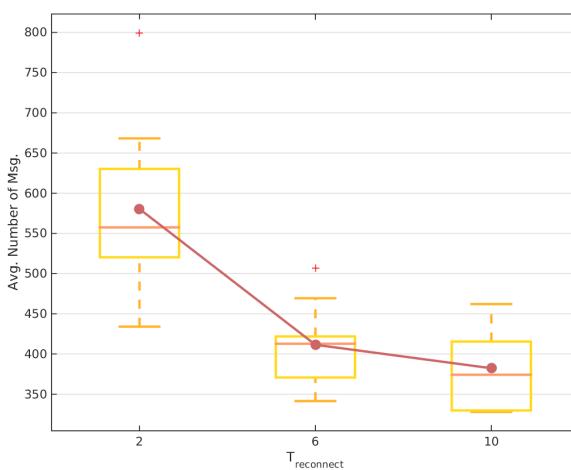


Figura 5.40
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de $T_{reconnect}$.



5.3.9 Tiempo de espera para respuestas T_{ack}

El tiempo de respuesta o T_{ack} es un temporizador ampliamente utilizado en todos los protocolos de comunicaciones existentes. Este temporizador marca el tiempo necesario que un nodo debe esperar para recibir una respuesta al mensaje enviado. Una vez el temporizador expira, se reenvía el último mensaje enviado.

El estudio de este parámetro a través de los 5 análisis que se han desarrollado permite calibrar y afinar este parámetro, al igual que se ha hecho en las iteraciones anteriores. Cabe recordar que la configuración usada es la descrita en la Tabla 5.3 y que pertenece a la última iteración realizada. En la Figura 5.42 se muestran los resultados del primer análisis.

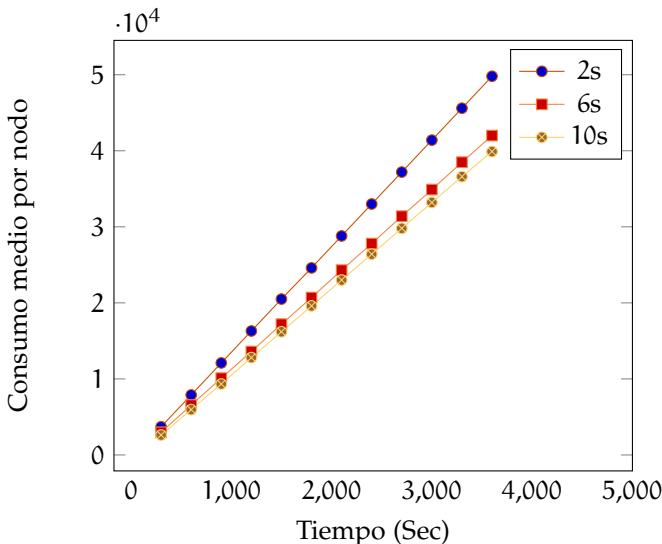


Figura 5.41
Análisis del Consumo de energía medio por nodo para diferentes niveles de $T_{reconnect}$.

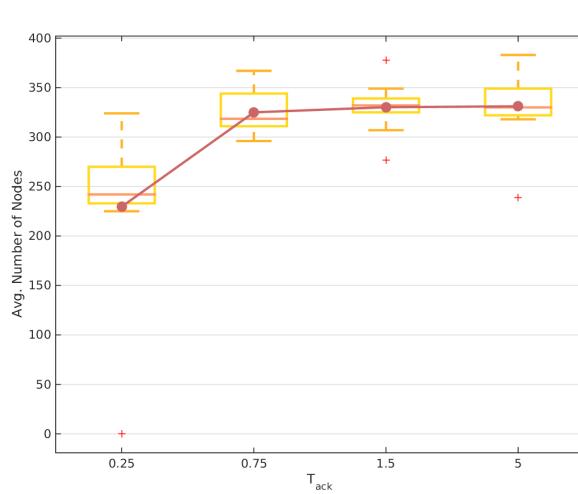
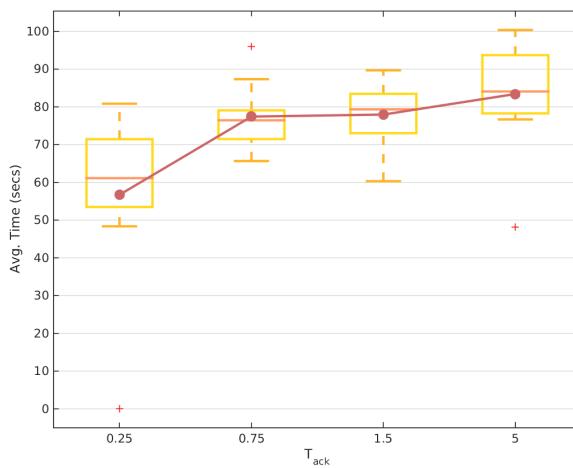


Figura 5.42
Número de nodos conectados para diferentes niveles de T_{ack} .

Figura 5.43
Análisis del Tiempo de Convergencia para diferentes niveles de T_{ack} .

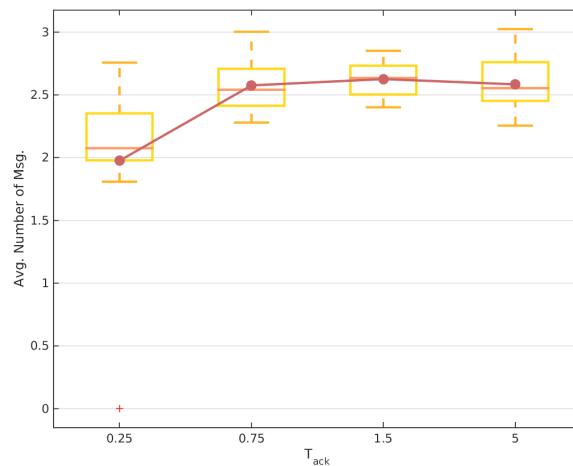


Este análisis revela que todas las configuraciones salvo $T_{ack} = 0.25$ presentan un comportamiento muy similar. $T_{ack} = 0.25$ muestra unos resultados realmente bajos, alcanzando su media y mediana valores alrededor de los 230 nodos. Todas las demás configuraciones se mueven en valores medios en torno a los 330 nodos conectados. $T_{ack} = 0.75$ presenta la menor dispersión, seguido de $T_{ack} = 1.5$.

La Figura 5.43 muestra los resultados del análisis del *tiempo de convergencia* para T_{ack} , donde se puede observar como $T_{ack} = 0.25$ presenta la mejor media y mediana, aunque también presenta la mayor dispersión. $T_{ack} = 5$ obtiene en este análisis los peores resultados.

En el análisis del *número de mensajes de control enviados durante la fase de formación* mostrado en la Figura 5.44, se observa nuevamente como $T_{ack} = 0.25$ obtiene los mejores resultados tanto a nivel de media, mediana máxima y mínima, aunque nuevamente presenta la mayor

Figura 5.44
Análisis del Número de mensajes de control enviados durante la fase de conexión para diferentes niveles de T_{ack} .



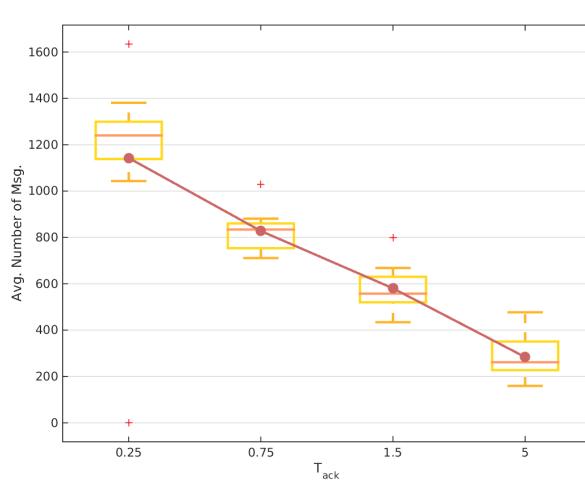


Figura 5.45
Análisis del Número de mensajes de control enviados durante toda la simulación para diferentes niveles de T_{ack} .

dispersión. Todas las demás configuraciones presentan resultados similares, aunque $T_{ack} = 1.5$ muestra la menor dispersión.

Al extender el análisis del número de mensajes de control enviados a todo el tiempo de simulación y como se puede apreciar en la Figura 5.45, los resultados se invierten, presentando $T_{ack} = 5$ el mejor resultado con una amplia diferencia. $T_{ack} = 5$ obtiene una reducida dispersión, quedando la media y la mediana en valores alrededor de los 300 mensajes por nodo. A continuación los mejores resultados son mostrados por $T_{ack} = 1.5$, que alcanza los 580 mensajes por nodo con una menor dispersión que $T_{ack} = 5$. Estos resultados muestran también como $T_{ack} = 0.25$ supera en media los 1100 mensajes por nodo, presentando además la mayor dispersión.

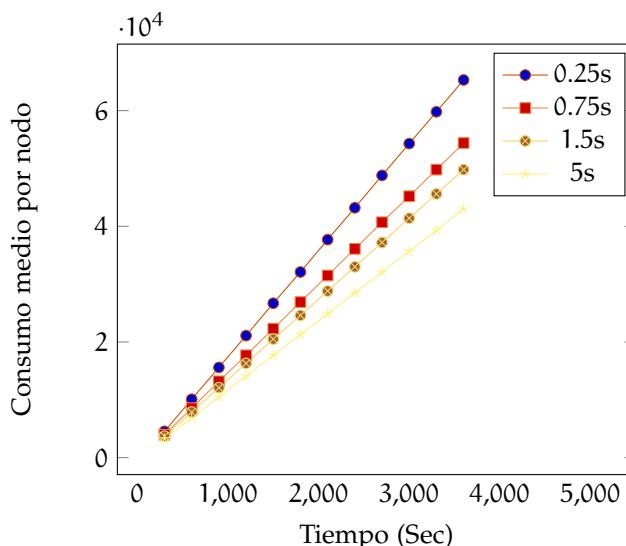


Figura 5.46
Análisis del Consumo de energía medio por nodo para diferentes niveles de T_{ack} .

Tabla 5.4
Configuración óptima
de DARP.

Parámetro	Valor
$TH_{baselevel}$	45
TH_{role}	80
L_{nodes}	50
T_{link}	1.00 s
T_{alive}	600.00 s
T_{down}	45.00 s
$T_{reconnect}$	2.00 s
T_{ack}	1.5 s
<i>DARAL_MAX_RETRIES</i>	3

En la Figura 5.46 se presentan finalmente los resultados de consumo asociados a T_{ack} , y como cabría de esperar tras el análisis del número de mensajes de control enviados a lo largo de toda la simulación, $T_{ack} = 1.5$ y $T_{ack} = 5$ obtienen los mejores resultados, superando ampliamente a $T_{ack} = 0.25$ y $T_{ack} = 0.75$.

Como resultado final se puede afirmar que tanto $T_{ack} = 0.25$ presenta los mejores resultados en términos de *tiempo de convergencia* y *número de mensajes de control enviados durante la fase de conexión* pero en esta ocasión, el número de nodos conectados presenta una importante reducción que no puede ser pasada por alto, por ello se ha seleccionado $T_{ack} = 1.5$. Esta configuración presenta un desempeño similar a $T_{ack} = 0.75$, pero mejorándolo en cuanto a número total de mensajes enviados y consumo energético, por lo que se aconseja su uso como parámetro óptimo.

5.4 CONCLUSIONES

A lo largo de este Capítulo se ha presentado un estudio de comportamiento de los diferentes parámetros de configuración que caracterizan DARAL y DARP. A través del desarrollo de un proceso de búsqueda iterativo a partir de una serie de parámetros fijados a mano, se ha convergido tras tres iteraciones en una configuración óptima para grandes escenarios con una densidad media o baja.

Modelo de simulación: Este algoritmo será validado a través del desarrollo de un modelo de simulación para OMNeT++, mediante el cual se harán diversos análisis de rendimiento del protocolo y de los diferentes algoritmos.

La configuración óptima alcanzada y que se presenta a modo de resumen en la Tabla 5.4, ha sido discutida y analizada de forma detallada, parámetro a parámetro y será la utilizada de ahora en adelante para todas las pruebas desarrolladas en esta Tesis Doctoral.

6

ESTUDIO COMPARATIVO DE DIFERENTES PROTOCOLOS PARA WSN

ÍNDICE

6.1	Configuración de los Experimentos	104
6.1.1	Protocolos de enrutamiento más utilizados	104
6.1.2	Configuración de los escenarios de simulación	104
6.1.3	Configuración de la aplicación	106
6.1.4	Obtención de resultados estadísticos	107
6.2	Resultados experimentales para el análisis del Tiempo de Convergencia	107
6.2.1	Cálculo del tiempo de convergencia	107
6.2.2	Tamaño del escenario: PEQUEÑO	108
6.2.3	Tamaño del escenario: MEDIANO	110
6.2.4	Tamaño del escenario: GRANDE	112
6.3	Resultados experimentales para el análisis del Número de Mensajes de Control Enviados	114
6.3.1	Cálculo del número de mensajes de control enviados	114
6.3.2	Tamaño del escenario: PEQUEÑO	115
6.3.3	Tamaño del escenario: MEDIANO	118
6.3.4	Tamaño del escenario: GRANDE	120
6.4	Conclusiones	124

El presente capítulo tiene como objetivo llevar a cabo el análisis de una comparativa para diversos tamaños de escenario. Así mismo, cada uno de estos casos de prueba llevará consigo sus correspondientes resultados experimentales, los cuales serán analizados en el Capítulo 8.

Este capítulo se divide en cuatro secciones principales: *Configuración de los Experimentos*, donde se detallan todas las pruebas y configuraciones llevadas a cabo, la Sección de *Resultados experimentales para el análisis del Tiempo de Convergencia*, donde se desarrolla el análisis del *Tiempo de Convergencia* en detalle. La tercera sección es la Sección de *Resultados experimentales para el análisis del Número de Mensajes de Control Enviados*, la cual refleja los resultados en términos de número de mensajes de control enviados. Finalmente, la última sección de este Capítulo, presenta unas breves conclusiones al respecto de los resultados obtenidos con los algoritmos propuestos en el Capítulo 4.

6.1 CONFIGURACIÓN DE LOS EXPERIMENTOS

Antes de comenzar a describir la configuración de los experimentos llevados a cabo en este trabajo, es necesario analizar las tecnologías que permitirán mejorar el tiempo de convergencia y mantener un bajo y constante número de mensajes de control enviados en WSNs aplicadas a grandes escenarios, tal y como se definió en la hipótesis de esta Tesis Doctoral.³ El capítulo 3, así como la información recopilada en el Apéndice A, resultan en una buena referencia para determinar las tecnologías más apropiadas para este trabajo.

6.1.1 *Protocolos de enrutamiento más utilizados*

Son muchas las opciones de comunicación inalámbrica propuestas por la comunidad científica durante los últimos años. En el Capítulo 3 se hizo una profunda revisión de todas estas opciones. De entre todas las basadas en IEEE 802.15.4 [46] destacan dos: Ad-Hoc On Demand Vector (AODV) [69] y Optimized IPv6 Routing Protocol for Low-Power and Lossy Networks (O-RPL) [87] [41].

Considerando que tanto el estándar IEEE 802.15.4 como ambos algoritmos se encuentran disponibles en OMNeT++, se han tomado sus implementaciones disponibles y se han utilizado a lo largo de los diferentes análisis y pruebas realizados en esta Tesis Doctoral.

6.1.2 *Configuración de los escenarios de simulación*

En el Capítulo 4 se detalla el por qué del desarrollo de DARAL y concretamente en la Tabla 5.4 sus principales parámetros de configuración. A continuación se expone información de configuración adicional que permite caracterizar los diferentes escenarios simulados. La red constará de los siguientes elementos:

- Un nodo *Coordinador* encargado de crear la red PAN y actuar como VC de la primera sub-red en DARAL y como nodo raíz del primer grafo en O-RPL.
- Todos los nodos participen en las simulaciones *emitirán tramas* según las diferentes temporizaciones configuradas.
- Todos los nodos que compongan las simulaciones *podrán recibir tramas*. La asignación de los orígenes y destinos de las comunicaciones se realiza de forma aleatoria.

Parámetro	Significado	Valor
C	Velocidad de la luz	299792458
F	Frecuencia de operación	2400000000
P _{Max}	Potencia máxima de emisión	0.001
λ	Coeficiente lambda	16
P _{Sat}	Umbral de atenuación de la señal	-85
α	Coeficiente de dispersión de la señal	3

Tabla 6.1
Configuración de los parámetros de la ecuación 6.2.

El objetivo de todo algoritmo de enrutamiento es el de transportar los mensajes de un nodo a otro de la red. En este sentido, diversos estudios científicos como los analizados en el Capítulo 3 han puesto de manifiesto que la densidad media de nodos es un elemento clave a la hora de evaluar el comportamiento de una red, ya que marca el potencial de conexiones e interferencias existentes.

La superficie de simulación vendrá determinada por la densidad de los nodos. Dicha densidad se puede obtener a partir de la ecuación 6.1, donde se hace uso de los parámetros definidos en los archivos de configuración de OMNeT++ y que son específicos para cada escenario a simular.

$$ND = \frac{(x - 1) \times Range^2 \times \pi}{Size_x \times Size_y} \quad (6.1)$$

La ecuación 6.1 se basa en los parámetros de configuración del escenario, donde x es el número de nodos, $Size_x$ y $Size_y$ representan el tamaño de las coordenadas X e Y del escenario y $Range$ es el alcance de la radio de un nodo y que se calcula a partir de la ecuación 6.2. La Tabla 6.1 muestra la lista de parámetros que caracterizan la ecuación 6.2 y que permiten calcular el radio de alcance de un nodo.

$$Range = \sqrt{\alpha \frac{\left(\frac{C}{F}\right)^2 \times P_{Max}}{\lambda \times \pi^2 \times 10^{-\frac{P_{Sat}}{10}}}} \quad (6.2)$$

Debido a que el escenario considerado en esta Tesis Doctoral hace referencia a las grandes infraestructuras como las *Ciudades Inteligentes* consideradas escenarios de densidad baja o media, se analizarán 9 escenarios diferentes. En estos escenarios se irán combinando tanto un número determinado de nodos, como diferentes valores para la densidad de nodos, variando $Size_x$ y $Size_y$, es decir, variando el tamaño de la superficie. La Tabla 6.2 presenta los diferentes escenarios a analizar.

Tabla 6.2
Escenarios de simulación.

Tamaño de la red	Nodos	Grado	Área
PEQUEÑO	100	5	250 × 250 m
PEQUEÑO	100	10	175 × 175 m
PEQUEÑO	100	15	145 × 145 m
MEDIANO	200	5	350 × 350 m
MEDIANO	200	10	250 × 250 m
MEDIANO	200	15	200 × 200 m
GRANDE	400	5	500 × 500 m
GRANDE	400	10	350 × 350 m
GRANDE	400	15	290 × 290 m

Tabla 6.3
Configuración de los parámetros del módulo *Traffic Generator*.

Parámetro	Significado	Valor
packetSize	Tamaño del paquete	70Bytes
interDepartureTime	Frecuencia de envío de paquetes	60s
firstPacketTime	Inicio de los envíos	60s
trajDest	Nodo Destino	host[*]

6.1.3 Configuración de la aplicación

El módulo de la aplicación se basará en *TrafficGen*, un módulo generador de tráfico existente en OMNeT++, que permite la generación de tráfico de forma controlada. Este módulo permite la configuración de la periodicidad en el envío de mensajes, del tamaño del paquete y del destino o destinos a los que se enviarán los paquetes. La configuración completa de este módulo se muestra en la Tabla 6.3.

Resulta interesante comentar que la selección del nodo destino está configurada de forma aleatoria, debido a que así se asegura que los nodos se comunicaran con cualquier punto de la red, pudiendo incluso coincidir en que un nodo sea el destino para varios emisores, de esta forma se presenta un uso más realista de la red.

Aún cuando el objetivo de desarrollo del presente algoritmo sea obtener una optimización en términos de agilidad en la formación de la red, todas las simulaciones estarán limitadas a 3600s de forma que también se puedan obtener otros parámetros de análisis que resulten de interés.

6.1.4 Obtención de resultados estadísticos

Con el fin de asegurar la veracidad de los datos recopilados así como la reproducibilidad de los mismos, se han llevado a cabo diferentes pruebas de ejecución en las que se han comparado los datos y la trazabilidad de los resultados en OMNeT++. Por otro lado, para presentar los datos más fidedignos posibles, se han considerado para cada combinación de parámetros y escenario, un mínimo de 10 ejecuciones, a partir de las cuales se ha calculado la media. De esta forma se han obtenido resultados medios, lo que nos permite obtener una buena imagen del comportamiento de los parámetros analizados.

Todos estos resultados estadísticos se han recopilado con total detalle en los siguientes apartados para todas y cada una de las pruebas realizadas.

6.2 RESULTADOS EXPERIMENTALES PARA EL ANÁLISIS DEL TIEMPO DE CONVERGENCIA

Esta sección tiene como objetivo mostrar con toda claridad de detalle los resultados estadísticos obtenidos.

Con el fin de poder obtener conclusiones sobre el comportamiento del algoritmo propuesto en el Capítulo 4, se ha realizado un estudio previo en el Capítulo 5 que permite determinar los parámetros óptimos de configuración de *DARAL* y *DARP*, como se muestra en la Tabla resumen 5.4.

Toda la información recopilada será mostrada por medio de gráficos de barras y tablas de datos a fin de poder estudiar de manera clara y concisa los resultados obtenidos. Todo ello permitirá simplificar la correcta comprensión de los resultados obtenidos.

En esta primera fase de los resultados experimentales, se han llevado a cabo una serie de pruebas basadas en los escenarios propuestos en la Tabla 6.2. A continuación se presentan los resultados organizados en base al tamaño del escenario.

6.2.1 Cálculo del tiempo de convergencia

El *tiempo de convergencia* es uno de los parámetros claves en el presente estudio. Es por este motivo que obtener una medición precisa de dicho tiempo permite realizar una comparación con otros algoritmos como *AODV* y *O-RPL*.

En *AODV* se ha establecido una variable que calcula el tiempo transcurrido desde que un nodo determinado envía un mensaje *RREQ* hasta que se recibe la respuesta en forma de *RREP*. Una vez calculados los tiempos para que todos los nodos de la red establezcan rutas, se calcula una media que permite conocer el tiempo de convergencia medio.

En el caso de *O-RPL* se ha utilizado el mecanismo implementado por Kermajani et al. [41], en el que miden el tiempo transcurrido desde el instante en el que una nueva raíz de un *DODAG* es iniciada hasta que todos los nodos se han unido a dicho *DODAG*. Debido al tipo de mecanismo implementado por *O-RPL* y al igual que Kermajani et al. hacen en su trabajo, hemos calculado 10 rutas diferentes cambiando la raíz del *DODAG*, calculando el tiempo medio para la formación de dichas rutas.

Finalmente en *DARAL* se implementó una variable que permitía calcular el tiempo desde el momento en que un nodo es iniciado hasta que pasa a estar en el estado *CONECTADO* o *AWAITING*. Dichos estados marcan el punto en el que un nodo pasa a estar conectado a la red y puede recibir mensajes, por ello una vez alcanzado alguno de esos estados se finaliza el cálculo. El resultado aquí mostrado es el cálculo medio para todos los nodos de la red.

A través de estos métodos se ha obtenido una medición del *tiempo de convergencia* para diferentes algoritmos de enrutamiento, lo que permite realizar una comparación entre ellos como la realizada a lo largo de las siguientes secciones.

6.2.2 Tamaño del escenario: PEQUEÑO

Este escenario se compone de 100 nodos, sería un escenario para una aplicación pequeña típica. En él se han analizado diferentes configuraciones en términos de densidad, utilizando diferentes tamaños de área de simulación. En estos análisis la frecuencia de envío o el tamaño de los mensajes no resulta de interés, ya que el objetivo principal es el análisis del tiempo de convergencia.

Los resultados para las diferentes configuraciones de densidad se muestran en la Tabla 6.4 y se agrupan visualmente en las Figuras 6.1, 6.2 y 6.3.

En este escenario se puede observar como *DARAL* presenta, en la mayoría de casos, el mejor y más fiable de los resultados. Para ND = 15, el tiempo medio es de 40.74s, mientras que *O-RPL* alcanza los 40.6s y *AODV* los 105.22s.

Por otro lado, en densidad media ND = 10, *DARAL* obtiene el mejor tiempo de convergencia con 38.91s, mientras que tanto *AODV* como *O-*

ND	Algoritmo	Tiempo de Convergencia (s)	Desviación Típica
5	O-RPL	916.05	982.93
5	AODV	122.59	55.53
5	DARAL	49.05	17.61
10	O-RPL	154.67	295.97
10	AODV	121.7	33.9
10	DARAL	38.91	15.81
15	O-RPL	40.6	151.51
15	AODV	105.22	33.28
15	DARAL	40.74	18.59

Tabla 6.4
Tiempo de Convergencia medio en escenarios pequeños.

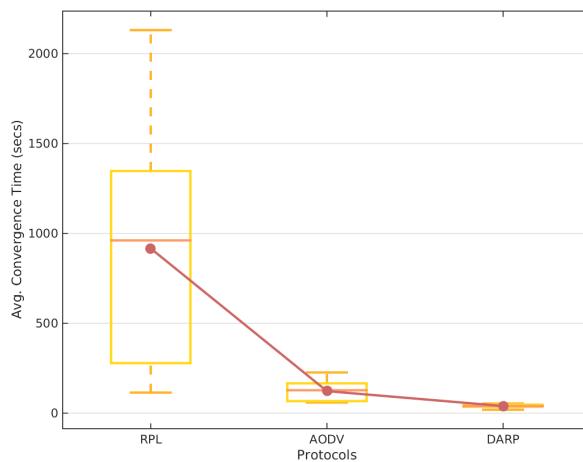


Figura 6.1
Tiempo de Convergencia medio en escenarios pequeños de baja densidad.

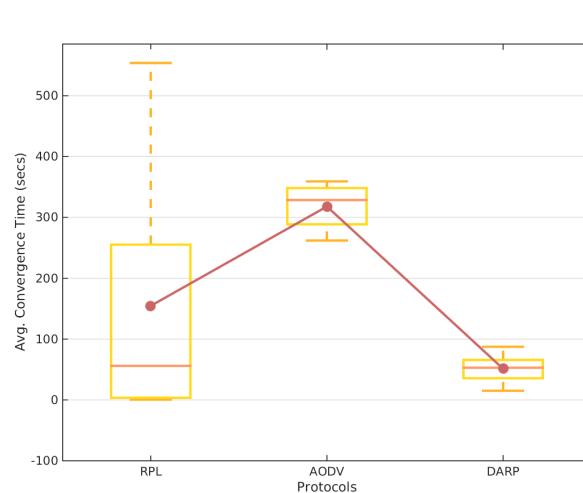


Figura 6.2
Tiempo de Convergencia medio en escenarios pequeños de media densidad.

RPL sobrepasan los 121s, reflejando el comportamiento detectado en la Sección 2.4.1.

Figura 6.3
Tiempo de Convergencia medio en escenarios pequeños de alta densidad.

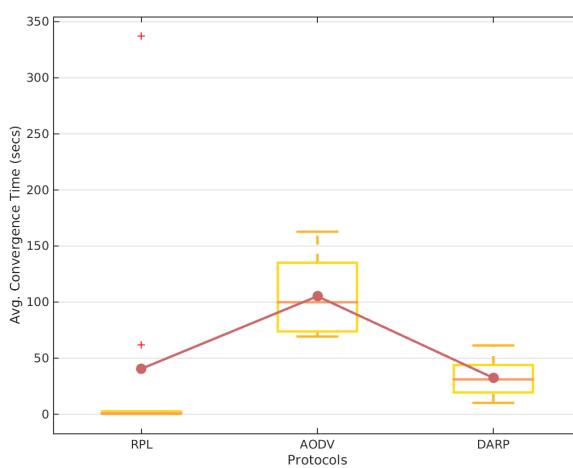


Tabla 6.5
Tiempo de Convergencia medio en escenarios medianos.

ND	Algoritmo	Tiempo de Convergencia (s)	Desviación Típica
5	O-RPL	1163.8	737.02
5	AODV	377.08	45.73
5	DARAL	60.51	23.39
10	O-RPL	156.11	253.97
10	AODV	317.95	36.15
10	DARAL	66.68	20.86
15	O-RPL	1.85	5.25
15	AODV	292.41	38.45
15	DARAL	61.02	13.02

En el caso de una baja densidad ND = 5, DARAL mantiene un tiempo de conexión de 39.61s, mientras que tanto AODV (122.59s) y O-RPL (916.05s) muestran un peor desempeño, justo como se esperaba.

6.2.3 Tamaño del escenario: MEDIANO

En esta ocasión, el escenario se compone de 200 nodos, lo que configuraría un escenario para una aplicación mediana típica. Se han analizado nuevamente diferentes configuraciones en términos de densidad, utilizando diferentes tamaños de área de simulación como recogía la Tabla 6.2. Como ya se ha comentado previamente, la frecuencia de envío o el tamaño de los mensajes no resulta de interés, ya que el objetivo principal es el análisis del tiempo de convergencia.

Los resultados para las diferentes configuraciones de densidad se muestran en la Tabla 6.5 y se agrupan visualmente en la serie de Figuras 6.4–6.6.

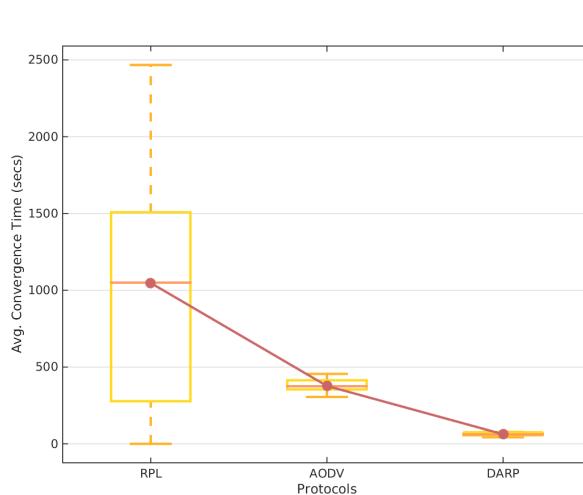


Figura 6.4
Tiempo de Convergencia medio en escenarios medianos de baja densidad.

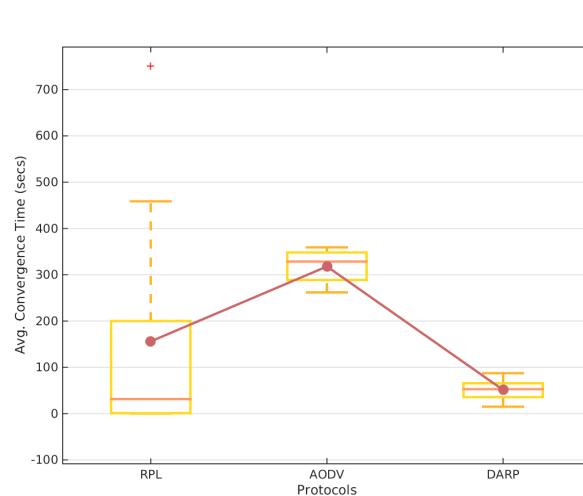


Figura 6.5
Tiempo de Convergencia medio en escenarios medianos de media densidad.

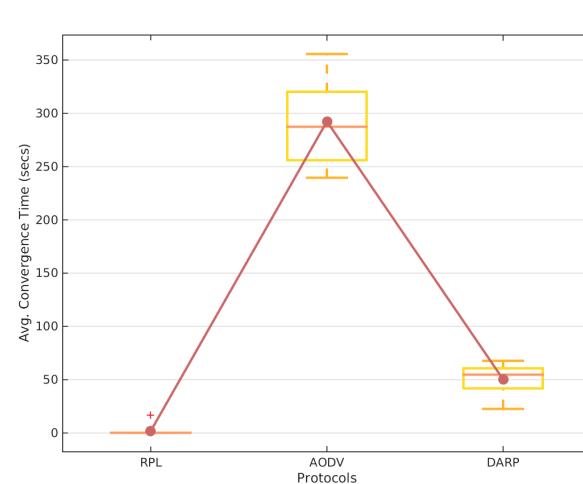


Figura 6.6
Tiempo de Convergencia medio en escenarios medianos de alta densidad.

Tabla 6.6
Tiempo de Convergencia medio en escenarios grandes.

ND	Algoritmo	Tiempo de Convergencia (s)	Desviación Típica
5	O-RPL	3022.7	2209.3
5	AODV	640.17	227.99
5	DARAL	90.27	11.18
10	O-RPL	91.28	196.67
10	AODV	688.87	35.46
10	DARAL	84.95	9.41
15	O-RPL	0.76	0.99
15	AODV	655.26	49.85
15	DARAL	98.43	11.98

Este escenario duplica el número de nodos del escenario PEQUEÑO, mostrando nuevamente tres comportamientos basados en la densidad de nodos. En alta densidad ND = 15, *O-RPL* muestra un comportamiento excelente de 1.85s, a continuación *DARAL* (61.02s) y *AODV* (292.41s). Este comportamiento por parte de *O-RPL* es esperado, aunque sorprende el excelente rendimiento. Por otro lado, *DARAL* mantiene un buen comportamiento a pesar del incremento de la densidad. Por el contrario *AODV* muestra un comportamiento irregular, debido a que por su funcionamiento, una mayor densidad de nodos, debería de mostrar una mayor eficiencia en términos de tiempo de convergencia.

Los resultados obtenidos para una densidad media, ND = 10, presentan nuevamente a *DARAL* como la mejor opción (66.68s), mientras que *O-RPL* muestra un comportamiento similar al del anterior escenario (156.11s), y *AODV* muestra un ligero empeoramiento del rendimiento (317.95s).

Para una baja densidad, ND = 5, *DARAL* muestra el mejor comportamiento, al igual que en la densidad media con 60.51s. Por otro lado, *AODV* y *O-RPL* arrojan un bajo rendimiento con 377.08s y 1163.8s, respectivamente.

6.2.4 Tamaño del escenario: GRANDE

Para finalizar el análisis del *Tiempo de Convergencia* se presenta el último caso de análisis basado en 400 nodos, lo que permite simular una aplicación grande típica. Como en los análisis previos se han utilizado diferentes configuraciones en términos de densidad, variando los tamaños de área de simulación. Los valores exactos utilizados pueden ser encontrados en la Tabla 6.2.

Los resultados para las diferentes configuraciones de densidad se muestran en la Tabla 6.6 y se agrupan visualmente en las Figuras 6.7–6.9.

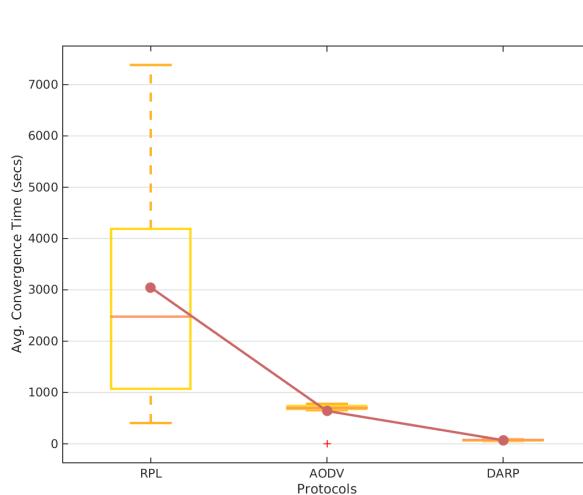


Figura 6.7
Tiempo de Convergencia medio en escenarios grandes de baja densidad.

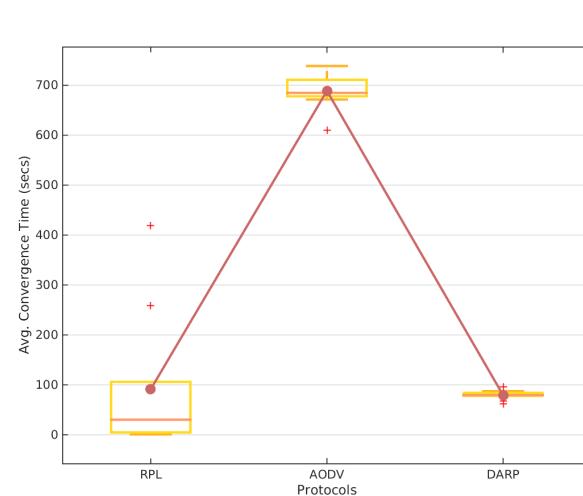


Figura 6.8
Tiempo de Convergencia medio en escenarios grandes de baja densidad.

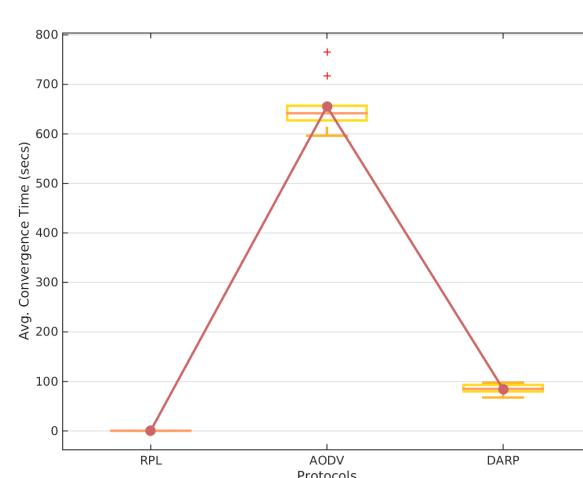


Figura 6.9
Tiempo de Convergencia medio en escenarios grandes de baja densidad.

El escenario grande se corresponde con un escenario en el que se vuelve a duplicar el número de nodos, en este caso con respecto al escenario MEDIANO. Como en los anteriores escenarios, se han analizado las tres densidades. Bajo condiciones de alta densidad $ND = 15$, los resultados de *O-RPL* resultan, con diferencia, los mejores, alcanzando los 0.76s. A continuación *DARAL* muestra unos valores compensados con respecto a los anteriores escenarios y al incremento en el número de nodos con 98.43s, mientras que *AODV* arroja unos pobres resultados alcanzando 655.26s.

En el caso de la densidad media $ND = 10$, como en los anteriores escenarios, *DARAL* presenta un mejor comportamiento que *O-RPL*, requiriendo 84.95s por los 91.28s de *O-RPL*. Nuevamente *AODV* queda por detrás de ambos necesitando hasta 688.87s.

Por último, el escenario de baja densidad $ND = 5$, nos muestra unos resultados muy interesantes, en los que *DARAL* presenta la menor dispersión y además solo requiere de 90.27s de media, *AODV* 640.17s y *O-RPL* unos sorprendentes 3022.7s. Esto se debe a la baja concentración y el amplio número de nodos, lo que provoca numerosos ciclos y pérdida de mensajes tanto para *AODV* como para *O-RPL*, significando además un incremento del tiempo de convergencia.

6.3 RESULTADOS EXPERIMENTALES PARA EL ANÁLISIS DEL NÚMERO DE MENSAJES DE CONTROL ENVIADOS

Esta sección presenta los resultados obtenidos en términos de número de mensajes de control enviados del algoritmo propuesto en el Capítulo 4 en comparación con *AODV* y *O-RPL*.

La información analizada es presentada como en la Sección previa, mediante gráficos de barras y tablas de datos, de forma que se puedan estudiar de forma pormenorizada todos los resultados obtenidos.

En esta primera fase de los resultados experimentales, se han llevado a cabo una serie de pruebas basadas en los escenarios propuestos en la Tabla 6.2. A continuación se presentan los resultados organizados en base al tamaño del escenario.

6.3.1 Cálculo del número de mensajes de control enviados

El cálculo del número de mensajes de control se ha realizado de forma diferente para cada uno de los algoritmos de enrutamiento analizados.

ND	Algoritmo	Num. de mensajes de control	Desviación Típica
5	O-RPL	1010	120.48
5	AODV	1555.27	9689.08
5	DARAL	201.42	55.55
10	O-RPL	407.82	161.11
10	AODV	1674.25	10418.46
10	DARAL	211.75	32.03
15	O-RPL	245.85	31.81
15	AODV	1739.48	9723.29
15	DARAL	201.39	39.8

Tabla 6.7
Número de mensajes de control medio en escenarios pequeños.

En *AODV* se ha establecido una variable que cuenta el número de mensajes enviados para cada petición *RREQ*, hasta que se recibe un *RREP*.

En *O-RPL* se ha utilizado un mecanismo similar al de *AODV*, estableciendo una variable que cuenta los *DAO* que envían los nodos para conectarse a un grafo.

En *DARAL* se implementó una variable que permitía contar todos los mensajes de control enviados hasta que un nodo cambia de estado a *CONECTADO* o *AWAITING*, momento en el que se termina dicha cuenta.

Mediante estas variables, se ha extraído la información de cada protocolo, permitiendo la realización del siguiente estudio comparativo.

6.3.2 Tamaño del escenario: PEQUEÑO

Para este análisis se ha usado la misma configuración que en la Sección 6.2.2. Los resultados de las diferentes configuraciones de densidad se muestran en la Tabla 6.7 y se agrupan visualmente en las Figuras 6.10, 6.12 y 6.14. Debido a la enorme diferencia en los resultados, se ha decidido eliminar *AODV* de la representación gráfica para obtener unos resultados más claros, estas representaciones se muestran en las Figuras 6.11, 6.13 y 6.15.

Los resultados muestran el número de mensajes de control medio enviado por un nodo para conectarse a una red para el escenario PEQUEÑO. *DARAL* arroja el mejor rendimiento para todas las densidades analizadas con un valor muy estable entre 201.39 – 211.75 mensajes, mientras que *O-RPL* obtiene peores resultados en este análisis, debido al menor número de nodos disponibles, aún así sus valores se mueven en el rango 245.85 – 1010. Por último cabe destacar el elevado número de mensajes generados por los nodos en *AODV*, con valores muy por encima de los 1500 mensajes de media. Este hecho se encuentra

Figura 6.10
Número medio de mensajes de control enviado para escenarios pequeños de baja densidad, considerando AODV.

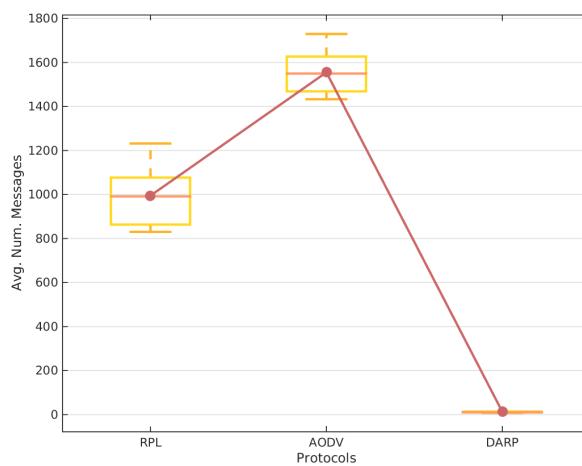


Figura 6.11
Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios pequeños de baja densidad.

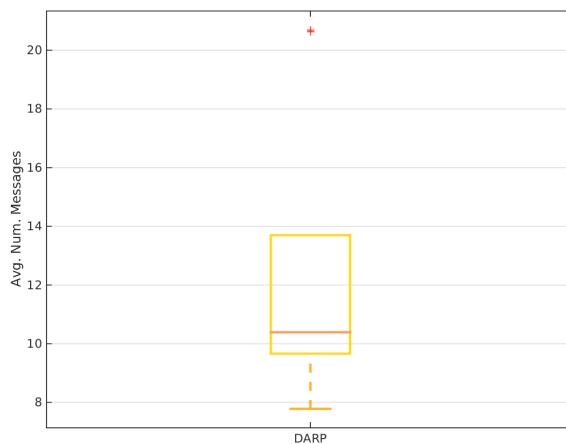
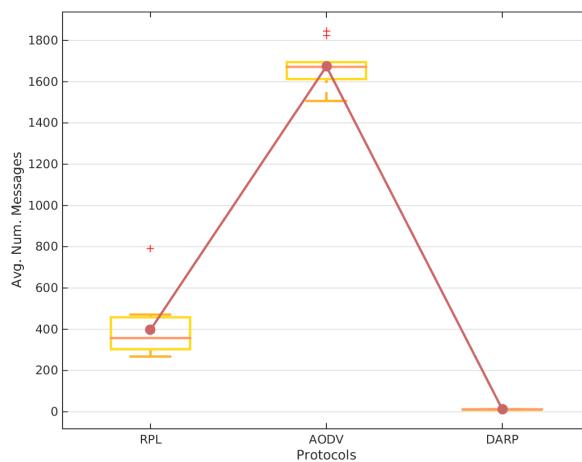


Figura 6.12
Número medio de mensajes de control enviado para escenarios pequeños de media densidad, considerando AODV.



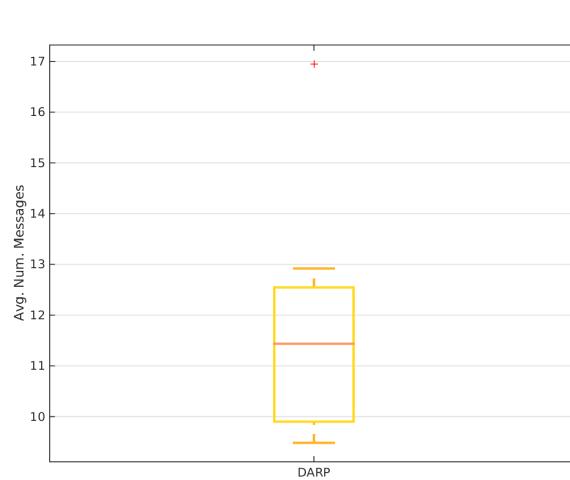


Figura 6.13
Análisis detallado
del número medio de
mensajes de control
enviado por DARP en
escenarios pequeños de
media densidad.

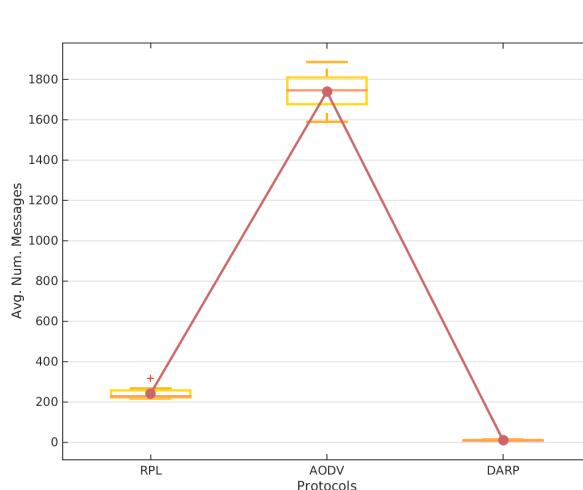


Figura 6.14
Número medio de
mensajes de control
enviado para escena-
rios pequeños de alta
densidad, consideran-
do AODV.

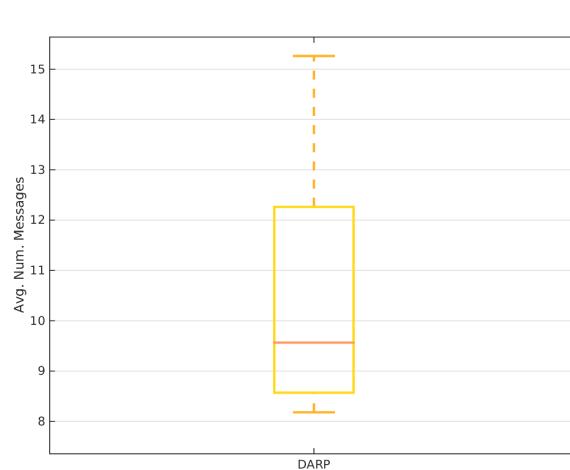
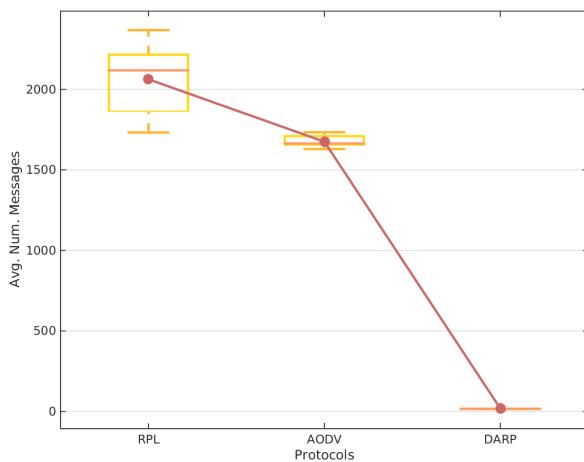


Figura 6.15
Análisis detallado
del número medio de
mensajes de control
enviado por DARP en
escenarios pequeños de
alta densidad.

Tabla 6.8
Número de mensajes de control medio en escenarios medianos.

ND	Algoritmo	Num. de mensajes de control	Desviación Típica
5	O-RPL	2056.1	217.99
5	AODV	1674.66	6413.87
5	DARAL	168.96	37.30
10	O-RPL	815.12	206.99
10	AODV	1678.56	9386.83
10	DARAL	149.90	25.98
15	O-RPL	496.63	85.01
15	AODV	1793.78	8606.69
15	DARAL	186.53	29.27

Figura 6.16
Número medio de mensajes de control enviado para escenarios medianos de baja densidad, considerando AODV.



relacionado con el comportamiento y funcionamiento del mecanismo de creación de rutas de *AODV*, el cual requiere el procesamiento y generación de un nuevo mensaje para el siguiente salto.

6.3.3 Tamaño del escenario: MEDIANO

En esta ocasión se ha utilizado la configuración usada en la Sección 6.2.3. Los resultados de las diferentes configuraciones de densidad se muestran en la Tabla 6.8 y se agrupan visualmente en las Figuras 6.16–6.21. Los resultados recogidos en esta sección presentan los resultados del escenario MEDIANO. DARAL arroja de nuevo un resultado muy estable, en el rango de los 168.96 – 186.53 mensajes, mientras que *O-RPL* obtiene un resultado de 496.63 mensajes en condiciones de alta densidad ND = 15, quedando por detrás de DARAL. Por último destacar de nuevo, el pobre rendimiento de *AODV*, con valores superiores a los 1674 mensajes.

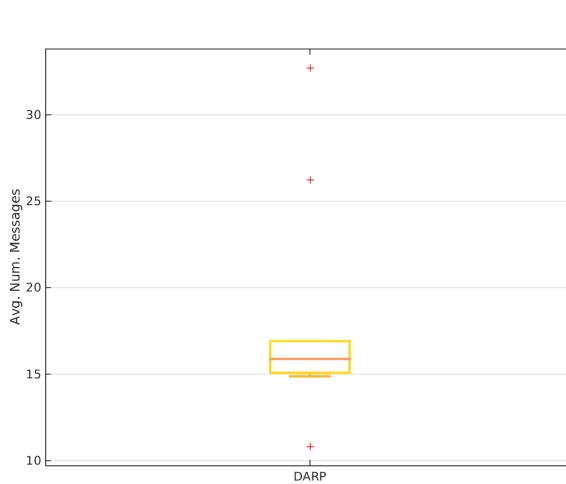


Figura 6.17
Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios medianos de baja densidad.

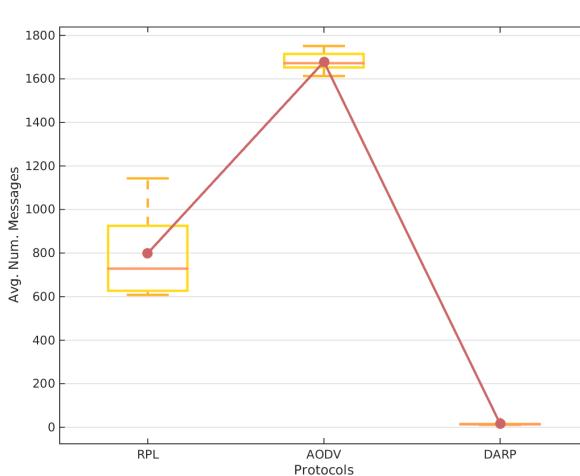


Figura 6.18
Número medio de mensajes de control enviado para escenarios medianos de media densidad, considerando AODV.

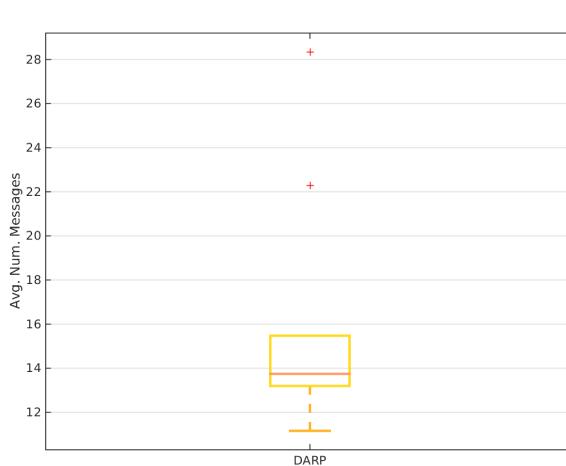


Figura 6.19
Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios medianos de media densidad.

Figura 6.20
Número medio de mensajes de control enviado para escenarios medianos de alta densidad, considerando AODV.

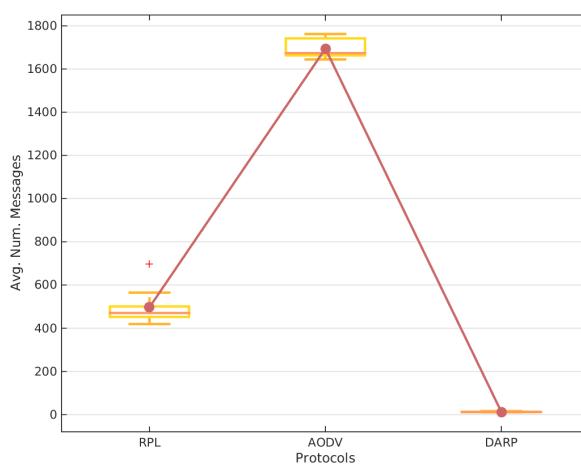
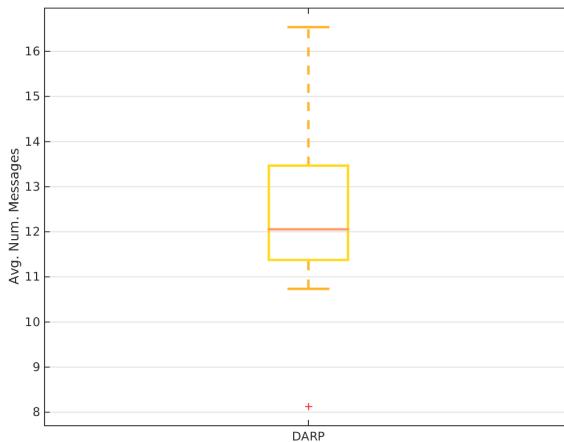


Figura 6.21
Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios medianos de alta densidad.



6.3.4 Tamaño del escenario: GRANDE

Este análisis hace uso de la configuración aplicada en la Sección 6.2.4, cuyos resultados para las diferentes configuraciones de densidad se muestran en la Tabla 6.9 y se agrupan visualmente en la serie de Figuras 6.22–6.27.

En este escenario, AODV arroja nuevamente unos resultados por encima de lo esperado, con valores superiores a 1694.73 mensajes, mientras que los resultados de DARAL y O-RPL continúan siguiendo el mismo patrón que en los escenarios anteriores. O-RPL presenta unos resultados bastante pobres en condiciones de baja y media densidad, 5342.2 y 1722.5 mensajes, respectivamente, mejorando en escenarios de alta densidad, alcanzando 1055.2 mensajes. Por otro lado, DARAL presenta un mejor rendimiento de forma generalizada, alcanzando un resultado de 47.29 – 112.59 mensajes.

ND	Algoritmo	Num. de mensajes de control	Desviación Típica
5	O-RPL	5343.2	472.64
5	AODV	1739.48	10750.61
5	DARAL	112.59	27.67
10	O-RPL	1722.5	473.59
10	AODV	1694.73	11821.05
10	DARAL	47.29	9.7
15	O-RPL	1055.2	81.77
15	AODV	1777.7	14901.79
15	DARAL	96.04	33.36

Tabla 6.9
Número de mensajes de control medio en escenarios grandes.

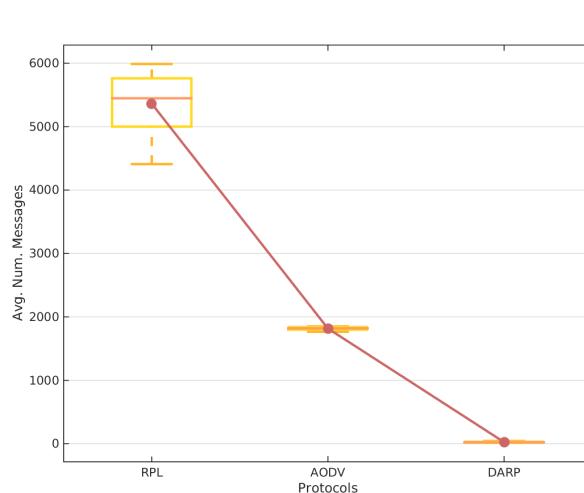


Figura 6.22
Número medio de mensajes de control enviado para escenarios grandes de baja densidad, considerando AODV.

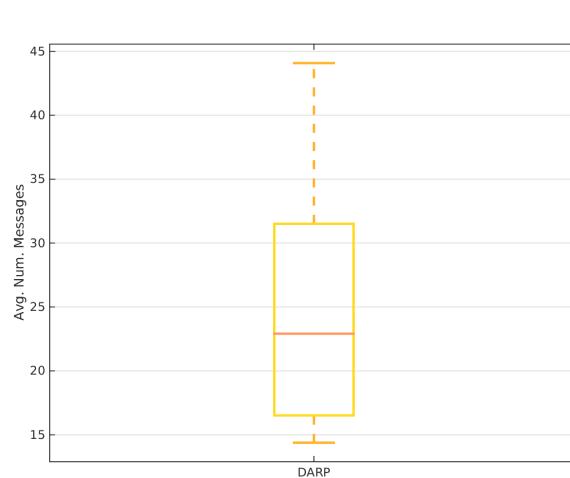


Figura 6.23
Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios grandes de baja densidad.

Figura 6.24
Número medio de mensajes de control enviado para escenarios grandes de media densidad, considerando AODV.

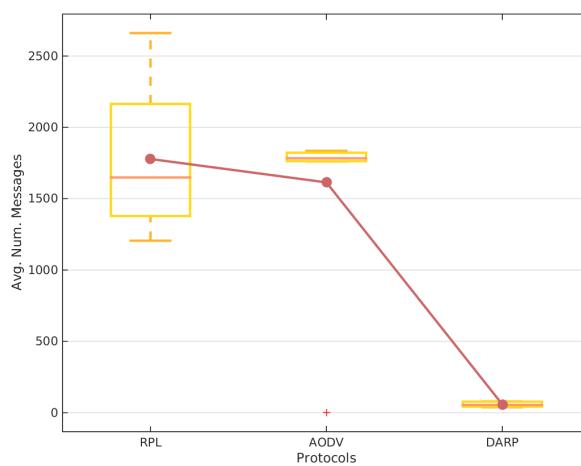


Figura 6.25
Análisis detallado del número medio de mensajes de control enviado por DARP en escenarios grandes de media densidad.

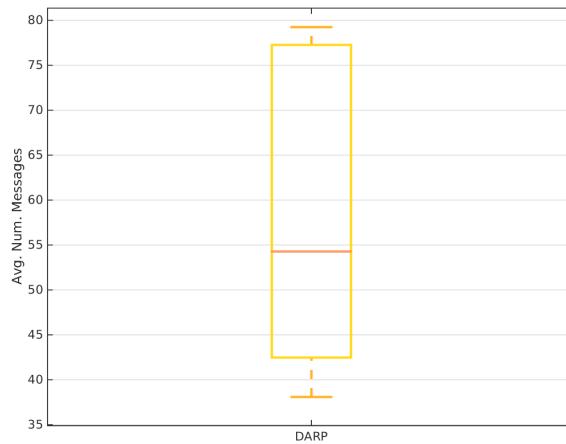
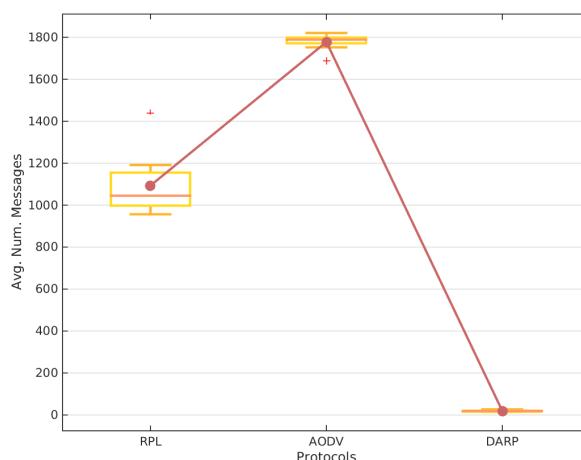


Figura 6.26
Número medio de mensajes de control enviado para escenarios grandes de alta densidad, considerando AODV.



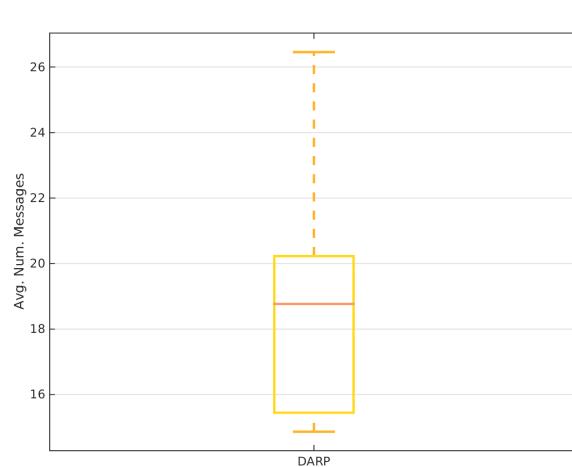


Figura 6.27
Análisis detallado
del número medio de
mensajes de control
enviado por DARP en
escenarios grandes de
alta densidad

6.4 CONCLUSIONES

Tras finalizar las pruebas de los diferentes algoritmos de enrutamiento, puede destacarse lo siguiente.

- El uso del tiempo de convergencia y el número de mensajes de control como estimadores del dinamismo resulta de interés para discernir el comportamiento futuro de un algoritmo.
- El uso de algoritmos de enrutamiento basados en grafos obtiene mejores resultados en términos de tiempo de convergencia cuanto mayor es la densidad de nodos.
- En los casos con densidad media, el incremento en el número de nodos tiende a mejorar los resultados de los algoritmos basados en grafos como *O-RPL*.
- *DARAL* muestra los resultados más estables con independencia del escenario y la densidad de nodos presentes.
- *DARAL* muestra los mejores resultados en situaciones de baja y media densidad.
- Se observa una relación directa entre el incremento en el tiempo de convergencia y un mayor número de mensajes de control enviados. Esta relación permite trazar una relación directa de ambos parámetros con la agilidad en la formación de las redes.

7

EVALUACIÓN DE OTRAS CARACTERÍSTICAS ADICIONALES

ÍNDICE

7.1	Análisis de la escalabilidad	125
7.1.1	Escenarios Muy Grandes	126
7.1.2	Escenarios Enormes	128
7.2	Análisis del consumo de energía durante la fase de conexión	130
7.3	Análisis del aprovechamiento del canal	131
7.4	Análisis de la minimización del número de colisiones	133
7.5	Análisis de la respuesta ante eventos dinámicos	133

El presente capítulo tiene como objetivo desarrollar una serie de análisis que permitan evaluar el cumplimiento de los diferentes objetivos de diseño adicionales expuestos en el Capítulo 4, así como validar algunas de las características esperadas en un protocolo WSN.

El capítulo se divide en 4 secciones en las que se estudia la *escalabilidad*, el *consumo de energía*, el *aprovechamiento del canal*, la *minimización de las colisiones* y el *dinamismo*, todo ello basado en diferentes tests.

La configuración de DARP para los estudios realizados en este capítulo está basada en la configuración presentada en el Capítulo 5. Como en análisis previos, se han desarrollado 10 iteraciones en escenarios generados aleatoriamente.

7.1 ANÁLISIS DE LA ESCALABILIDAD

El presente estudio, compara el rendimiento obtenido por los protocolos en diferentes escenarios, donde se ha escalado el tamaño a dos y cuatro veces los escenarios previamente definidos como grandes. Esto resulta en escenarios de gran tamaño con 800 nodos o directamente enormes con 1600 nodos. Ambos estudios se detallan en diferentes sub-secciones para facilitar su análisis.

Figura 7.1
Comparación del tiempo de convergencia en escenarios muy grandes para los diferentes protocolos.

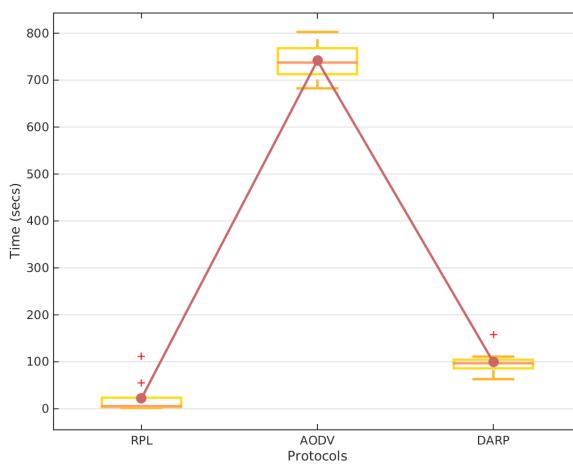
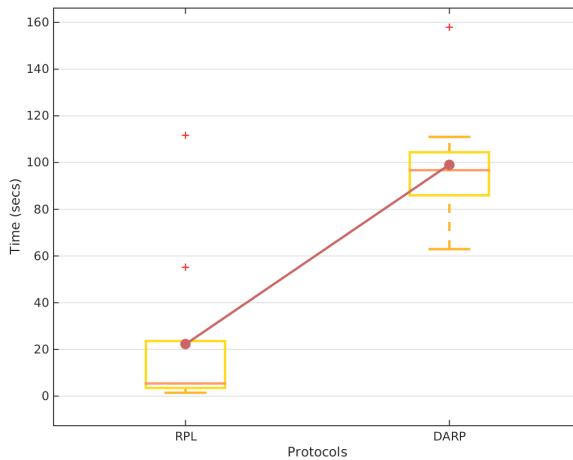


Figura 7.2
Comparación detallada entre O-RPL y DARP del tiempo de convergencia en escenarios muy grandes.



7.1.1 Escenarios Muy Grandes

Para el presente estudio se han generado 10 escenarios aleatorios compuestos cada uno de 800 nodos, con una densidad media $ND = 10$. Los valores de configuración adoptados han sido los mismos que se aplicaron en el Capítulo 6.

Como en estudios previos, se han analizado O-RPL, AODV y DARP, estudiando tanto el *tiempo de convergencia* como el *número de mensajes de control enviados durante la fase de conexión*.

En la Figura 7.1 se estudia el comportamiento de todos los protocolos en escenarios muy grandes. En ella se puede apreciar como AODV muestra un *tiempo de convergencia* alto, aún a pesar de contar con muchos enlaces disponibles.

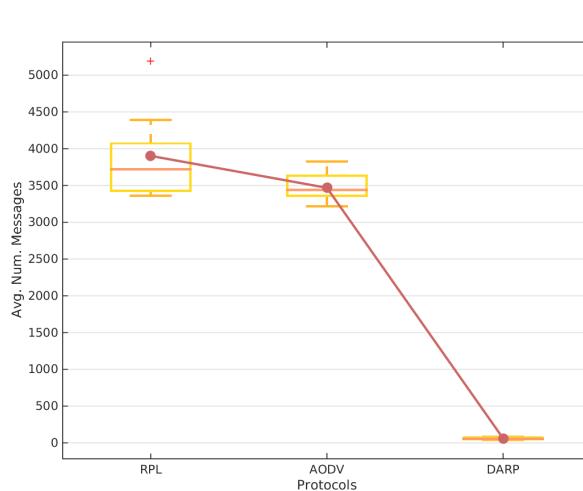


Figura 7.3
Comparación del número de mensajes de control durante la fase de conexión en escenarios muy grandes para los diferentes protocolos.

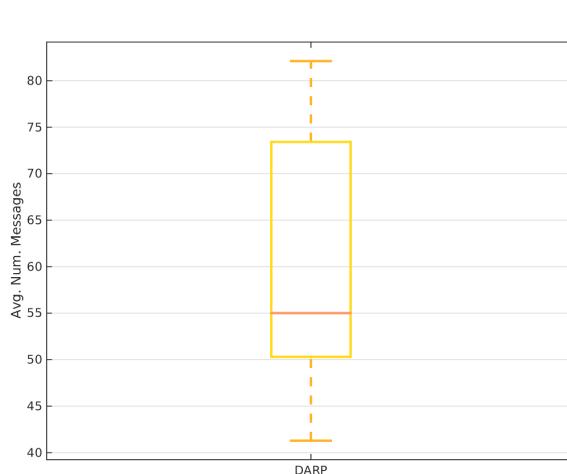


Figura 7.4
Análisis del número de mensajes de control durante la fase de conexión en escenarios muy grandes de DARP.

La Figura 7.2, presenta nuevamente la comparativa del *tiempo de convergencia* de forma más detallada, dejando fuera de la misma a AODV. En esta se puede apreciar como O-RPL y DARP obtienen buenos resultados, aunque O-RPL mejora considerablemente a todos los demás protocolos.

Otra de las características estudiadas a lo largo de esta Tesis Doctoral ha sido el *número de mensajes de control enviados durante la fase de conexión*. En la Figura 7.2 se pueden apreciar los resultados generados por el presente estudio para todos los protocolos.

En dicha figura se puede observar como DARP mejora de forma considerable a los otros protocolos, para observar los resultados en detalle se muestra la Figura 7.4 con el análisis de DARP en detalle.

Finalmente, la Figura 7.5, muestra el consumo de energía entre AODV y DARP. O-RPL queda fuera de este análisis a causa de que el modelo

Figura 7.5
Análisis del Consumo de energía medio por nodo para escenarios muy grandes.

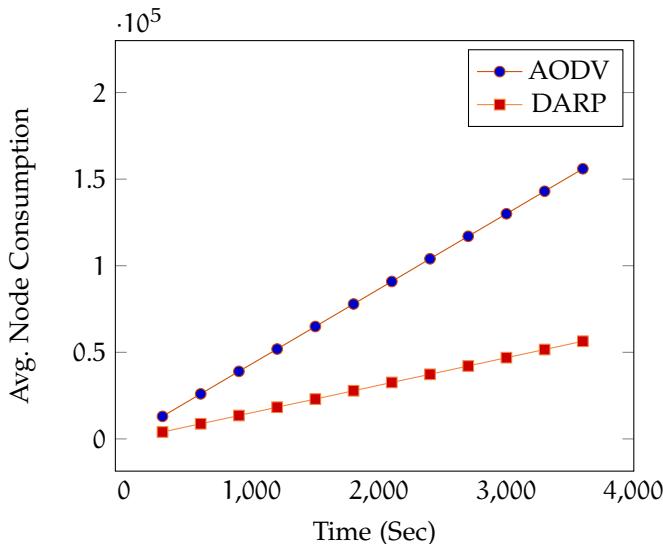
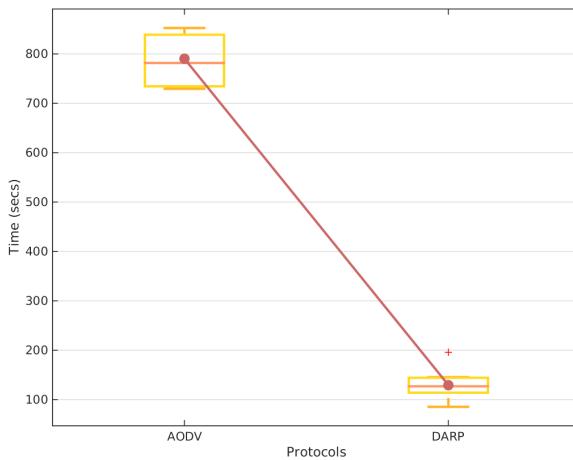


Figura 7.6
Comparación del tiempo de convergencia en escenarios enormes para los diferentes protocolos.



de simulación utilizado no soporta el análisis de consumo energético durante simulaciones completas.

En este último análisis, se puede apreciar cómo el consumo de *AODV* es mayor desde el inicio, y unido a la cantidad de mensajes de control enviados, conlleva un mayor consumo de forma general durante toda la simulación.

7.1.2 Escenarios Enormes

Este tipo de escenario cuadriplica el tamaño de escenario considerado grande en esta Tesis Doctoral, alcanzando los 1600 nodos. Nuevamente

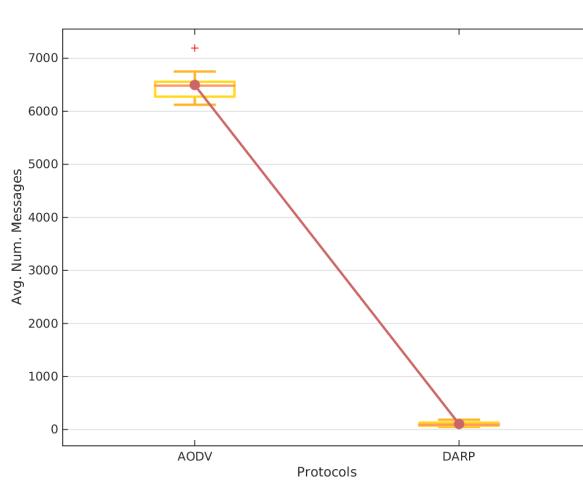


Figura 7.7
Comparación del número de mensajes de control durante la fase de conexión en escenarios enormes para los diferentes protocolos.

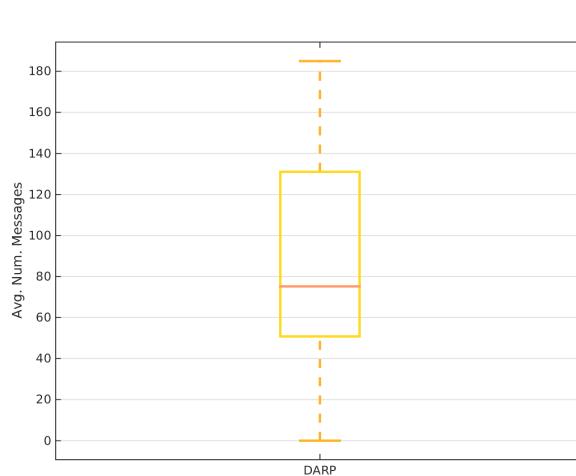


Figura 7.8
Comparación del número de mensajes de control durante la fase de conexión en escenarios enormes para los diferentes protocolos.

se ha utilizado una densidad media $ND = 10$ y se han simulado 10 escenarios generados aleatoriamente.

Los protocolos analizados en este estudio son *DARP* y *AODV*, debido a que no se ha podido desarrollar un análisis de *O-RPL*, a causa de los requisitos necesarios para simular escenarios enormes.

En la Figura 7.6 se presenta el análisis del *tiempo de convergencia* entre *AODV* y *DARP* para escenarios enormes, y se puede apreciar, como en el escenario previo, que *DARP* mejora ampliamente los resultados de *AODV*.

A continuación, la Figura 7.7 muestra el resultado en términos de *número de mensajes de control enviados durante la fase de conexión*. En este estudio se aprecia nuevamente cómo *DARP* mejora aún más los resultados de *AODV*. Debido a que los resultados de *DARP* se aprecian vagamente, la Figura 7.8 los muestra en detalle.

Figura 7.9
Análisis del Consumo de energía medio por nodo para escenarios enormes.

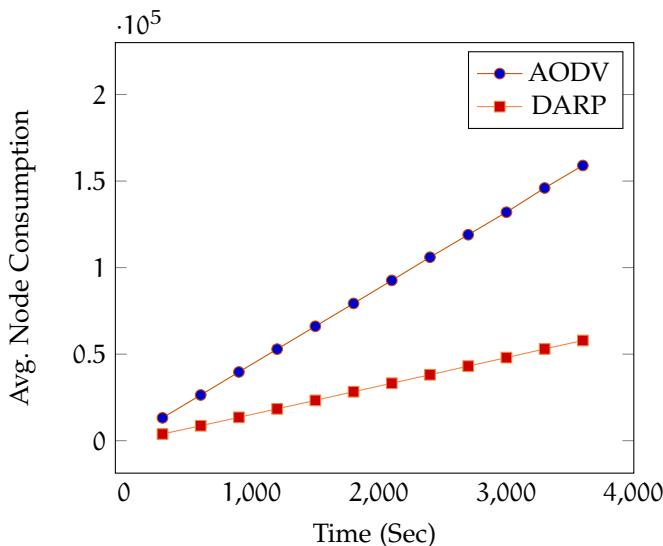
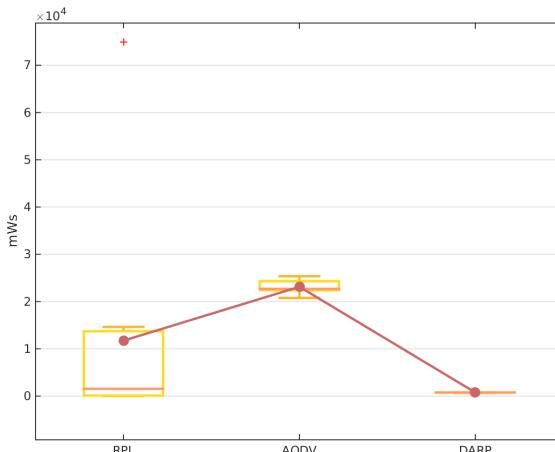


Figura 7.10
Comparación de los consumos energéticos de los protocolos en la fase de conexión.



Finalmente, la Figura 7.9 muestra los resultados relativos al consumo energético de los protocolos. Nuevamente y como en la Sección 7.1.1, *DARP* mejora sustancialmente a *AODV*.

7.2 ANÁLISIS DEL CONSUMO DE ENERGÍA DURANTE LA FASE DE CONEXIÓN

Este estudio del consumo de energía analiza el consumo de los diferentes protocolos durante la fase de formación de la red, para redes grandes con 400 nodos, como las utilizadas en el Capítulo 6, y con una configuración de densidad de nodos media, equivalente a $ND = 10$.

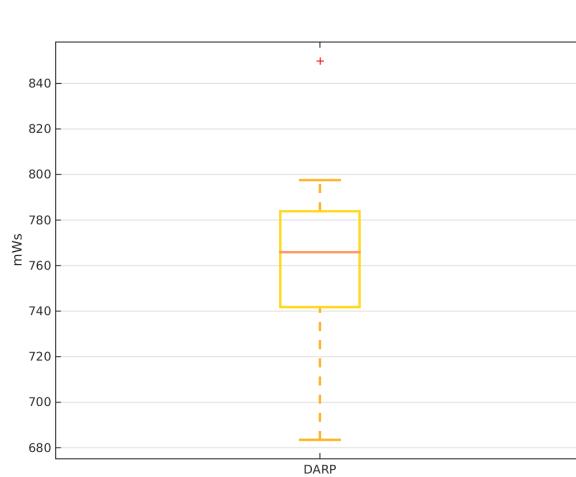


Figura 7.11
Análisis del consumo energético de DARP detallado.

En este estudio se comparan todos los protocolos disponibles. La Figura 7.10 muestra los resultados de la comparación. Adicionalmente, la Figura 7.11 muestra el consumo de DARP detalladamente, dado que en la figura previa, no puede ser correctamente observado debido a la escala.

Como se puede observar, O-RPL mejora el consumo energético de AODV, aunque AODV presenta una menor dispersión que O-RPL. Aún así DARP mejora en todos los aspectos al consumo, tanto de O-RPL como de AODV. DARP presenta los valores más reducidos de media y mediana, los menores máximo y mínimo, así como la menor dispersión.

Este resultado es la consecuencia directa de lo observado en el Capítulo 6, donde DARP cosechaba los mejores resultados en términos de tiempo de conexión y número de mensajes de control. Este hecho refleja la relación directa entre el tiempo de funcionamiento y el consumo de energía, por lo que a menor tiempo de convergencia, la red presenta un menor consumo medio.

7.3 ANÁLISIS DEL APROVECHAMIENTO DEL CANAL

El estudio del aprovechamiento del canal de los diferentes protocolos se ha realizado simulando escenarios grandes con 400 nodos bajo condiciones de densidad media $ND = 10$, como los simulados en el Capítulo 6.

En este estudio se han comparado los protocolos que soportan la simulación de un escenario completo durante 3600s, es decir AODV y DARP. En las Figuras 7.12 y 7.13 se analiza la duración media de los backoffs de CSMA. Estos tiempos son una medida del tiempo necesario

Figura 7.12
Comparativa del aprovechamiento del canal de los diferentes protocolos.

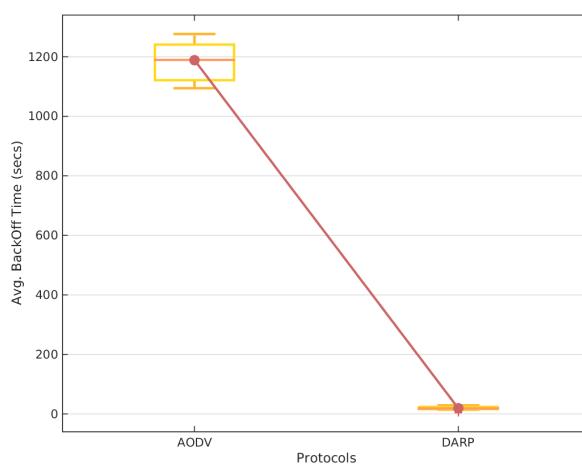
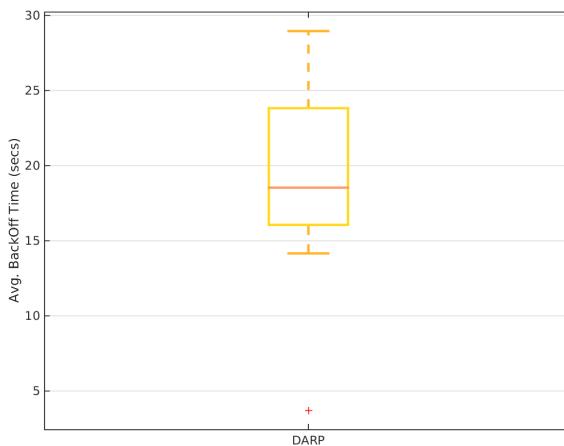


Figura 7.13
Análisis del aprovechamiento del canal de DARP detallado.



para que un nodo pueda enviar un mensaje a través del canal, y es una medida directa tomada a partir de la capa MAC del estándar IEEE 802.15.4. Las Figuras 7.12 y 7.13 muestran la comparativa en duración de las *backoffs* y el análisis de la duración para DARP exclusivamente, debido a la amplia diferencia a partir de la comparativa que no permite observar correctamente la duración en el caso de DARP.

Como se puede observar DARP mejora considerablemente a AODV, lo cuál significa que en media, DARP aprovecha mejor el canal disponible, necesitando menos tiempo de espera para poder acceder al canal y enviar los mensajes.

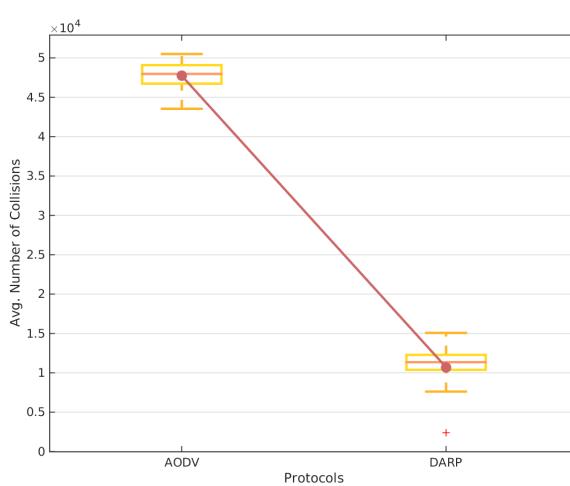


Figura 7.14
Comparativa del número de colisiones de los protocolos.

7.4 ANÁLISIS DE LA MINIMIZACIÓN DEL NÚMERO DE COLISIONES

El estudio del número de colisiones a nivel de capa MAC, permite obtener una métrica independiente de las estadísticas propias de cada protocolo, mostrando el comportamiento de los protocolos a la hora de acceder al canal.

Para el presente estudio, cuyos resultados se presentan en la Figura 7.14, se han utilizado los resultados de simular escenarios grandes de 400 nodos bajo condiciones de densidad media $ND = 10$ durante 3600s. Debido a que el análisis se ha realizado a partir de una simulación completa, *O-RPL* no ha podido ser analizado, a causa del tipo de implementación disponible.

Esta comparativa muestra nuevamente un mejor rendimiento de *DARP* que de *AODV*, obteniendo un número medio de colisiones mucho más bajo, lo que repercute tanto en un menor consumo de energía como en una mayor tasa de envío de paquetes.

7.5 ANÁLISIS DE LA RESPUESTA ANTE EVENTOS DINÁMICOS

Esta sección presenta un estudio acerca de cómo se comportan los protocolos ante eventos dinámicos, en los que los nodos se desconectan y se conectan de forma aleatoria y dinámica durante toda la simulación.

Este estudio se basa en un escenario grande, de 400 nodos, con densidad media $ND = 10$, donde un número aleatorio de nodos se verá afectado por problemas de suministro de energía, desconectándose de

Figura 7.15
Comparativa del *tiempo de convergencia* en un escenario dinámico.

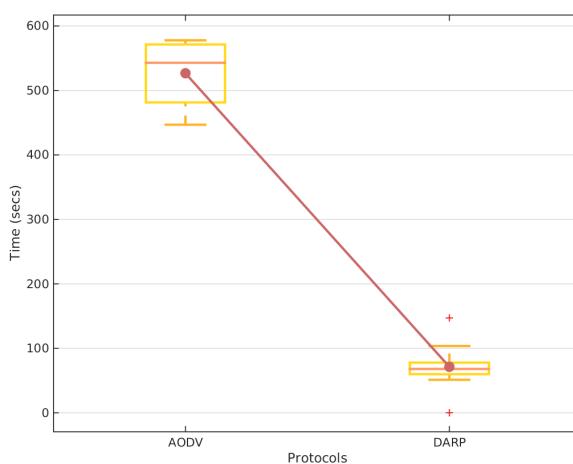
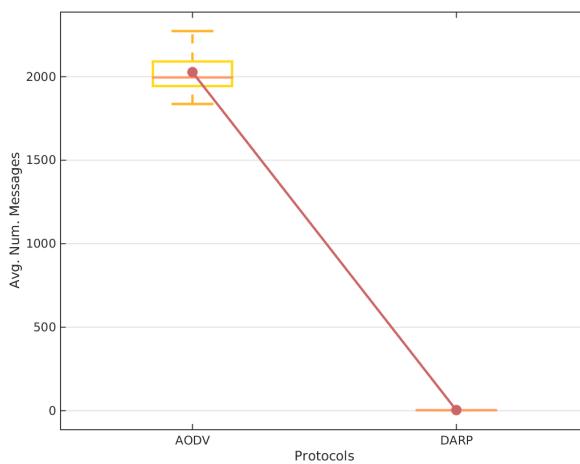


Figura 7.16
Comparativa del *número de mensajes de control enviados durante la fase de conexión* en un escenario dinámico.



la red. Por otro lado otro número aleatorio de nodos son iniciados con un determinado retardo, todo ello generado aleatoriamente, lo cual conlleva que deban conectarse a la red más tarde, resultando en que todo el sistema debe adaptarse.

Debido a la implementación de *O-RPL*, no ha sido posible realizar este estudio, por lo que la comparativa será exclusivamente desarrollada entre *AODV* y *DARP*. Los parámetros de configuración aplicados son los mismos que se detallaron en el Capítulo 6.

En la Figura 7.15 se observan los resultados relativos al *tiempo de convergencia*, donde se observa cómo la media de *AODV* mejora respecto a los resultados mostrados en el Capítulo 6, aunque siguen siendo peores que los mostrados por *DARP*.

El estudio del *número de mensajes de control enviados durante la fase de conexión* se muestra en la Figura 7.16 y en él se puede apreciar como

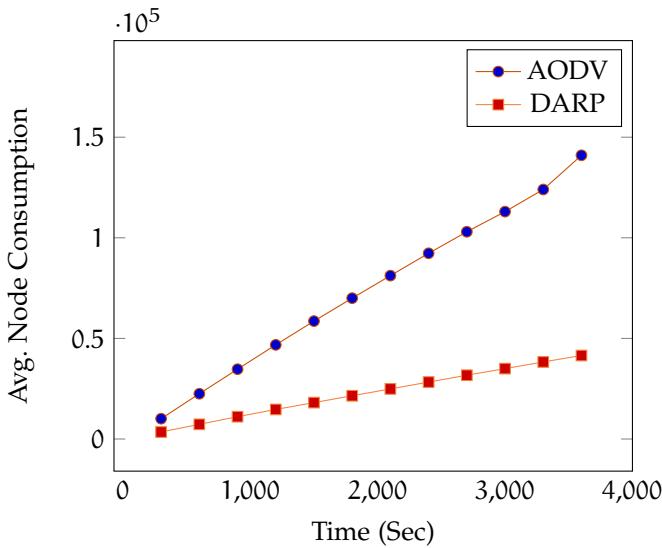


Figura 7.17
Análisis del Consumo de energía medio por nodo para escenarios dinámicos.

nuevamente *DARP* presenta mejores resultados que *AODV*, aunque en esta ocasión ambos protocolos empeoran con respecto a los resultado obtenidos en el escenario estático analizado en el Capítulo 6.

El consumo energético mostrado en la Figura 7.17, presenta nuevamente un mejor consumo por parte de *DARP*, lo que viene a confirmar la mejora de rendimiento con respecto a *AODV*.

Parte VI

DISCUSIÓN Y CONCLUSIONES

8

DISCUSIÓN

ÍNDICE

- 8.1 Discusión sobre la configuración óptima del protocolo de enrutamiento para grandes escenarios [140](#)
 - 8.2 Discusión sobre el protocolo de comunicaciones y los algoritmos propuestos [141](#)
 - 8.2.1 Mejora en la creación de rutas en grandes escenarios [142](#)
 - 8.3 Discusión sobre la evaluación de características adicionales de DARP [143](#)
 - 8.4 Conclusiones [145](#)
-

En este capítulo se analizan y discuten los resultados obtenidos en la presente Tesis Doctoral. El protocolo de enrutamiento desarrollado, así como los algoritmos en los que se basa, ofrecen un nuevo aporte a la comunidad investigadora en el área de las WSN. Con el objetivo de facilitar al lector la correcta comprensión de este capítulo de discusiones, éstas se han llevado a cabo usando el siguiente orden:

Primero, se discute acerca de la configuración óptima del protocolo propuesto, discutiendo la influencia de cada parámetro sobre el comportamiento en un escenario de grandes dimensiones.

Segundo, se discuten los aspectos relativos a los algoritmos propuestos para grandes escenarios, todos ellos basados en el uso de la calidad del enlace, el tiempo de convergencia y el número de mensajes de control enviados. Se tienen *DARAL* y *DRSP*, cuyo desarrollo representa la mayor contribución de este trabajo, así como la unión entre ambos y el estándar *IEEE 802.15.4*, que conforma la propuesta de protocolo *DARP* también recogida en este documento. Estas propuestas se basan en el concepto de dinamismo para grandes escenarios, utilizado en esta Tesis Doctoral.

Tercero, se discute acerca de una serie de características adicionales que se han desarrollado para satisfacer algunas de las características deseables en los protocolos de enrutamiento hoy por hoy.

8.1 DISCUSIÓN SOBRE LA CONFIGURACIÓN ÓPTIMA DEL PROTOCOLO DE ENRUTAMIENTO PARA GRANDES ESCENARIOS

La principal aportación de esta Tesis Doctoral es el protocolo de enrutamiento *DARP*, el cuál como ya se comentó en el Capítulo 4, se basa en un algoritmo de enrutamiento específico, *DARAL*, el cual también es parte de las aportaciones científicas de este trabajo.

Este algoritmo presenta diferentes parámetros de configuración, los cuales permiten una amplia variedad de configuraciones, haciendo el protocolo adaptable a diferentes escenarios, con diferentes características. Dado que el objetivo de esta Tesis son las grandes infraestructuras, que generalmente poseen densidades bajas o medias, como se discutió en el Capítulo 2, resultaba necesario ajustar una configuración apropiada para ellas.

En el Capítulo 5 se realiza un proceso iterativo de análisis que permite alcanzar la convergencia en torno a unos parámetros de configuración. Dicho proceso iterativo ha resultado en una configuración suficientemente estable tras tres iteraciones, tal y como se presenta a lo largo del Capítulo 5 y en la Tabla 5.4 a modo de resumen.

Estos parámetros aplicados a las grandes infraestructuras nos presentan ciertas características interesantes, como por ejemplo, la necesidad de fijar un elevado umbral base ($TH_{baselevel}$), lo que permite descartar determinados enlaces, asegurando que los nodos que se conecten entre sí tengan mayores posibilidades de conseguir completar el envío de mensajes. Importante resulta también el hecho de que el umbral que permite fijar el rol de los nodos (TH_{role}) en las redes quede fijado a 80. Con un umbral base elevado y un umbral de rol bajo, el número de VCs existentes será limitado, lo que forzará a los VCs existentes a admitir más nodos en cada una de las sub-redes. El tiempo que los nodos esperan para recibir respuestas de potenciales VCs (T_{link}) es también reducido, limitando el número de respuestas que pueden recibir y permitiendo de esta forma que los nodos se conecten a VCs en su cercanía. Por otro lado, el tiempo que un nodo espera una vez detectado un problema de conexión (T_{down}), quizás resulte elevado, pero en el estudio realizado en el Capítulo 5 se muestra que este resultado es el óptimo tanto a nivel de tiempo de convergencia como a nivel de consumo energético. El tiempo de reconexión ($T_{reconnect}$) se fija a 2s lo que permite una rápida reacción en el momento en que los nodos detectan que no tienen conectividad. El tiempo de respuesta (T_{ack}) se ha mantenido elevado de forma que se minimicen los reenvíos y se genere el menor tráfico posible en la red, como consecuencia de posibles colisiones entre paquetes. Finalmente el número de reintentos se ha mantenido en 3 tal y como lo fijan la mayoría de protocolos de comunicaciones existentes, dado que proporciona un buen compromiso en términos de fiabilidad a la hora de enviar un mensaje.

A través de la búsqueda y la utilización de esta configuración se ha alcanzado un rendimiento óptimo de *DARP* en grandes escenarios, lo que ha permitido desarrollar el estudio comparativo del Capítulo 6 que se discute a continuación.

8.2 DISCUSIÓN SOBRE EL PROTOCOLO DE COMUNICACIONES Y LOS ALGORITMOS PROPUESTOS

Una de las principales aportaciones científicas de esta Tesis Doctoral es el desarrollo de un nuevo protocolo de comunicaciones para WSN basado en el estándar *IEEE 802.15.4*. Los detalles acerca del desarrollo de los algoritmos se han expuesto en el Capítulo 4 y sus principales características analizadas previamente, se muestran en la Tabla 8.1. A continuación se analizan los puntos innovadores que presentan estos nuevos algoritmos.

- a. Conociendo los problemas para establecer rutas en grandes escenarios de media o baja densidad de los protocolos actuales, se propone un nuevo protocolo basado en algoritmos de organización jerárquica dinámica a partir de sub-redes. Esto permite optimizar la forma en que se crean las rutas, además de permitir la contención el tráfico en las sub-redes.
- b. Tomando como base los algoritmos de enrutamiento basados en *clustering*, se presenta una nueva propuesta, llamada DARP. Esta propuesta se basa en el concepto de organización jerárquica de las sub-redes lo cual permite optimizar el dinamismo en la fase de creación de la red.
- c. La utilización de una cabeza de clúster pre-definida no es eficiente y no permite la creación dinámica de las rutas. Sin embargo, la utilización de un algoritmo de selección dinámica de la cabeza del clúster presenta mejoras con respecto a la maximización de la cobertura de la red. Estos resultados corroboran la naturaleza dinámica presente en la fase de creación de las WSN.
- d. Actualmente no existe un medio que permita evaluar el dinamismo de un protocolo para WSN, ni a nivel de simulación ni al de implementación, por ello se propone la utilización del tiempo de convergencia y el número de mensajes de control enviados como parámetros de referencia para la creación de un marco de evaluación, tanto para simulaciones como para implementaciones.

Tabla 8.1
Listado de características principales ofrecidas por DARP.

8.2.1 Mejora en la creación de rutas en grandes escenarios

La justificación del punto A de la Tabla 8.1 queda satisfactoriamente resuelta tras los resultados obtenidos en el Capítulo 6 y presentados visualmente mediante las Figuras 6.1–6.27. Por lo tanto, es posible afirmar que en grandes escenarios tanto de baja como de media densidad, la propuesta de DARP mejora en términos de tiempo de convergencia y número de mensajes de control de la formación de las rutas que permiten crear la red a los resultados obtenidos por O-RPL y AODV.

Atendiendo a los resultados obtenidos del estudio de los grandes escenarios y bajo diferentes condiciones de densidad, DARAL presenta una importante mejora del dinamismo para grandes escenarios con densidades media o bajas. Las pruebas se han categorizado en función del número de nodos en el escenario: pequeño, medio y grande.

Como se puede apreciar en las Figuras 6.1–6.9, en todos los escenarios analizados con densidad media, DARAL supera, no solo a AODV, sino también a O-RPL. Cabe recordar que la versión de RPL utilizada en este trabajo, es una versión específicamente optimizada de cara a minimizar el tiempo de convergencia. La mejora del tiempo de convergencia con respecto a AODV en escenarios pequeños con densidad baja alcanza un 59.9 %, en escenarios medianos con densidad baja llega al 83.9 % y en escenarios grandes con esa misma densidad alcanza el 85.9 %. Con respecto a O-RPL la mejora alcanza cotas aún mayores, en el escenario pequeño con densidad baja la mejora de DARAL llega al 94.6 %, en escenarios medianos al 94.8 % y en escenarios grandes al 97.01 %. El resultado respecto del número de mensajes de control enviados en la fase de formación también resulta favorable a DARAL. En comparación con AODV el escenario pequeño de baja densidad muestra una mejora del 87.1 % a favor de DARAL. En los escenarios mediano y grande DARAL sigue mejorando a AODV en hasta un 89.9 % y un 93.5 %, respectivamente. Con respecto a O-RPL la mejora de rendimiento en condiciones de baja densidad para los escenarios pequeño, mediano y grande son de un 80.1 %, 91.7 % y un 97.9 %, respectivamente. Estas cifras se explican fácilmente debido a la naturaleza de los escenarios de baja densidad y los protocolos basados en grafos acíclicos. Debido a que los nodos se encuentran muy dispersos y como consecuencia de esta dispersión, resulta necesario que las comunicaciones pasen por determinados nodos situados en posiciones críticas. En el caso de un grafo acíclico, este evitará pasar por un mismo nodo varias veces, por lo que esos nodos en posiciones críticas solo podrán ser recorridos en una única ocasión. Este comportamiento conlleva la necesidad de generar diversos grafos para una única red, con el consiguiente gasto de recursos extra asociado y como se puede apreciar, con un incremento en los tiempos de convergencia y el número de mensajes de control enviados.

Repetiendo el análisis para los diferentes escenarios bajo condiciones de densidad media, DARAL vuelve a mejorar a AODV y O-RPL. Para escenarios pequeños la mejora del tiempo de convergencia alcanza el

68.02 % con respecto a *AODV* y el 74.8 % con respecto a *O-RPL*. En escenarios medianos la mejora continua, alcanzando el 79.03 % y el 57.3 %, respectivamente. Para los escenarios grandes esta mejora del tiempo de convergencia alcanza el 87.7 % y el 6.94 %. Estudiando también los resultados en términos de número de mensajes de control enviados durante la fase de formación para los distintos escenarios bajo densidad media obtenemos que *DARAL*, como en el caso del tiempo de convergencia, mejora el rendimiento tanto de *AODV* como de *O-RPL*. En los escenarios pequeños el porcentaje de mejora con respecto a *AODV* alcanza el 87.35 % y con respecto a *O-RPL*, el 48.1 %. En el escenario mediano los porcentajes son de un 91.07 % y un 81.61 %, mientras que en el escenario grande llegan a un 97.21 % y un 97.26 % con respecto a *AODV* y *O-RPL*. Nuevamente se puede apreciar como bajo condiciones de densidad media se repiten los resultados y se muestra como *DARAL* mejora a ambos protocolos en términos de tiempo de convergencia y número de mensajes de control.

Uno de los aspectos más interesantes de este análisis es que *O-RPL* presenta una tendencia en la que a mayor concentración de nodos, ya sea por un incremento en el número total de nodos, como por un incremento en la densidad de los mismos, sus resultados mejoran considerablemente. Como conclusión, cabe destacar que los resultados obtenidos ponen de manifiesto la idoneidad de utilizar un protocolos de enrutamiento específicamente optimizado en términos de dinamismo para WSN.

8.3 DISCUSIÓN SOBRE LA EVALUACIÓN DE CARACTERÍSTICAS ADICIONALES DE DARP

Los diferentes análisis realizados han permitido extraer interesantes conclusiones acerca de las características adicionales que aporta *DARP*, como por ejemplo la escalabilidad, el consumo de energía durante la fase de conexión, el aprovechamiento del canal, la reducción de las colisiones o la respuesta ante eventos dinámicos.

Como se puede apreciar en los test de escalabilidad realizados, ante un incremento en el número de nodos, aún con una densidad media, *O-RPL* mejora en términos de *tiempo de convergencia* a las demás alternativas, aunque *DARP* resulta mejor en términos de *número de mensajes de control enviados durante la fase de conexión*. A nivel de *tiempo de convergencia*, *DARP* se muestra más estable, obteniendo resultados similares a los que obtiene en el escenario grande analizado en el Capítulo 6. *AODV* presenta con un amplia diferencia los peores resultados. A nivel general, estos resultados se producen debido al amplio número de nodos existentes, lo que provoca que incluso ante una densidad media, los grafos no redundantes que genera *RPL* cuenten con suficientes posibilidades de enlace como para construir las rutas de formas eficiente, aunque a costa de un mayor consumo. En este sentido *DARP*, muestra un mejor desempeño, debido a que aunque obtenga peores resultados

para el *tiempo de convergencia*, mejora ampliamente el *número de mensajes de control enviados durante la fase de conexión*, lo que está estrechamente relacionado con el consumo de energía.

Es necesario puntualizar que debido a las características de los modelos de simulación estudiados, *RPL* requiere de unas necesidades en términos de memoria y capacidad de cómputo que hacen inviable la simulación del escenario enorme. Por otro lado y como en el escenario muy grande, el resultado de esta simulación permite observar una cierta mejoría en los resultados de *AODV* debido también a la gran cantidad de posibilidades de salto con las que cuenta, pero aún así, *DARP* mejora sus resultados, debido a la eficiencia del clustering y a la gran cantidad de posibilidades de enlace disponibles. Por este motivo, el rendimiento entre el escenario muy grande y el enorme presenta una mínima reducción.

En ambos escenarios *DARP* se muestra como la opción más estable y eficiente en términos de *número de mensajes de control enviados durante la fase de conexión*, dado que al organizarse en clústeres, los nodos necesitan muy pocos mensajes para certificar la conexión a una sub-red. Resulta claro que con un menor número de mensajes de control, el consumo energético resultará menor, lo cual queda confirmado en el análisis de ambos escenarios, donde *DARP* se encuentra un orden de magnitud por debajo de *AODV*.

Si el análisis de consumo se acota a la fase de conexión en escenarios grandes, se puede apreciar como *DARP* mejora tanto a *O-RPL* como a *AODV*, tal y como muestran las Figuras 7.10 y 7.11. La mejora de consumo de *DARP* vuelve a situarse en dos órdenes de magnitud, lo que es debido a que una vez desarrollada una rápida conexión de forma distribuida, no es necesario enrutar mensajes de forma constante como en *AODV* o retransmitir mensajes como en *O-RPL*. También influye el hecho de que son los propios nodos los que inicien el proceso de conexión, mejorando así el consumo medio en la red, al distribuir la carga de tareas.

El análisis del número de colisiones está estrechamente relacionado con el aprovechamiento del canal y el consumo de energía, en ambos casos y basado en la simulación de escenarios grandes, las Figuras 7.12 y 7.14 muestran como *DARP* tiene una menor duración de los *backoffs* y un menor número de colisiones que *AODV*, lo que repercute directamente tanto en la minimización del consumo energético como en la mejora de la tasa de envíos.

Finalmente, en el estudio de cómo responden los protocolos, tanto a nivel de *tiempo de convergencia* como de *número de mensajes de control enviados durante la fase de conexión*, en escenarios donde los nodos se conectan y desconectan de forma dinámica, presenta nuevamente un mejor rendimiento medio de *DARP*. *AODV* mejora ligeramente con respecto a sus resultados originales del Capítulo 6, pero sigue siendo inferior a *DARP*. *DARP* presenta unos resultados muy estables, en los

que no se aprecia un empeoramiento medio, a pesar de que los nodos se conectan y desconectan de los VCs.

8.4 CONCLUSIONES

Los resultados experimentales obtenidos avalan los avances e innovaciones científicas presentados en esta Tesis Doctoral y que se han resumido en este capítulo que se ha dividido en tres grandes bloques para dar respuesta a los principales temas tratados en la presente Tesis Doctoral.

El bloque (8.1) dedicado a la discusión sobre la convergencia de los parámetros de *DARP*, presenta la configuración óptima encontrada tras la realización de un proceso de convergencia iterativo. En este bloque también se discute la influencia de cada uno de los valores adoptados en un escenario tipo como las grandes infraestructuras propuestas en el marco de actuación.

El segundo bloque (8.2), muestra las principales aportaciones científicas obtenidas gracias al desarrollo de un nuevo protocolo y dos algoritmos, uno de enrutamiento y otro de selección dinámica del rol y el enlace de comunicaciones.

El tercer y último bloque (8.3), se ha dedicado a la discusión de algunas de las características adicionales con que cuenta *DARP* y que fueron presentadas en el Capítulo 4. Gracias a estos aportes se demuestra que *DARP* puede ser utilizado como alternativa a los protocolos de enrutamiento más extendidos actualmente, mejorando algunas características como consumo energético, respuesta ante eventos dinámicos o incluso mostrando una buena escalabilidad a mayores escenarios.

9

CONCLUSIONES

ÍNDICE

- 9.1 Producción científica **148**
 - 9.2 Principales aportaciones científicas **149**
 - 9.3 Investigaciones futuras **150**
-

Este capítulo sirve como colofón final al trabajo desarrollado a lo largo de esta Tesis Doctoral. En él, se presentan las conclusiones y las principales aportaciones científicas, así como posibles ampliaciones futuras en esta línea de investigación.

El capítulo se estructura en tres secciones. En la primera de ellas, se enuncian las publicaciones científicas que se han generado a partir del trabajo realizado en esta Tesis.

En la segunda sección, se exponen las conclusiones alcanzadas, tanto en relación con los objetivos propuestos inicialmente, como aquellos aspectos que no se habían contemplado en un principio, pero que se han logrado a lo largo de este trabajo. En este apartado, también se hace un especial hincapié en indicar las aportaciones científicas de esta Tesis.

Por último se presentan las líneas de trabajo futuro surgidas de esta Tesis Doctoral.

9.1 PRODUCCIÓN CIENTÍFICA

La presente Tesis Doctoral ha generado una serie de publicaciones científicas, que se detallan a continuación:

- **F. Estevez, P. Gloesekoetter, J. González** DARAL: *A Dynamic and Adaptive Routing Algorithm for Wireless Sensor Networks*. Enviado a *Sensors*, 2016. Índice de Impacto: 2.24.
- **F. Estevez, G. Rebel, J. González, P. Gloesekoetter** DARP: *Dynamic and adaptive radio protocol for Wireless Sensor Networks*. Aceptado en *Electronic Letters*, 2014. Índice de Impacto: 0.93.
- **F. Estevez, G. Rebel, J.M. Castillo-Secilla, P. Gloesekoetter, J. González** Cognitive Radio as solution for Saturation Problems in Wireless Sensor Networks: A survey of Coginitive Radio policies. En las Actas del V Jornadas de Computación Empotrada: Conferencia JCE2014, Sociedad de Arquitectura y Tecnología de Computadores (SARTECO), Valladolid (España), 2014.

9.2 PRINCIPALES APORTACIONES CIENTÍFICAS

Los objetivos inicialmente propuestos en el Capítulo 2, se han satisfecho en su totalidad. A continuación, y siguiendo el mismo esquema utilizado en el capítulo de conclusiones, se expone la consecución de los mismos:

- Se ha desarrollado un protocolo y un conjunto de algoritmos, de enrutamiento y de selección del enlace, basados en la minimización del *tiempo de convergencia* y el *número de mensajes de control enviados*. Como consecuencia se ha creado el protocolo *DARP* y los algoritmos *DARAL* y *DRSP*.
- Se ha modelado el comportamiento del protocolo *DARP*, proponiendo una configuración óptima para los escenarios de grandes infraestructuras considerados en el presente trabajo.
- El protocolo propuesto mejora ampliamente los resultados en términos de dinamismo obtenidos con *AODV* y *O-RPL* en grandes escenarios similares a las grandes infraestructuras bajo condiciones de baja o media densidad.
- El algoritmo de enrutamiento *DARAL* no solo optimiza el *tiempo de convergencia* y el *número de mensajes de control enviados*, sino que integra otras características necesarias en WSN como son: la *escalabilidad de la red*, el *aprovechamiento del canal*, la *minimización del número de colisiones*, la *respuesta ante eventos dinámicos* y la *optimización del consumo de energía*.

9.3 INVESTIGACIONES FUTURAS

Esta Tesis Doctoral proporciona un conjunto de bases de carácter científico que permiten iniciar nuevas líneas de investigación. Algunas de ellas, se describen a continuación:

- Difundir aquellos resultados que aun no han sido publicados en congresos o revistas científicas.
- Realizar estudios energéticos que permitan conocer el gasto de *DARP* con exactitud y, permitan optimizarlo en consecuencia.
- Desplegar *DARP* en un escenario real y bajo condiciones reales, para realizar nuevas mediciones y análisis, con el fin de publicar los conocimientos científicos generados.
- Analizar la influencia del tráfico y la densidad, evaluando la posibilidad de integrar técnicas de radio cognitiva a *DARP*.
- Analizar y estudiar la integración de mecanismos de comunicación por malla o por grafos dentro de las sub-redes, desarrollando una topología híbrida.
- Implementar un sistema de calibración automático de los parámetros de *DARP* mediante el uso de *Sistemas Expertos* para ajustar los parámetros en función de las características del escenario de despliegue.

Parte VII

DISCUSSION AND CONCLUSIONS (ENGLISH VERSION)

DISCUSSION

CONTENTS

- 10.1 Discussion about the optimum configuration of DARAL for routing in large scenarios **154**
 - 10.2 Discussion on the routing protocol and the algorithms **155**
 - 10.2.1 Improvement in the route creation for large scenarios **155**
 - 10.3 Discussing the additional features of DARP **157**
 - 10.4 Conclusions **158**
-

The present chapter presents the analysis and discussion of the results generated in this Thesis. The main contributions are the routing protocol and the two algorithms used by the protocol. In order to ease the readability, the chapter is organized as follows:

First of all, the optimum configuration for the protocol is discussed, analyzing the influence of each configuration parameter in the performance for a large-size scenario.

Secondly, the aspects related to large-size scenarios are discussed, focusing on the use of the link quality, the *convergence time* and the *number of messages sent during the forming phase*. The main contributions of this work, *DRSP*, *DARAL*, and *DARP* are based on the concept of dynamism, widely used among this Thesis.

Thirdly, some additional features developed within the routing protocol are discussed, analyzing their performance and comparing them with other alternatives in some well-known protocols.

10.1 DISCUSSION ABOUT THE OPTIMUM CONFIGURATION OF DARAL FOR ROUTING IN LARGE SCENARIOS

The main contribution of this Thesis is the routing protocol *DARP*, which is based on a specific routing algorithm *DARAL*, which is also one of the contributions of this work. All of them were described in Chapter 4.

DARAL presents different configuration parameters, which offer a wide range of configurations, making the routing protocol highly adaptable to different scenarios. As the objective scenarios of this Thesis are the large infrastructures, which usually have medium or low densities, as it was discussed in Chapter 2, it results necessary to properly adjust the protocol configuration to these scenarios.

An iterative analysis is carried out in Chapter 5, which allows us to find a convergent configuration. The convergence is reached after three iterations, thus it is presented in Table 5.4, which also presents the optimum configuration found.

These parameters applied to large infrastructures present some interesting characteristics, for example, the necessity of setting a high base-level threshold ($TH_{baselevel}$). This configuration allows us to discard more low-quality links, assuring that the nodes interconnected have a higher probability of success sending messages between them. The role threshold (TH_{role}) is set at 80, which means a low number of VCs. This configuration forces the VCs to accept more ENs in the sub-networks, resulting in larger sub-networks. The nodes wait a certain time for answers from the VCs in order to choose one of them and try to connect to the network. That time is called T_{link} and it is set at 1s, which limits the number of answers that a node can receive, indirectly limiting the possible connections to the nodes in its neighborhood. Another parameter is the waiting time as soon as a node detects a problem with the connection (T_{down}). The optimum configuration is set to 45s, which seems to be the best configuration in terms of convergence time and energy consumption. The reconnection time ($T_{reconnect}$) is set at 2s, which means a fast response when the nodes detect no connectivity. The acknowledgment time (T_{ack}) is set a bit high, but that means a lower number of resend and lower traffic in the network, minimizing also the collisions. Finally the number of resends is set to 3, in the same way that other routing protocols and standards, as it gives a good compromise between reliability and use of resources.

Using this configuration, *DARP* shows an optimum performance for large scenarios, which allows the comparison with other protocols presented in Chapter 6 and discussed in the next section.

10.2 DISCUSSION ON THE ROUTING PROTOCOL AND THE ALGORITHMS

One of the main scientific contributions of this Thesis is the development of a new *IEEE 802.15.4*-based routing protocol for Wireless Sensor Networks. The details about their algorithms are presented in Chapter 4 and the main innovative features of these algorithms are presented in Table 10.1.

Tabla 10.1
Main DARP features list.

- a. The problems to establish new routes in large scenarios under medium or low density conditions are well known, therefore, a new routing protocol based on algorithms that dynamically build a hierarchical sub-network-based organization is developed. This organization not only optimizes the route creation process, but also the traffic containment in the sub-networks.
- b. Based on clustering-based routing algorithms, *DARAL* is presented. This routing algorithm is based on the sub-network hierarchical organization concept, which increases dynamism during the network forming phase.
- c. The use of a pre-defined cluster-head is neither efficient enough, nor allows the dynamic route creation. However, the use of a dynamical head-cluster selection algorithm presents improvements such as maximizing the network coverage. These results corroborate the dynamism during the network forming phase.
- d. Nowadays, there is not any way to evaluate the dynamism of a WSN protocol, neither at simulation level nor at implementation level. Therefore, a proposal based on the use of the convergence time and the number of messages sent during the forming phase is presented, which uses these parameters as reference for the creation of an evaluation framework.

10.2.1 Improvement in the route creation for large scenarios

The justification to point A in Table 10.1 is satisfied through the results of Chapter 6 and visually presented by Figures 6.1–6.27. Therefore, the proposal made in this Thesis, *DARP*, improves the results obtained with *O-RPL* and *AODV* in terms of convergence time and number of messages sent during the forming phase, for large scenarios under medium and low density conditions.

Based on the results of large scenarios under different density conditions, *DARAL* presents a wide dynamism improvement for large scenarios.

rios under medium and low densities. The tests have been categorized by number of nodes, classifying them as: small, medium, and large.

As it can be seen in Figures 6.1–6.9, all the medium-density scenarios analyzed show a better performance of *DARAL*, improving *AODV* and *RPL*. It is necessary to mention that the *RPL* version used in this Thesis is an optimized version of *RPL* called *O-RPL*, focused on the convergence time optimization. The convergence time improvement of *DARAL* respect to *AODV* in small scenarios under low-density conditions reaches 59.9 %, in medium scenarios under low-density conditions, it reaches 83.9 %, and in large scenarios with the same density conditions, it reaches 85.9 %. Regarding *O-RPL*, the improvement of *DARAL* in small scenarios under low-density conditions reaches 94.6 %, in medium scenarios, 94.8 %, and in large scenarios, 97.01 %. The improvement in terms of number of messages sent during the forming phase also results favorable to *DARAL*, presenting an improvement of 87.1 % for small scenarios under low-density conditions. For large and medium scenarios, the improvement reaches in both cases 89.9 % and 93.5 %, respectively. Analyzing *O-RPL*, the improvement under low-density conditions in small, medium, and large scenarios reaches 80.1 %, 91.7 %, and 97.9 % respectively. These results are easily explained due to the nature of low-density scenarios and the conflict with acyclic graph-based protocols. Due to the wide dispersion of the nodes, it results necessary that the communications flow through some nodes located in critical positions. Acyclic graphs avoid the use of the same node several times, although those nodes are in critical positions, they can only be used once. This behavior leads to the creation of several acyclic graphs for a unique network, resulting in a higher resource consumption and an increment in the convergence time and the number of messages sent.

Repeating the analysis under medium-density conditions, *DARAL* also improves *AODV* and *O-RPL*. In small scenarios, the convergence time improvement respect to *AODV* reaches 68.02 %, and respect to *O-RPL*, it reaches 74.8 %. In medium-size scenarios, the improvement of *DARAL* reaches 79.03 % and 57.3 %, in large scenarios respectively, it reaches 87.7 % and 6.94 % for *AODV* and *O-RPL*, respectively. The analysis of the results in terms of number of messages sent during the forming phase for different-size scenarios under medium-density conditions also shows, as in the convergence time case, a better performance of *DARAL* respect to *AODV* and *O-RPL*. In small scenarios, the improvement respect to *AODV* reaches 87.35 %, and as compared to *O-RPL*, 48.1 %. In the medium-size scenario, the improvement reaches 91.07 % and 81.61 % comparing *DARAL* with *AODV* and *O-RPL*. In the large-size scenario, it reaches 97.21 % and 97.26 % as compared to *AODV* and *O-RPL*. Under medium-density conditions, the results are very similar and *DARAL* improves both protocols, *AODV* and *O-RPL*, in terms of convergence time and number of messages sent during the forming phase.

Finally, an interesting aspect about this analysis is that *O-RPL* presents a trend where as node concentration is increased, either by an incre-

ment in the number of nodes or by an increment in the density, its results improve accordingly. As a conclusion, the presented results show the suitability of using dynamism-optimized routing protocols for large scenarios under medium or low density conditions in WSN.

10.3 DISCUSSING THE ADDITIONAL FEATURES OF DARP

The different analyses carried out have shown interesting results about the additional features developed within *DARP*, such as scalability, energy consumption reduction, channel exploitation, collisions reductions, or response against dynamic events.

As can be seen in the scalability test developed, for a higher number of nodes, even if the density remains at medium-level, *O-RPL* improves, in terms of convergence time, the other alternatives, although *DARP* shows a better performance in terms of number of messages sent during the forming phase. Focusing on the convergence time analysis, *DARP* presents more stability in the results, obtaining similar results with the large-size scenario analyzed in Chapter 6. On the other hand, *AODV* presents the worst results.

These results are the consequence of the high number of nodes present in the test scenarios, which results, even under medium-density conditions, in a high number of possibilities to efficiently create routes. Topologies like acyclic graphs are very efficient in this kind of scenarios. However, *DARP* shows a better performance in global terms, even though it presents the worst performance in terms of convergence time, due to the wide improvement in terms of number of messages sent during the forming phase, which is closely related to the reduction of energy consumption.

In the enormous-size scenarios, it is necessary to point out that *O-RPL* requires a very high amount of memory and computational power, making its simulation non-viable. On the other hand, these simulation results show a certain improvement of *AODV*, due to the high number of potentially available jump options. But even considering this improvement, *DARP* improves the results of *AODV*, due to the efficiency shown by clustering, which derives in the fact that the performance of the enormous-size scenario and the very large-size scenario present a minimal difference.

In both scenarios *DARP* is the most efficient option in terms of number of messages sent during the forming phase, due to the application of clustering, where the nodes need only a few messages to be connected to the network. A low control messages number requires a lower energy consumption, which is confirmed by the analysis of both scenarios, where *DARP* is a magnitude order lower than *AODV*.

If the energy consumption analysis is limited to the forming phase in large-size scenarios, it is possible to observe the improvement of *DARP* regarding *O-RPL* and *AODV*, just as Figures 7.10 and 7.11 show. The energy consumption improvement is two magnitude orders lower, due to the fast route-forming process, which requires fewer control messages and fewer routing tasks. Another difference is that in *DARP*, the non-connected nodes are the ones that start the connection process, improving the average energy consumption and distributing the work load among the network.

The analysis of the number of collisions is closely related to the channel exploitation and the energy consumption, in both cases and based on large-size scenarios, Figures 7.12 and 7.14 show a lower *backoffs* duration and a lower number of collisions for *DARP*. Both improvements have an effect on the reduction of the energy consumption and the improvement of the packet delivery rate.

Finally, the study of the protocol response in dynamic scenarios, where the nodes randomly connect and disconnect, shows a better performance of *DARP*, in terms of convergence time and number of messages sent during the forming phase. *AODV* improves slightly respect to its previous results from Chapter 6, but it is anyway worse than *DARP*, which presents very stable results, where there is no downturn, even though the nodes connect and disconnect dynamically.

10.4 CONCLUSIONS

The experimental results obtained support the scientific improvements and innovations presented in the present Thesis, which are summarized in this chapter. The chapter is organized in three major blocks, which treat the major themes studied in this Thesis.

Section (10.1) discusses about the convergence of the *DARP* configuration parameters, presenting the optimum configuration found after an iterative study. This block also discusses about the influence of each value selected in a large infrastructure type scenario.

In Section (10.2), the main scientific contributions are presented. These contributions are a new routing protocol and two algorithms, one with routing purposes and another one for a dynamic role and route selection.

The third and last section, Section (10.3), has been dedicated to the discussion of some additional features from *DARP*, which have been presented in Chapter 4. These features evidence that *DARP* can be used as an alternative to some well-known routing protocols, improving some characteristics such as energy consumption, response against dynamic events, channel exploitation, number of collisions, and scalability.

CONCLUSIONS

CONTENTS

- 11.1 Scientific Production **160**
 - 11.2 Main Scientific Contributions **161**
 - 11.3 Future Research **162**
-

This chapter serves as the culmination to the work developed in this doctoral thesis presenting the conclusions and the main scientific contributions, as long as future research lines.

The chapter is organized in three sections. In the first one, the publication list generated from this thesis is presented.

The second section continues the conclusions, both those related with the first objectives defined and those other objectives not firstly defined but proposed throughout this work. This section also emphasizes the main scientific contributions achieved in the development of this thesis.

Finally, the third section presents the future research lines arising from this doctoral thesis.

11.1 SCIENTIFIC PRODUCTION

The present doctoral thesis has generated some scientific publications, which are detailed below:

- F. Estevez, P. Gloesekoetter, J. González *DARAL: A Dynamic and Adaptive Routing Algorithm for Wireless Sensor Networks*. Sent to *Sensors*, 2016. Impact Factor: 2.24.
- F. Estevez, G. Rebel, J. González, P. Gloesekoetter *DARP: Dynamic and adaptive radio protocol for Wireless Sensor Networks*. Published in *Electronic Letters*, 2014. Impact factor: 0.93. DOI: [10.1049/el.2013.2488](https://doi.org/10.1049/el.2013.2488).
- F. Estevez, G. Rebel, J.M. Castillo-Secilla, P. Gloesekoetter, J. González *Cognitive Radio as solution for Saturation Problems in Wireless Sensor Networks: A survey of Cognitive Radio policies*. In the Proceedings of V Jornadas de Computación Empotrada: JCE2014 Conference, Sociedad de Arquitectura y Tecnología de Computadores (SARTECO), Valladolid (Spain), 2014.

11.2 MAIN SCIENTIFIC CONTRIBUTIONS

The first objectives proposed in Chapter 2 have been totally satisfied and below, their attainment is presented:

- A routing protocol and a set of algorithms have been developed, one algorithm for routing tasks and another, for link selection tasks. Both are based on the minimization of the *convergence time* and the *control messages sent during the set-up phase*. As a consequence, the *DARP* protocol and the algorithms *DARAL* and *DRSP* have been developed.
- The *DARP* protocol has been modeled to analyze its performance, proposing an optimal configuration of its parameters for large scenarios, as this is one of the main objectives in the present work.
- The proposed protocol improves the results of *AODV* and *O-RPL* in large scenarios under low and medium density conditions.
- The proposed algorithm *DARAL*, not only optimizes the *convergence time* and the *number of messages sent during the set-up phase*, but also integrates other necessary features for WSN such as: *network scalability, increasing the channel exploitation, reducing the collision rate, response to dynamic events, and energy consumption optimization*.

11.3 FUTURE RESEARCH

This doctoral thesis provides a basis that allows the initiation of new research lines. Some of these possibilities are described below:

- Spreading the results that are not yet published, either in conferences or journals.
- Developing detailed energy consumption studies, to determine the energy use with high-precision in order to be able to minimize it.
- Deploying *DARP* in a real scenario and testing it under real conditions.
- Analyzing the influence of traffic and density on *DARP*, evaluating the possibility of integrating new cognitive radio methods.
- Analyzing and studying the integration of communication mechanisms from other topologies or routing protocols, developing a hybrid-topology routing method.
- Implementing an automatic parameter calibration system for *DARP*, using *expert systems* to adjust the different configuration parameters depending on the peculiarities of the deployment scenario.

Parte VIII

APÉNDICES

A

PLATAFORMAS ACTUALES EN WSN

ÍNDICE

A.1	Plataformas Hardware para WSN	166
A.1.1	MICAz	166
A.1.2	TelosB/Tmote Sky	166
A.1.3	WaspMote	167
A.1.4	Arduino/Genuino	167
A.1.5	RedBee-Econotag	168
A.2	Plataformas Software para WSN	168
A.2.1	TinyOS	168
A.2.2	Contiki	169
A.2.3	Nano-RK	170
A.2.4	FreeRTOS	172
A.3	Plataformas de Simulación para WSN	172
A.3.1	NS-2/NS-3	173
A.3.2	OMNeT++/OMNEST	173
A.3.3	TOSSIM	174
A.3.4	Cooja	175

Los avances en los sistemas microelectrónicos con sensores han permitido el desarrollo de nodos de bajo coste y tamaño reducido. Estos nodos cuentan con la capacidad de comunicarse de forma inalámbrica, obtener datos del entorno por medio de sus sensores o llevar a cabo el procesamiento de la información. Actualmente existen numerosos dispositivos que pueden actuar como nodos, aunque el hardware por sí solo resulta en un elemento inutilizable. Por ello, resulta también necesario el uso de Software. En este caso, puede utilizarse una aplicación específica o un Sistema Operativo. Esto depende de la aplicación a la que el nodo vaya a dedicarse, aunque por su propósito multi-aplicación y por las ventajas que ofrece, en muchas ocasiones es no solo más utilizado, sino más recomendable, el uso de un Sistema Operativo.

A lo largo de este apéndice se presentan las plataformas hardware, los Sistemas Operativos Empotrados y los simuladores más utilizados.

A.1 PLATAFORMAS HARDWARE PARA WSN

A continuación, se detallan las plataformas hardware más utilizadas en el área de las WSN.

A.1.1 *MICAZ*

MICAZ es un dispositivo fabricado por MEMSIC y diseñado por la Universidad de Berkeley, con el objetivo de proporcionar a la comunidad investigadora una plataforma de desarrollo de WSNs de bajo coste. MICAZ opera en las bandas de frecuencia 2.4GHz, 868 y 916MHz, y ofrece soporte para redes *IEEE 802.15.4*.

MICAZ se basa en un microcontrolador ATmega128L, una antena integrada y una radio compatible con *IEEE 802.15.4* a 2.4GHz. MICAZ es conocida por su facilidad de programación y su compatibilidad con TinyOS 1.1.11 o superior y Contiki, además de contar con un bajo consumo energético y soporte para la recolección de datos vía USB. Esta plataforma cuenta también con diferentes sensores integrados, entre ellos: sensor de iluminación, de temperatura y de humedad.

A.1.2 *TelosB/Tmote Sky*

A partir de la aparición de MICAZ, la propia Universidad de Berkeley desarrolló una nueva plataforma aún más competitiva en términos de costo y consumo energético. Estas nuevas plataformas son las conocidas como *TelosB* o *Tmote Sky*. Actualmente, la plataforma *TelosB/Tmote Sky* es una de las más utilizadas en trabajos de investigación, dada su flexibilidad y compatibilidad con diferentes Sistemas Operativos.

TelosB/Tmote Sky se basa en un microcontrolador MSP430 [18] de Texas Instruments con 10KB RAM, una radio cc2420X [1], también de Texas Instruments compatible con el estándar *IEEE 802.15.4* y una antena integrada. *TelosB/Tmote Sky* posee además, la capacidad de añadir dispositivos adicionales, como sensores analógicos o pantallas LCD, a través de dos conectores de expansión. Esta plataforma es compatible con TinyOS 1.1.11 o superior y Contiki. Sus principales ventajas al igual que con las MICAZ, son el bajo consumo energético y la posibilidad de recolección de información vía USB.

A.1.3 *WaspMote*

La plataforma WaspMote es un diseño de Libelium, esta plataforma opera en las bandas de frecuencia 2.4GHz, 900MHz, 868MHz definidas por el estándar *IEEE 802.15.4*. Waspmote se caracteriza por una alta potencia de transmisión, así como una gran sensibilidad de recepción, lo que les permite alcanzar grandes distancias. También resulta de interés destacar que Waspmote posee un acelerómetro de 3 ejes integrado.

Waspmote está diseñado con una arquitectura modular que permite la adición de cualquier tipo de dispositivo externo, como lectores de tarjetas, módulos GSM o GPS, pantallas, sensores de gases, de luminosidad, de peso, de vibración, etc.

A.1.4 *Arduino/Genuino*

Arduino/Genuino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos. Aun no siendo un dispositivo enfocado al ámbito investigador, resulta relevante debido a lo extendido de su uso.

En el ámbito de las WSN, Arduino/Genuino posee compatibilidad para ser usado con radios compatibles con el estándar *IEEE 802.15.4*, bajo las frecuencias 2.4GHz, 900MHz, 868MHz. Existen un gran número de modelos basados en la plataforma Arduino. La mayoría están basados en la familia de microcontroladores ATmega de Atmel, aunque recientemente han aparecido Arduinos basados en arquitecturas de 32-bits, del tipo Cortex-M3, o con base en sistemas x86 como resultado de la reciente colaboración con Intel.

El Arduino UNO se basa en microcontrolador ATmega328 [3], y debido a su diseño modular, permite la utilización de shields. Estos shields son básicamente placas adaptadoras que amplían las capacidades de Arduino. Para obtener conectividad inalámbrica se hace uso de, por ejemplo, el XBee shield, que va colocada directamente sobre el Arduino UNO. El uso del XBee shield permite, junto con su correspondiente antena XBee, desplegar la red. Arduino no es compatible con ningún Sistema Operativo al uso, ya que se basa en una implementación de firmata que permite ejecutar un lenguaje específico llamado *Processing* y que resulta compatible con la programación de código fuente en C estándar.

A.1.5 RedBee-Econotag

La plataforma RedBee-Econotag es un diseño de Mariano Alvira [16], esta plataforma opera en la banda de frecuencia 2.4GHz definida en el estándar IEEE 802.15.4. Redbee-Econotag se caracteriza por funcionar con un SoC MC1322x de Freescale. Este SoC está compuesto de un núcleo ARM7 a 26MHz, una radio integrada y un módulo hardware para tareas de encriptación AES.

RedBee-Econotag es una plataforma compatible con Contiki y provee de una mayor capacidad de cómputo, debido al uso de una arquitectura de 32-bits y al uso de memorias de más tamaño. Esta plataforma también viene preparada para su programación vía USB a través de un programador integrado.

A.2 PLATAFORMAS SOFTWARE PARA WSN

Una Plataforma Software no es ni más ni menos que un Sistema Operativo Empotrado, el cual actúa como administrador de los recursos de la plataforma hardware. Estos recursos incluyen procesadores, memorias, temporizadores, discos, periféricos, interfaces de red, etc. El trabajo del Sistema Operativo es el de administrar los recursos de cara al usuario de manera controlada y ordenada. Los Sistemas Operativos ponen a disposición de los programadores una serie de llamadas al sistema con las que pueden interactuar de manera directa con los recursos hardware, conocidas como CAPA DE APLICACIÓN (APIs). Los recursos limitados del paradigma de las WSN obligan a desarrollar programas que hagan un uso responsable y limitado de los recursos.

A.2.1 TinyOS

TinyOs [57] es un Sistema Operativo monolítico de código abierto, flexible y basado en componentes. Está desarrollado para su uso en WSN. Entre los beneficios de TinyOS se encuentra la posibilidad de ejecutar programas de manera concurrente con una baja utilización de memoria. Prueba de ello es el tamaño de su imagen, la cual ocupa apenas 400 Bytes en memoria. La librería de componentes de TinyOS es amplia, incluyendo protocolos de red, servicios distribuidos, drivers para sensores y herramientas de adquisición de datos.

TinyOS usa un modelo de componentes los cuales, de acuerdo a los requerimientos de la aplicación, son conectados entre sí junto con el planificador para componer una imagen que arranca. TinyOS 2.1 proporciona soporte multi-hilo mediante los conocidos como *TOS Threads*.

La utilización de hilos, permite llevar a cabo una programación más intuitiva de cara a los desarrolladores.

Los hilos tienen la posibilidad de apropiarse de los recursos utilizados por otro hilo siempre que se estime necesario, pero no pueden apropiarse de recursos utilizados por tareas o interrupciones. Para gestionar todo esto, se tiene un hilo de alta prioridad ejecutando el planificador del Sistema Operativo. Las comunicaciones entre los hilos y el kernel de TinyOS son llevadas a cabo mediante una política de paso de mensajes. TinyOS también proporciona soporte a la sincronización con la ayuda de variables de condición y semáforos mutex. Tras analizar el modo de gestión de la memoria en TinyOS se comprueba que sigue un esquema de memoria estática.

TinyOS ofrece una implementación de la capa MAC del estándar *IEEE 802.15.4*, lo que permite el uso de muchos protocolos y algoritmos como TDMA de salto simple o TDMA/CSMA híbrido.

A.2.2 *Contiki*

Contiki [37] es un Sistema Operativo modular de código abierto. Está escrito en C y orientado a las WSN, prueba de ello es su reducido tamaño en memoria (apenas alcanza los 2 KBytes de RAM y 40 KBytes de ROM). Es un Sistema Operativo altamente portable y su kernel funciona a base de eventos. Contiki es multitarea y, por tanto, basa su funcionamiento en el concepto de apropiación.

La instalación completa de Contiki, ofrece al usuario numerosos servicios: kernel multitarea, funcionamiento multi-hilo con apropiación, proto-hilos, pila TCP/IP, soporte IPv6, interfaz gráfica de usuario, navegador web, cliente telnet o incluso cifrado de seguridad por software.

A nivel de kernel, su funcionamiento está basado en eventos junto con características multi-hilo para facilitar la ejecución de procesos individuales. El kernel de Contiki dispone de un planificador de reducido tamaño, el cual genera eventos para comenzar la ejecución de los diferentes procesos en espera. La ejecución de los procesos es llevada a cabo mediante un mecanismo de tipo polling. Mediante este mecanismo se evitan los conflictos generados por los procesos al hacer uso de los recursos disponibles, asegurando el correcto acceso al sistema de cada uno de ellos.

Contiki soporta dos tipos de eventos: *Eventos asíncronos* y *eventos síncronos*. Contiki proporciona en forma de servicios la administración de datos de sensores, las comunicaciones y los drivers de los dispositivos. Cada servicio tiene su propia interfaz e implementación. Cualquier aplicación que quiera hacer uso de un determinado servicio deberá conocer la interfaz que lo proporciona. De este modo, la aplicación

es abstraída de la implementación del servicio de manera similar a la programación orientada a objetos.

El multi-hilo es implementado mediante una librería ubicada por encima del kernel dirigido por eventos. Cualquier aplicación que necesite hacer uso del soporte multi-hilo no tendrá más que enlazar su código hacia la librería proporcionada para tal efecto. La librería multi-hilo de Contiki se divide en dos partes: una de ellas es independiente de la plataforma hardware y la otra, por el contrario, es dependiente de la plataforma.

Para dar soporte multi-hilo, Contiki define el concepto de protothreads. Los protothreads han sido específicamente diseñados para trabajar en condiciones muy restringidas de memoria. Las características principales de los protothreads son: utilización mínima de la memoria del sistema (2 bytes por protothread), no se utiliza un stack propio por cada protothread y son altamente portables gracias a su programación en lenguaje C. En contiki, no existe sincronización de procesos debido a que los eventos continúan su ejecución hasta finalizar y no se permite la utilización de manejadores de interrupciones para añadir nuevos eventos.

Contiki es un Sistema Operativo dirigido por eventos lo que hace innecesario el uso de planificador. Los eventos son disparados hacia las aplicaciones en estricto orden de llegada mientras que, en el caso de las interrupciones, los manejadores de interrupciones permiten que las aplicaciones se ejecuten según la prioridad que tengan asignada.

Al estar completamente orientado a las WSN, Contiki soporta un amplio abanico de protocolos de comunicación. Permite utilizar tanto IPv4 como IPv6. Teniendo en cuenta la máxima de cualquier WSN (es decir, la existencia de nodos de bajo coste con microcontroladores reducidos), Contiki implementa el protocolo TCP/IP para microcontroladores de 8 bits bajo el nombre de μ IP. La implementación de μ IP posee las características básicas para formar una pila TCP/IP, está escrito en lenguaje C y sólo permite la existencia de una interfaz de red por dispositivo. Gracias a la interfaz de red, puede comunicarse haciendo uso de los protocolos TCP/IP, ICMP y UDP. Contiki también proporciona otra pila de reducido tamaño conocida como Rime, la cual está orienta a comunicaciones basadas en redes. Rime puede enviar mensajes tanto directos como en difusión. Contiki también proporciona una implementación del estándar IEEE 802.15.4 que permite la utilización de diferentes algoritmos y protocolos para WSN.

A.2.3 Nano-RK

Nano-RK es un Sistema Operativo monolítico de tiempo real (**RTOS**) diseñado por la Universidad Carnegie Mellon. Hace uso de un modelo apropiativo y ofrece funciones de red para WSN. Actualmente, Nano-

RK puede ser utilizado en las motas de tipo FireFly y MicaZ. El kernel de Nano-RK destaca por su gran capacidad de administración de los temporizadores hardware así como por su reducido tamaño: tan sólo utiliza 2 Kbytes de RAM y 18 KBytes de ROM. Con el objetivo de cumplir el deadline de los procesos, Nano-RK es un sistema multitarea con posibilidad de fijar las prioridades de las tareas. Las tareas realizan peticiones para el uso de los recursos y, Nano-RK proporciona acceso a los mismos garantizando el acceso a la CPU así como al resto de recursos del sistema.

Dada la naturaleza de tiempo real de este Sistema Operativo, este ofrece un conjunto de herramientas que permiten el correcto uso del tiempo: la prioridad de tareas, deadlines, períodos y reservas deben ser asignados de manera offline. Nano-RK proporciona una API a través de la cual, los parámetros de las tareas pueden ser configurados en tiempo de ejecución. Esto ofrece la ventaja de poder modificar el sistema sin tener que interrumpir el funcionamiento,

La naturaleza apropiativa de Nano-RK, provoca la necesidad de almacenar el contexto de la tarea que está ejecutando antes de comenzar la ejecución de una nueva tarea. El almacenamiento de cada tarea provoca un alto consumo de memoria y cambios de contexto frecuentes, lo cual, se ve traducido en una disminución del rendimiento y un mayor consumo energético.

En Nano-RK, cada tarea tiene asociada un bloque de control o TCB. El TCB almacena los contenidos de los registros, prioridades, períodos de recurrencia, tamaños de reserva e identificadores de puertos de las tareas. Para proporcionar soporte completo de tiempo real, Nano-RK utiliza el concepto de multitarea apropiativa. Esto implica que la tarea de mayor prioridad siempre podrá acceder a ser ejecutada en CPU. Nano-RK proporciona diferentes algoritmos de planificación como el *Rate-Harmonized Scheduling for Saving Energy* o el *algoritmo de techo de prioridad*.

Nano-RK ofrece un protocolo de comunicación de tamaño reducido basado en sockets. La aplicación que desee llevar a cabo un proceso de comunicación deberá crear una estructura de tipo socket para comunicarse a través de la misma. De la misma manera, en modo de recepción, la aplicación deberá tener un socket a la espera de comunicaciones entrantes.

En Nano-RK se ha llevado a cabo la implementación del protocolo RT-LINK [72]. El objetivo principal de RT-Link es el de aumentar el tiempo de vida de la red así como dar garantías de mínimo retardo en las comunicaciones. RT-Link da soporte de tiempo real asegurando que la transmisión es libre de colisiones a lo largo de todos los saltos de la red. Está implementado sobre el protocolo de capa de enlace TDMA, donde la transmisión de cada nodo ocurre en una ventana temporal previamente definida.

RT-LINK: Time
Synchronized Link
Protocol for Energy
Constraint Multi-Hop
Wireless Networks.

A.2.4 FreeRTOS

FreeRTOS [8] es un Sistema Operativo de tiempo real flexible y modular, de código abierto y distribuido bajo una licencia GPL modificada. Este Sistema Operativo se diseña en lenguaje C basado en un microkernel que facilita su portabilidad a diferentes arquitecturas. Aún así provee de los métodos básicos que todo Sistema Operativo requiere: hilos con diferentes niveles de prioridad, semáforos, mutex, temporizadores y manejador de memoria.

FreeRTOS utiliza un planificador de tipo round-robin basado en prioridad y permite la ejecución multi-hilo a través de las librerías específicas para ello. En FreeRTOS el planificador puede ser configurado de forma que funcione en base a prioridad o que funcione de forma colaborativa. FreeRTOS al igual que otros de los sistemas comentados, ofrece un tamaño de imagen muy pequeño, por ejemplo, en una aplicación para el IoT específica con las librerías de Nabto alcanza un tamaño de 10Kb.

Una de las principales características de FreeRTOS es la capacidad de ajustar el reloj del sistema (Tick) en tiempo de ejecución, de forma que reduzca el consumo de energía de una determinada aplicación si resulta necesario.

Al ser un sistema muy básico, FreeRTOS hace uso de drivers y librerías de terceros que se integran perfectamente en el Sistema Operativo como: pilas de comunicaciones IPv4 o IPv6, soporte para almacenamiento de archivos tipo FAT, interfaces gráficas, etc. Existen además librerías de periféricos específicos de multitud de fabricantes, Atmel, NXP, ST, Future Designs Inc., etc. Estas librerías ofrecen soporte para interfaces como: USART, SPI, I2C, CAN; O para componentes como los temporizadores, los manejadores de interrupciones propios de cada arquitectura o las entradas/salidas.

FreeRTOS también ofrece implementaciones del estándar *IEEE 802.15.4*, aunque en su mayoría son implementaciones derivadas de otros sistemas operativos y no un diseño nativo. De todos modos, estas implementaciones son completamente funcionales, lo que permite el uso de los diferentes protocolos y algoritmos para WSN que han sido objeto de análisis en esta Tesis Doctoral.

A.3 PLATAFORMAS DE SIMULACIÓN PARA WSN

Los nodos de una WSN cuentan, por lo general, con baterías que los alimentan, así como con chips de radio inalámbrica, como se comentó en la Sección A.1. Basándose en estos elementos, resulta de interés analizar el comportamiento de dichos nodos en escenarios simulados, con anterioridad a un despliegue real. La simulación es una acción

que permite analizar el comportamiento esperado de la red, además de observar la reacción ante situaciones inesperadas. Por otro lado, la simulación es útil, no solo como elemento de análisis, sino como medio para ahorrar gastos ante contratiempos que pueden ser previamente descubiertos vía simulación.

Es por este motivo, por lo que la simulación de WSN se ha vuelto popular. En este sentido, este Anexo presenta los simuladores más utilizados en investigación, realizando un análisis general de ellos y proporcionando información adecuada para la selección de uno de ellos, como método de trabajo.

Las siguientes subsecciones llevan a cabo un análisis de los simuladores más relevantes en investigación de WSN y en la Tabla A.1 se puede consultar un resumen de las principales características de los simuladores analizados en esta sección.

A.3.1 NS-2/NS-3

NS-2 [11] y NS-3 [12] son simuladores de redes basados en eventos discretos, utilizados principalmente en investigación y educación. Estos simuladores se encuentran disponible para su descarga en las webs [11] y [12] respectivamente, y se ofrecen bajo una licencia GNUv2 que permite su uso académico.

NS-2 y NS-3 están escritos en C++ y los protocolos o algoritmos que se deseen simular deben ser escritos también en C++.

En NS-2 los scripts de simulación deben ser definidos en un lenguaje llamado **OTCL**. NS-2 está especialmente diseñado para simular protocolos unicast y multicast basados en IP, aunque debido a que no fue originalmente diseñado para WSN, no escala bien las redes con más de 100 nodos.

OTCL:
OBJECT-ORIENTED
TOOL COMMAND LINE.

NS-3, por otra parte, si está diseñado específicamente para la simulación de redes de comunicaciones, incluyendo modelos para IP, Ethernet, Wi-Fi o LTE. También incluye soporte para algoritmos de enrutamiento específicos como OLSR [81] o AODV.

A.3.2 OMNeT++/OMNEST

OMNeT++ [84] es un simulador modular de eventos discretos orientado a objetos, el cuál ha sido adaptado para modelar el tráfico en redes de telecomunicaciones, protocolos, sistemas distribuidos, etc. OMNeT++ es una versión gratuita basada en una licencia para uso académico y se encuentra disponible para su descarga en [84]. Este simulador posee también una versión comercial llamada OMNEST [13].

El sistema modular de *OMNeT++* se basa en el intercambio de mensajes entre los módulos, conforme a una jerarquía definida por el usuario en los archivos específicos del lenguaje de descripción de red. Los módulos que contienen otros módulos son considerados *módulos compuestos* y los que contienen la funcionalidad o los algoritmos son los conocidos como *módulos simples*. Estos *módulos simples* se implementan usando la biblioteca de simulación de *OMNeT++*. Los módulos necesitan definir además de su funcionalidad, una serie de puertas y enlaces que permitirán el intercambio de los mensajes con el resto de módulos que compongan un nodo.

El sistema de simulación contiene un núcleo que contiene las bibliotecas y las rutinas de control de las simulaciones. Las interfaces de usuario usadas en los modelos a simular también deben ser alojadas en el núcleo de simulación.

Un modelo de simulación debe contener la descripción de la topología en lenguaje **NED**, así como la definición de los mensajes utilizados por el modelo y de los módulos simples. Todos los modelos de *OMNeT++* tienen una estructura modular. El proceso para desarrollar un modelo de simulación contiene, al menos, las siguientes cuatro etapas:

- Descripción de la estructura del sistema mediante el lenguaje NED.
- Implementación de los módulos simples en C++.
- Compilación de los módulos. Generación del modelo y enlace con la biblioteca de simulación.
- Configuración de la simulación, especificando los parámetros para su ejecución.

OMNeT++ permite la simulación de todo tipo de modelos, desde modelos de consumo a energético, a modelos de tráfico de red, pasando por una amplia variedad de modelos adicionales desarrollados por su amplia comunidad y que en muchos casos son agrupados en frameworks, como por ejemplo: *INET* [9], *INETMANET* [22], *Mixim* [10], *Castalia* [6], etc.

A.3.3 TOSSIM

TOSSIM [19] es un simulador de eventos discretos que se encuentra integrado en el Sistema Operativo empotrado *TinyOS*, descrito en la Sección A.2.1 y que provee de una herramienta para la simulación de aplicaciones basadas en el lenguaje de programación *nesC*.

El simulador funciona reemplazando componentes con elementos de simulación. Los componentes pueden ser reemplazados con una am-

SIMULATOR	SIMULATOR TYPE	LANGUAGE	LICENSE	Doc.
NS-2	Discrete Event	C++/OTCL	Open Source	Free
NS-3	Discrete Event	C++	Open Source	Free
OMNeT++	Discrete Event	C++	Open Source	Free
OMNEST	Discrete Event	C++	Comercial	Closed
TOSSIM	Discrete Event	C++/Python	Open Source	Free
Cooja	Discrete Event	C	Open Source	Free

Tabla A.1

Comparativa de Plataformas de Simulación para WSN: Resumen de Características

plia flexibilidad, permitiendo la sustitución de cualquier componente, ya sea por uno de los pertenecientes a la colección de *TOSSIM* o por uno nuevo desarrollado específicamente.

TOSSIM extrae eventos de una cola de eventos y los ejecuta, dependiendo del nivel de simulación, los eventos de simulación pueden representar interrupciones hardware o tareas de alto nivel. Por lo general, las tareas representan eventos de simulación, por lo que al publicar una tarea se produce un retardo de microsegundos hasta que esta es simulada.

TOSSIM, como otros simuladores, se basa en librerías, por lo que se necesita la descripción de la simulación para poder ejecutarla. Este simulador ofrece soporte tanto para la programación en Python como en C++. Aunque tiene una importante desventaja con respecto a otros simuladores, al no ofrecer soporte para la simulación del consumo energético de los nodos.

A.3.4 *Cooja*

Cooja [7] es básicamente una extensión del Sistema Operativo empotrado *Contiki*, descrito en la Sección A.2.2, para JAVA. *Cooja* permite la simulación simultánea de la aplicación, el sistema operativo y el conjunto de instrucciones de código máquina de un determinado nodo. En este simulador se combinan la simulación del hardware de bajo nivel con aplicaciones de alto nivel.

Al encontrarse basado en JAVA, *Cooja* es flexible y extensible mediante el uso de la interfaz nativa de Java JNI. Esto permite la interacción a todos los niveles del sistema, plataforma, software del sistema operativo, control de la radio y modelos de propagación de la radio.

Cooja básicamente simula nodos que ejecutan el sistema operativo *Contiki*. El código para la simulación es obtenido de la propia compilación del núcleo de *Contiki*, de sus procesos y de unos drivers específicos para la plataforma de simulación.

Cooja no se encuentra exento de limitaciones, las principales vienen derivadas de su capacidad de expansión, ya que esto provoca una baja eficiencia. La simulación de todos los elementos que conforman una red por separado requiere de una gran cantidad de cálculos en paralelo, lo que conlleva la simulación de un limitado número de nodos, como en el caso de *ns-2*. Otra limitación es el hecho de utilizar una interfaz gráfica, lo que conlleva asociado un mayor uso de recursos y provoca un mal desempeño en términos de tiempo de simulación.

B

OMNET++ TUTORIAL

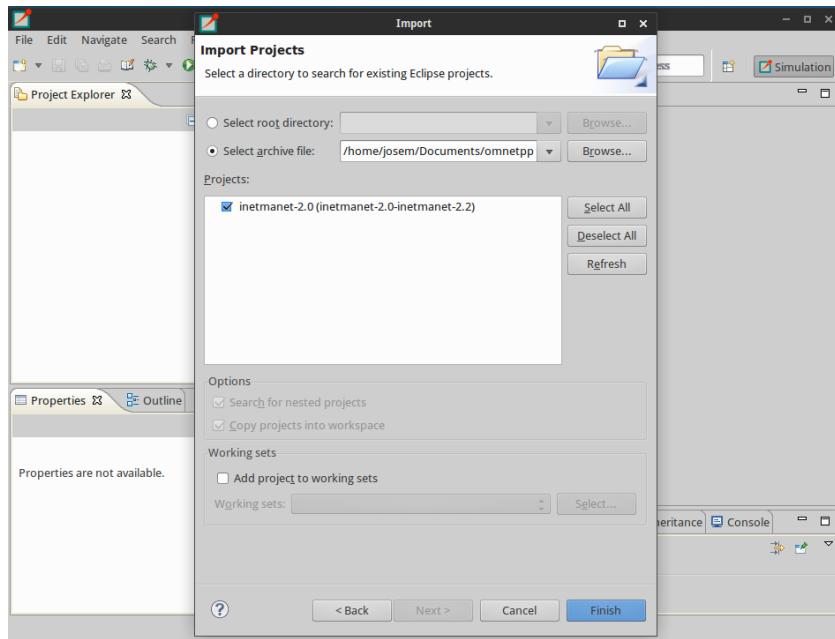
CONTENTS

B.1	First Step: Installing a Framework	178
B.2	From the TicToc Tutorial to a real Node	179
B.3	Useful Functions and tips	187
B.3.1	cModule Class	188
B.3.2	ModuleAccess.cc	188
B.3.3	cMessage	188
B.3.4	cPacket Class	189
B.3.5	Additional Tips	189
B.4	Example of a real network	190
B.5	Additional Documentation	190

The present tutorial tries to complete the basic concepts provided by the TicToc tutorial [15] from OMNeT++ [84]. Therefore, it is assumed that you already have a working OMNeT++ installation, and the Tic-Toc tutorial has been completed. It is also necessary a minimum C++ knowledge. The TicToc tutorial [15] has explained about the creation and handling of simple networks, message and timers definition, etc. However, a real network is quite more complex than the last OMNeT++ example.

On the other hand, if OMNeT++ has not been installed yet, the *Installation Guide*, provided by the OMNeT++ software explains the installation process. It should be pointed out that this tutorial is based on the OMNeT++ IDE, because it is more intuitive to follow the first steps.

Figura B.1
Importing a Project
into workspace.



B.1 FIRST STEP: INSTALLING A FRAMEWORK

The OMNeT++ installation provides several examples, but they are not enough to develop a real network. Instead, this tutorial uses a framework, which is a set of models and resources that has been developed to simulate real standards, protocols, etc.

There are different frameworks available in the OMNeT++ webpage. As it was explained in the Chapter 5, this Thesis makes use of *INET-MANET* framework, which includes a wide range of protocols, some of the mainly used for mobile ad-hoc networks. It can be downloaded through the OMNeT++ main page [84], accessing to the Simulation Models area.

Once the framework is downloaded, copy the .zip file in the workspace folder and import it:

File->Import...>General>Existing Projects into Workspace, as it is shown in the Figure B.1.

After that, it is necessary to build the project (Project->Build All) loading all the libraries. If an error appears, please consult the framework help files.

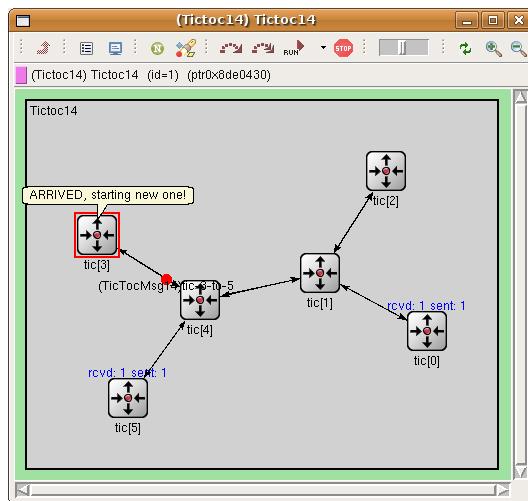


Figura B.2
TicToc Example Net-
work.

B.2 FROM THE TICTOC TUTORIAL TO A REAL NODE

If the TicToc tutorial is finished, it should be possible to answer the following questions:

- What is a module parameter and how can it be defined and read?
- What is a module gate and how can you use it?
- How does the OMNeT++ send() function work?
- How does the OMNeT++ scheduleAt() function work?
- What is a message and how can be defined a message class?

Those are basic concepts that is necessary to know in order to successfully follow this tutorial. Otherwise, it might need to revise it again.

The real network described in the TicToc tutorial helps with the understanding of the simulator behaviour, but is not even closed to a real network example. The reason is clear, it only includes simple modules, as Figure B.2 shows.

The modules used are in responsible for creating, routing and sending the messages. However, in a real node those different behaviours are defined through different modules, as Figure B.3 shows.

Therefore, this tutorial shows the development of a simple network, which contains two simple nodes, but using the concept of submodules. Two submodules are defined, one is a high-layer module and another is a low-layer module. The high-layer one creates the messages and the low-layer one sends them. The Figure B.4 shows the structure of the desired network.

Figura B.3
Real Simple Node.

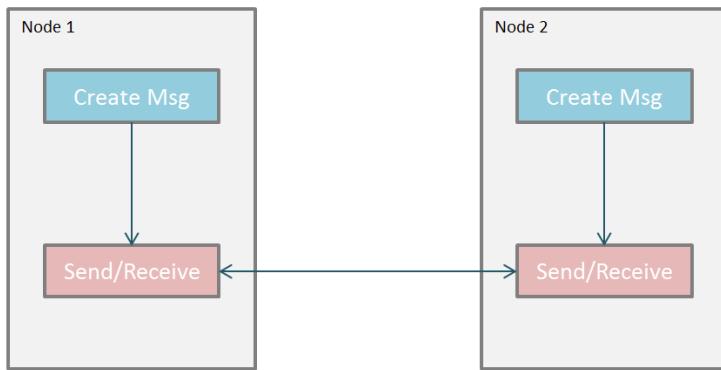
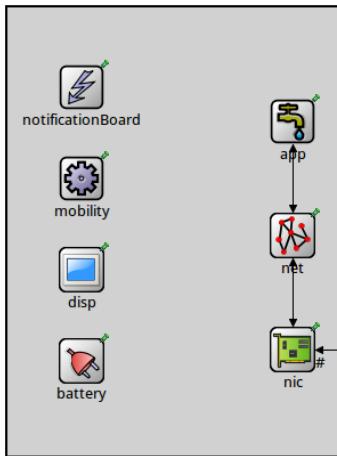


Figura B.4
Simple Network Topology.



In order to create a compound network, it is necessary to use a bottom-up approach. Therefore, this tutorial starts developing the submodules. The creation process is similar for both submodules, hence this tutorial only explains the creation of the high-layer module. Firstly, create a new folder inside the src folder called *tutorial*. Then, create a new *Simple Module* inside that folder, as Figure B.5 shows.

It should be done in the same way for the other submodule. Now, the .ned files of the submodules need to be written. The source is really simple as B.1 shows.

To provide the functionality, the source and header files in C++ should be defined. These files have been already partially created by the OMNeT++ IDE. The source code is presented in B.2.

The code is quite simple, similar to the first steps of the TicToc tutorial. The new functions involve the handling of gates. There are two ways to define a gate inside a module, as an integer and as a cGate (OMNeT++ class).

That gate definition can be used in the send function, so it is quite useful. The other addition is the evaluation of the gate from where

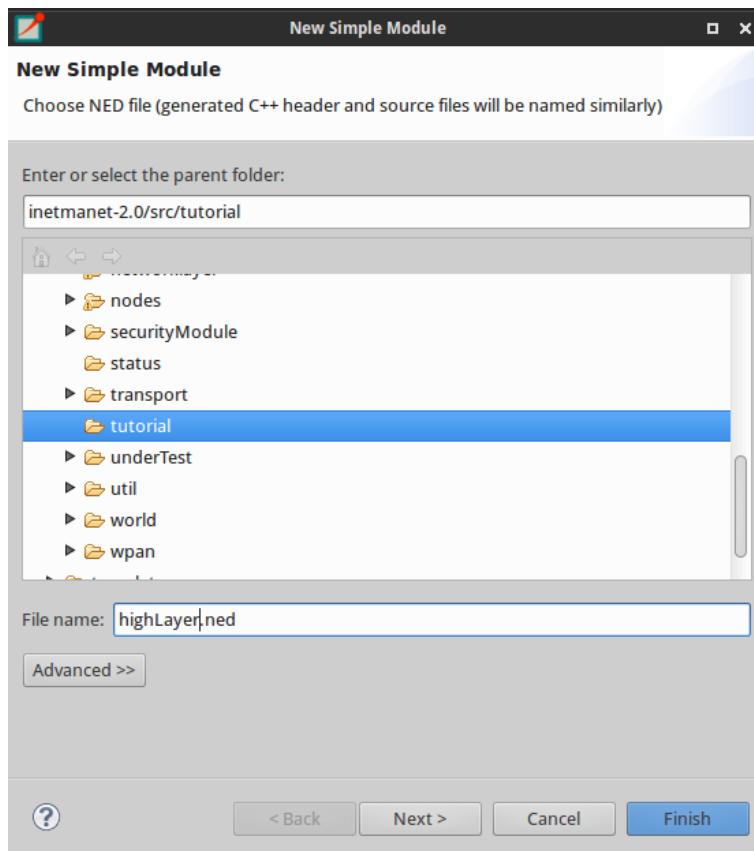


Figura B.5
New Simple Module
Dialog Windows.

a message has arrived. That information can be extracted from the messages, and then make a decision depending on the source.

Código B.1
ned low layer file example.

```

1 package inet.tutorial;
2 //
3 // TODO auto-generated module
4 //
5 simple LowLayer
6 {
7     parameters:
8         @display("i=block/tx");
9         gates:
10            // The module has 3 gates. One will receive the ←
11            // messages from
12            // the high layer. The other ones will be connected ←
13            // to the
14            // node gates
15            input highIn;
16            input nodeIn;
17            output nodeOut;
18 }
```

Código B.2
ned high layer file example.

```

1 package inet.tutorial;
2 //
3 // TODO auto-generated module
4 //
5 simple HighLayer
6 {
7     parameters:
8         // The variable DepartureTime allows you to define the
9         // time between messages
10        int departureTime @unit("s")=default(2s);
11        @display("i=block/source");
12        gates:
13            // This module has only one gate, because it only has←
14            // to
15            // send messages to the LowLayer
16            output out;
17}

```

Código B.3
Low layer c++ class header file.

```

1 #ifndef __INETMANET_2_0__LOWLAYER_H__
2 #define __INETMANET_2_0__LOWLAYER_H__
3
4 #include <omnetpp.h>
5
6 class LowLayer: public cSimpleModule
7 {
8     public:
9         long receivedMsg;
10        long sendMsg;
11        // Defining a gate through its id
12        int nodeGateId;
13
14     protected:
15        virtual void initialize();
16        virtual void handleMessage(cMessage *msg);
17        virtual void updateDisplay();
18    };
19
20 #endif

```

Once the submodules have been defined, the node module should be defined. To do that, it is necessary to create a compound module in the same way that the previous simple modules were created:

New->Compound Module, then, select an empty compound module. Figure B.6 shows the process.

Código B.4
Low layer c++ class file.

```

1 #include <lowLayer.h>
2
3 Define_Module(LowLayer);
4
5 void LowLayer::initialize()
6 {
7     sentMsg = 0;
8     receivedMsg = 0;
9     // Defining a gate through its id
10    nodeGateId = gate("nodeIn")->getId();
11    WATCH(sentMsg);
12    WATCH(receivedMsg);
13 }
14
15 void LowLayer::handleMessage(cMessage *msg)
16 {
17     // The module can receive messages from the high layer or←
18     // from the node
19     // We check the origin and carry out a simple count
20     if(msg->getArrivalGateId()==nodeGateId){
21         receivedMsg++;
22         delete msg;
23     }
24     else{
25         sentMsg++;
26         send(msg, "nodeOut");
27     }
28     if(ev.isGui())
29         updateDisplay();
30 }
31 void LowLayer::updateDisplay()
32 {
33     char buf[40];
34     sprintf(buf, "rcvd: %d sent: %d", receivedMsg, sentMsg←
35             );
36     getDisplayString().setTagArg("t", 0, buf);
37 }
```

Once the compound module have been created, the gates of the node and the connections should be defined. The node should have two gates, one to send messages and another one to receive them. The connection between the submodules is simple, the highLayer out-gate is connected to an input-gate from the lowLayer submodule. The other lowLayer gates are connected to the nodes gates. That behaviour is shown in B.7.

Código B.5

High layer c++ class header file.

```

1 #ifndef __INETMANET_2_0__HIGHLAYER_H__
2 #define __INETMANET_2_0__HIGHLAYER_H__
3
4 #include <omnetpp.h>
5
6 class HighLayer: public cSimpleModule
7 {
8     public:
9         // Instead insert the name of the gate, a gate can be ←
10        // read as a parameter
11        int departureTime;
12        cGate *lowLayer;
13
14     protected:
15         virtual void initialize();
16         virtual void handleMessage(cMessage *msg);
17 };
18 #endif

```

Código B.6

High layer c++ class file.

```

1 #include <highLayer.h>
2
3 Define_Module(HighLayer);
4
5 void HighLayer::initialize()
6 {
7     // Initialize the gate and the departureTime parameter
8     lowLayer = gate("out");
9     departureTime = par("departureTime");
10    // Scheduling the send of a new message
11    cMessage *msg = new cMessage("appMsg");
12    scheduleAt(simTime() + departureTime, msg);
13 };
14
15 // This module will only receive scheduled messages
16 void HighLayer::handleMessage(cMessage *msg)
17 {
18     // Sending a copy of the message
19     cMessage *copy = msg->dup();
20     send(copy, lowLayer);
21     // Scheduling the send of a new message
22     scheduleAt(simTime() + departureTime, msg);
23 };

```

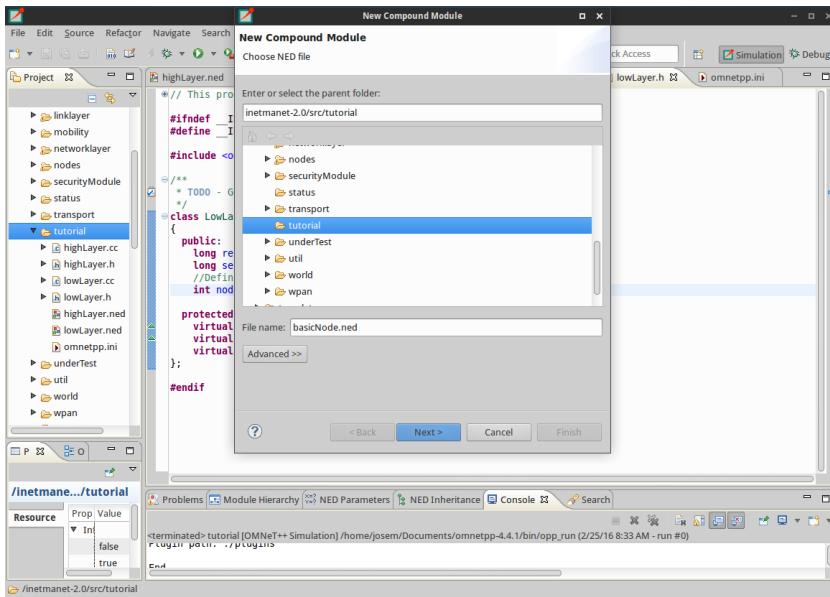


Figura B.6
New Compound Module Dialog Window

Código B.7
Basic node ned file.

```

1 package inet.tutorial;
2 /**
3 // TODO auto-generated module
4 /**
5 module BasicNode
6 {
7     parameters:
8         // The @node label defines the current module as a node
9         // . It is
10        // useful to identify them quickly within the
11        // simulation
12        @node();
13        @display("i=device/pocketpc;bgb=310,269");
14
15     gates:
16         input in;
17         output out;
18
19     submodules:
20         highLayer: HighLayer{
21             parameters:
22                 @display("i=block/source;p=148,42");
23         }
24         lowLayer: LowLayer{
25             parameters:
26                 @display("i=block/tx;p=148,169");
27         }
28
29     connections:

```

```

28     highLayer.out -> lowLayer.highIn;
29     in -> lowLayer.nodeIn;
30     lowLayer.nodeOut -> out;
31 }
```

Now, it is possible to create the network, like it has been done during the TicToc tutorial.

Código B.8
Basic network ned file.

```

1 package inet.tutorial;
2 //
3 // TODO auto-generated module
4 //
5 module BasicNetwork
6 {
7     submodules:
8         node1: BasicNode{
9             parameters:
10                @display("i=device/pocketpc;bgb=287,119;p=94,80");
11            }
12         node2: BasicNode{
13             parameters:
14                @display("i=device/pocketpc;bgb=125,119;p=243,80");
15            }
16
17     connections:
18         node1.out -> { delay = 100ms; } -> node2.in;
19         node2.out -> { delay = 100ms; } -> node1.in;
20 }
```

To test the modules created, it is necessary to define an .ini file as B.9 presents.

Código B.9
ini configuration file example.

```

1 [General]
2 network = inet.tutorial.BasicNetwork
3
4 # Define different times between messages
5 **.node1.highLayer.departureTime = 3s
6 **.node2.highLayer.departureTime = 5s
```

Then, the project needs to be built again. After built it, the simulation can be launched, by clicking the right button over the .ini file.

At this point, it is possible to explain the essential submodules that must take part in a real node.

- Application Module: The main function of an application module is to generate all the traffic in the network. The `highLayer` submodule created could belong to this group. Normally, they include a wide range of options, different parameters as the destination node, and the size of the packets, etc. Other examples of an application module could be the implementation of a ping application or an UDP application.
- Network Module: The network module have several functions, but its main function is to define the network topology, i.e. a star network topology; or a protocol for the network, i.e. IPv4. So it is responsible of routing the messages and adapt them to the network interface card layer used.
- Network Interface Card (NIC): This module it is responsible of sending messages between nodes, according to the standard used, as could be IEEE 802.15.4 or IEEE 802.11 in wireless nodes.
- Notification Board: The notification board acts as an intermediary module between nodes. The modules publish state changes or share useful information. If a module wants to know about any of those information updates, it must subscribe to them.
- Mobility: The mobility module is used to define the position of the nodes, either static or dynamic position.
- Battery: The battery module is not essential, but it is useful to obtain energy consumption information.

B.3 USEFUL FUNCTIONS AND TIPS

Before starting with the development of a real network, it is necessary to learn some functions provided by the simulator. OMNeT++ is an event simulator, as it is commented in the Annex A, therefore, tasks are not in a continuously execution. The modules work individually and the complete execution is achieved passing messages between the different elements of a simulation.

This fact implies that a module only knows its variables, thus, you need to write some code if you want to access to variables or information from other modules. The `cModule` class defined by OMNeT++, among other things, provide some functions that allow accessing the node/network hierarchy.

B.3.1 *cModule Class*

In the same way that the *cMessage* class defines a message, a module is defined. There are some public member functions of this class really useful for a project development such as:

- `getFullName()`: This function returns a const char pointer with the full name of the module, including an index, if it exists. It is really useful to identify string variables on a message.
- `getParentModule()`: This function returns the parent module pointer. It is useful if it is necessary to access to public members functions/variables from modules with a higher hierarchy in the network tree.
- `getSubmodule(moduleName)`: This function returns the module pointer of a submodule directly accessible in the hierarchy. It is useful to access to public members functions/variables from modules with a lower hierarchy in the network tree.
- `getModuleByPath(modulePath)`: This function returns the module pointer of any module in the network tree.

B.3.2 *ModuleAccess.cc*

The *ModuleAccess* also defines an interesting function in order to move in the module tree:

- `findContainingNode()`: This function returns the node module pointer, which contains the actual functions. It is really useful to create modules, which should be integrated by other users. Then, it allows getting the node pointer regardless of the node structure.

B.3.3 *cMessage*

The `handleMessage` functions are the fundamental functions. Since OMNeT++ is a discrete event simulator, those functions define the behaviour after an event occurs. All the messages arrive at this function, so it is completely necessary to correctly identify the gate and the kind of message. We have already introduced how to filter the arrival gate.

Finding out the kind of message could be done in several ways. Therefore, some public member functions from the *cMessage* class are presented below in order to contribute to that identification.

- `setKind()/getKind()`: The `cMessage` has a member variable to define the kind of message. These member functions write/read the variable. So, if the message is identified, it is possible to know its kind.
- `getArrivalGate()/getArrivalGateId()`: These functions allow obtaining a `cGate` pointer or the gate identifier.
- `getArrivalModule()/getArrivalModuleId()`: They offer a similar functionality than the previous ones, but referring to the source module of the message.
- `isSelfMessage()`: If a message have been scheduled for an event, it is possible to identify that kind of message using this boolean function.

However, if there are inherited classes from the `cMessage` class, it is also possible to identify the message, casting its class. In order to do that, the function `dynamic_cast()` is really useful. It is commonly used in C++ language, so it is not necessary to add any information about it. The TicToc tutorial also introduced the OMNeT++ `check_and_cast()` function. This function can be used to create and detect errors if the `dynamic_cast()` function returns a zero. It is really useful if a module can only handle a concrete `cMessage` subclass.

B.3.4 *cPacket Class*

This class is inherited from `cMessage`, allowing to include encapsulated messages into an existing message. So, the additional functions provided for this class are explained below.

- `encapsulate(cPacket)`: The `encapsulate` function add the packet/message given as a parameter to the `cPacket` variable.
- `decapsulate()`: The `decapsulate()` function returns the pointer to the packet/message attached to the `cPacket` variable.

Further information about these and other interesting functions can be found inside the OMNeT installation folder: `UserPath/omnetpp-x.x.x/doc/api/main.html` within Simulation core classes.

B.3.5 *Additional Tips*

During the development of a code, may be necessary to initialize a module in a different way. It is possible to change the `initialize()` function for `initialize(int aStage)`, but then, it is also necessary to modify

the virtual function numInitStages(). B.10 shows a function redefinition example, which allows a three steps initialization.

Código B.10
Module initialization example.

```
1 virtual void numInitStages() const {return 3;};
```

Is is important to note that integration is an essential part of a complete network design.

B.4 EXAMPLE OF A REAL NETWORK

The next step is to handle a real network example. In order to do that, it is necessary to work with a basic example provided by the INETMANET framework. Open the folder examples>wpan>StarNet.

Here, it is possible to find a .ned file, which describes the network. It only contains one type of node, but it is declared as a vector. So the number of nodes can be modified, changing the value of the numHosts parameter.

By pressing on the node class, it is possible to see the structure. There is a traffic generator, a network layer defining the topology, and a network interface card. The traffic generator is similar to the one previously developed on the attached tutorial. The network layer implements a star topology, so all the nodes send packets through the network root. The NIC implements a standard IEEE 802.15.4-based radio (MAC and PHY layers).

The NIC source code (.cc) is difficult to understand for a beginner. However, the code of the network layer and the application layer are easier.

B.5 ADDITIONAL DOCUMENTATION

OMNeT++ offers a complete software manual. It fully describes the different features of OMNeT++, providing useful information for project development. There is an interesting Google [14] group about OMNeT++, where different users share their software and modules problems and doubts.

C

DARP TECHNICAL MANUAL

CONTENTS

c.1	Specifications	191
c.1.1	Related Documents	192
c.2	Implementation Details	192
c.2.1	State Machine	193
c.2.2	Virtual PAN ID	193
c.2.3	Available Operations in DARP	194
c.2.4	Frame Format	198

This protocol results by the combination of the *IEEE 802.15.4* and the *DARP* routing algorithm. *DARP* is an algorithm developed by Francisco J. Estévez.

c.1 SPECIFICATIONS

This section describes the specifications for *DARP* as long as the related documentation.

c.1.1 Related Documents

- IEEE 802.15.4:2006
- IEEE 802.15.4:2011
- Atmel BitCloud Developers Guide

The following specifications detail different key aspects about the protocol design:

- All the nodes must be *IEEE 802.15.4* FFD-definition compatible.
- There are two different roles for a node: End Node (EN) and Virtual Coordinator (VC).
- All nodes but the network root store, at least two network address, its own and its father address.
- A VC only stores the routing tables of nodes below it in the network topology.
- The routing is based on sub-networks identifications (VID).
- A VC is member of its father sub-network and it manages its own sub-network.
- All the nodes but the network root execute DRSP (Dynamical Routing Selection Process) to choose its own role in the network.
- Each certain time, it is necessary to control the status of the network, checking if every node is connected.
- A network is self-healed if a node or branch drops.
- It exists a state machine with 3 different states: Searching, Waiting and Connected.

c.2 IMPLEMENTATION DETAILS

DARP is designed as an *IEEE 802.15.4*-based network layer, the resulting network stack is described in the Figure 4.1 in Chapter 4.

The next sections fully describe the technical details concerning DARP protocol.

c.2.1 State Machine

Figure C.1 shows the *DARP* state machine, which is based on 3 states:

- Searching: At this state, a node searches and tries to connect to an existing sub-network. Once it is done, it chooses its role in the network, changing to Awaiting (if it is a VC) or to Connected (if it is an EN).
- Awaiting: A VC is at this state, waiting until its network identification is received. The VC can receive and send messages but cannot route them. Once it receives the vID for its own network, it changes its state to Connected.
- Connected: This state represents a fully connection to the network. In the case of an EN it is directly connected to a VC. In the case of a VC, it means that it has its own sub-network ID and it can accept new nodes and route packets.

A disconnection request results in a state change. A node can change to Searching (temporal disconnection request) or to Awaiting (fully disconnection request).

c.2.2 Virtual PAN ID

The *IEEE 802.15.4* standard defines that every network must have a Personal Area Network identification (PAN ID). Therefore, *DARP* is based on the concept of virtual sub-networks, which means different logical networks with their own virtual PAN ID under a general *IEEE 802.15.4* network.

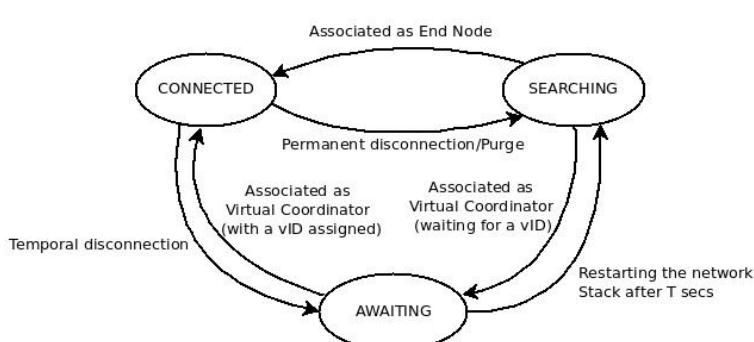


Figura C.1
DARP State Machine.

Figura C.2
Message flow in an EN Association Operation.

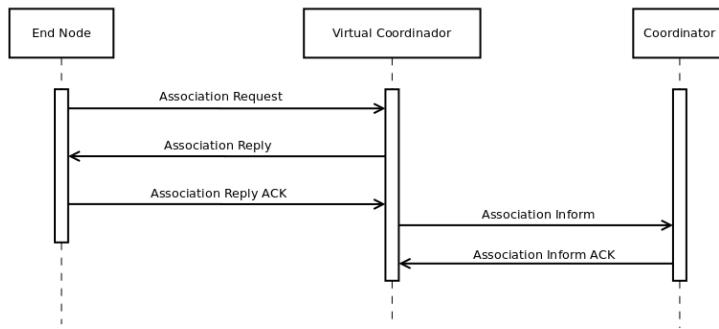
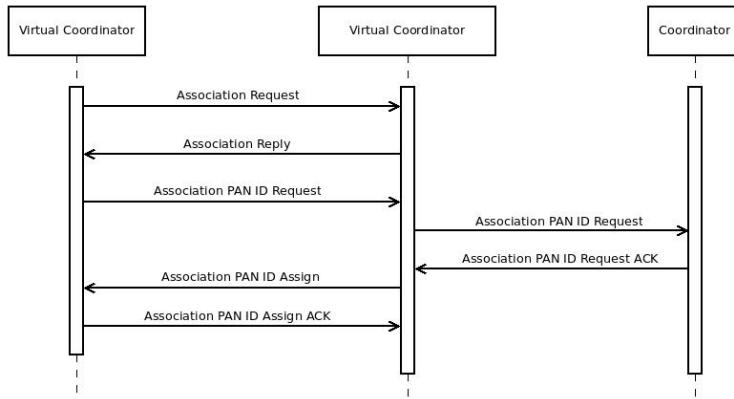


Figura C.3
Message flow in a VC Association Operation.



c.2.3 Available Operations in DARP

This section describes the different operations carried out by this protocol.

Node Association

The following message sequence describes the association process between nodes, which is described in Chapter 4, differentiating between EN and VC.

There are two possible results with different behaviours in terms of network communication:

- Successful Association:

The role selection is successfully based on DRSP, fully described in Figure 4.7, as it is shown in Figures C.2 and C.3.

- Unsuccessful Association:

If a node does not connect, the state machine remains at Connected, trying continuously to connect the network.

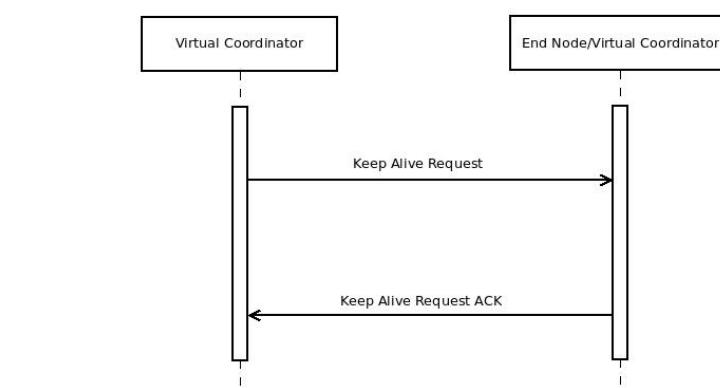


Figura C.4
Message flow in a successful Status Control Operation.

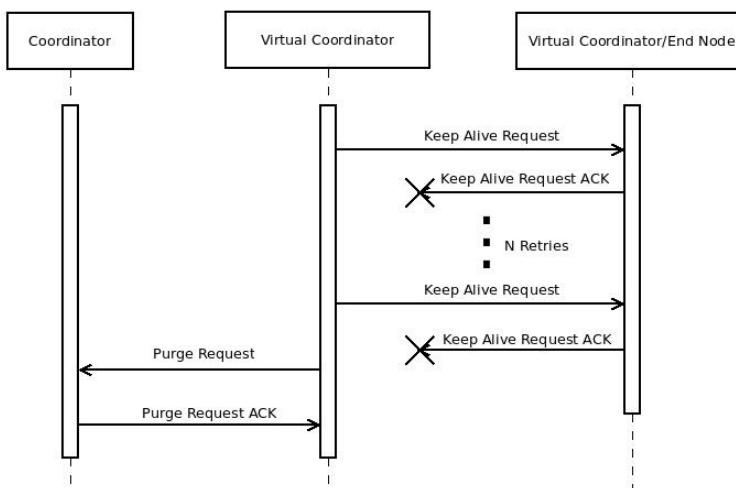


Figura C.5
Message flow in an unsuccessful Status Control Operation.

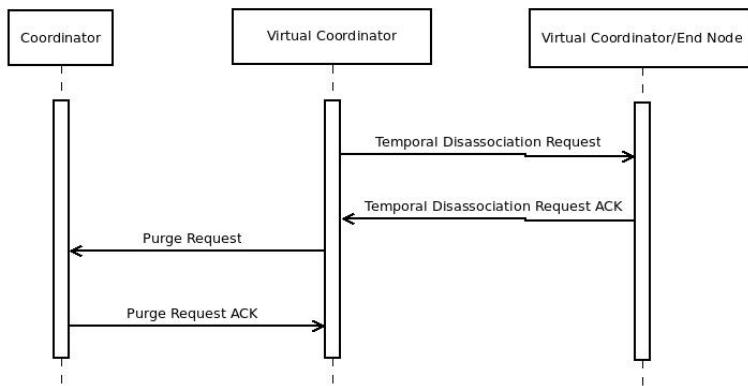
Network status control

The following message sequence describes the status control process between nodes, which is described in Chapter 4, also known as keep alive process.

There are two possible scenarios. If the answer is received in a time (T_{ack}) or if it is not.

- Successful network status control can be seen in Figure C.4.
- Unsuccessful network status control:
If a VC sends a keep alive message and it is not answered before T_{ack} , the VC waits T_{down} before temporarily remove that node from the local routing table. Figure C.5 shows an example the message flow.

Figura C.6
Message flow in a successful Temporal Disassociation Operation.



Temporal disassociation (TO-DO)

The following message sequence describes the temporal disassociation between nodes.

For several reasons a VC could request to an EN to temporally disconnect. There are two possible answers:

- Successful disconnection:
The VC sends the message and receives an ACK, therefore removes the node from the local routing table.
- Unsuccessful disconnection:
The VC tries N times, if the process fails anyway, a purge process starts, removing the node unilaterally. Figure C.6 shows a graphical example.

Full disassociation (TO-DO)

The following message sequence describes the full disassociation between nodes.

For several reasons a VC could request to an EN to fully disconnect. There are two possible answers:

- Successful disconnection:
The VC sends the message and receives an ACK, removing immediately the node from the local routing table.
- Unsuccessful disconnection:
The VC tries N times, if the process fails all of them, then a purge process starts, removing the node. Figure C.7 presents the process.

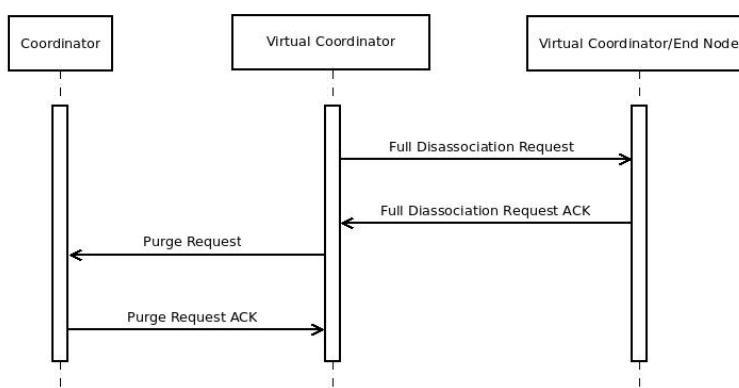


Figura C.7
Message flow in a successful Full Disassociation Operation.

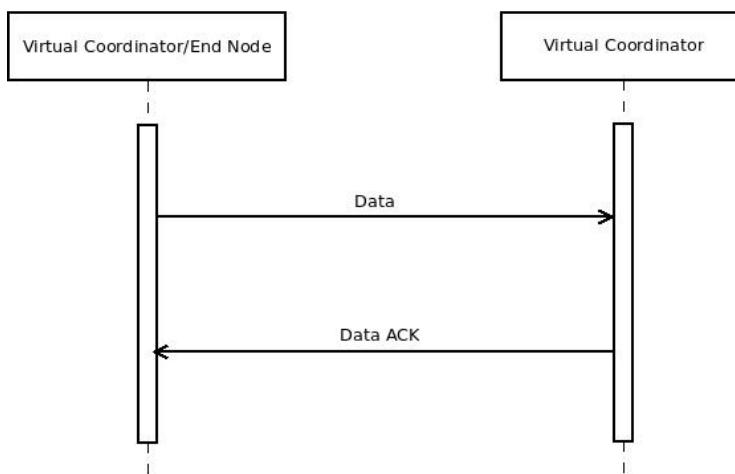


Figura C.8
Message flow in a Data Sending Operation.

Data Message

The following message sequence describes the sending of a data message between nodes, explained in Chapter 4.

- Successful sending, which is shown in Figure C.8.

It is necessary to point that the Keep Alive Timers (T_{alive}) are reset every time that a data packet is successfully received.

- Unsuccessful sending:

If the communication fails the VC retries $DARAL_{MAX_TRIES}$ times. If the error continues after all the retries, the VC waits until the next network status control phase to obtain information about the link. Once the network status control phase is over, there are two different possible results:

- Disconnected node: The VC informs to the network root about the drop of the node.

Tabla C.1
DARP control header
Description.

Offset	Field	Size
0	OPERATION CODE	1B
1	PACKET LENGTH	1B
2	ROUTING TYPE	1B
3	HOP LIMIT	1B
4	CHECKSUM	2B
6	MESSAGE IDENTIFICATOR	1B
7	SOURCE VIRTUAL NETWORK IDENTIFICATOR	2B
9	DESTINATION VIRTUAL NETWORK IDENTIFICATOR	2B
11	SOURCE ADDRESS	4B
19	DESTINATION ADDRESS	4B

- Connected node: The node is connected and can now receive data packets.

C.2.4 Frame Format

DARP frames are IEEE 802.15.4-based, see the standard documentation for more references about control headers. The following table is reproduced here for simplicity but it is contained as Table 4.1 in Chapter 4.

As Table 4.1 shows, DARP header sums up to 27 Octects, containing 10 different fields, which are detail in the following sections.

Operation Code (OP_CODE)

There are specific messages for this network protocol. The structures are predefined for everyone. The payload for each message is also set. This header field uses 1 Byte and can take the values represented in Tables C.2, C.3, C.4 and C.5.

Message Type	OP_CODE
ASSOCIATION_REQUEST	0x00000001
ASSOCIATION_REPLY	0x00000010
ASSOCIATION_REPLY_ACK	0x00000011
ASSOCIATION_INFORM	0x00000100
ASSOCIATION_INFORM_ACK	0x00000101
ASSOCIATION_PAN_ID_REQUEST	0x00000110
ASSOCIATION_PAN_ID_REQUEST_ACK	0x00000111
ASSOCIATION_PAN_ID_ASSIGN	0x00001000
ASSOCIATION_PAN_ID_ASSIGN_ACK	0x00001001

Tabla C.2
Available packets type for a node association.

Message Type	OP_CODE
KEEP_ALIVE_REQUEST	0x00001010
KEEP_ALIVE_REQUEST_ACK	0x00001011
PURGE_REQUEST	0x00001100
PURGE_REQUEST_ACK	0x00001101

Tabla C.3
Available packets type for a network control status.

Message Type	OP_CODE
TEMPORAL_DISSASOCIATION_REQUEST	0x00001110
TEMPORAL_DISSASOCIATION_REQUEST_ACK	0x00001111
FULL_DISSASOCIATION_REQUEST	0x00010000
FULL_DISSASOCIATION_REQUEST_ACK	0x00010001

Tabla C.4
Available packets type for a node disassociation.

Message Type	OP_CODE
DATA	0x00010100
DATA_ACK	0x00010101

Tabla C.5
Available packets type for a data sending.

Packet Length (PACKET_LENGTH)

This field needs 1 Byte and it will take the value 0 in the control messages, so that means, that it should be used only in data sending.

Routing Type (ROUTING_TYPE)

This field represents the routing type for a packet in a certain moment. It needs 1 Byte and it can take the values from the Table C.6.

Tabla C.6
Available Routing types in DARP

Routing Type	CODE	Associated Task
GATEWAY	0x0001	Parse and route to the network root, using the links between VCs.
FORWARDING	0x0010	Parse and route to a destination sub-network. It only analyzes the Destination PAN_ID and route the packet. If the destination PAN_ID is the own, it should change the routing type to PARSING.
PARSING	0x0011	It represents that a node should process the packet.

Hop Limit (HOP_LIMIT)

It is a field of 1 Byte size and it limits the number of hops of a packet. At the moment is not used and it is under revision to deprecate it.

Error Verification (CHECKSUM)

The error verification needs 2 Bytes and its calculation is made by the sum of every field in the header, unless the routing type field and the checksum itself. The result is stored in the CHECKSUM field. If the CHECKSUM has errors, the packet is discarded, there is no ACK and the destination node waits until another message arrives.

Message Identification (MESSAGE_ID)

This field needs 1 Byte and it is used to identify uniquely a message. If it is necessary to resend a message, the MESSAGE_ID identifies the copy received and it allows controlling the traffic.

Origin Virtual Sub-Network Identification (SOURCE VIRTUAL PAN_ID)

It represents the vID of the origin node. It has 2 Bytes size. If a node sends a BROADCAST, it sets this field to 0.

Destination Virtual Sub-Network Identification (DESTINATION VIRTUAL PAN_ID)

It represents the vID of the destination node. It has 2 Bytes size. If a node sends a BROADCAST, it sets this field to 0.

Origin Address (SOURCE ADDRESS)

It is a header field with 8 Bytes size and it represents the DARP address of the source node.

Destination Address (DESTINATION ADDRESS)

It is a header field with 8 Bytes size and it represents the DARP address of the destination node.

If the message is broadcasted, this field takes the value 0xFFFFFFFFFFFFFFF, following the *IEEE 802.15.4:2006* and *IEEE 802.15.4:2011* standards definition.

Parte IX

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- [1] *2.4 ghz IEEE 802.15.4 - zigbee-ready rf transceiver.* <http://www.ti.com/lit/ds/symlink/cc2420.pdf>.
- [2] *Arduino.* <http://www.arduino.cc/>.
- [3] *ATMEL atmega328p.* <http://www.atmel.com/Images/doc8161.pdf>.
- [4] *Bluetooth device list.* <https://www.bluetooth.org/tpg/listings.cfm>.
- [5] *Bluetooth special interest group (SIG).* <https://www.bluetooth.com/>.
- [6] *Castalia wireless sensor network simulator.* <https://castalia.forge.nicta.com.au/index.php/en/>.
- [7] *Contiki OS java.* <https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja>.
- [8] *FreeRTOS, a free open source rtos for small embedded real time systems.* <http://www.freertos.org/>.
- [9] *INET: Models for the internet stack.* <https://inet.omnetpp.org/>.
- [10] *Mixim: Mixed simulator.* <http://mixim.sourceforge.net/>.
- [11] *The network simulator version 2.* <http://www.isi.edu/nsnam/ns/>.
- [12] *The network simulator version 3.* <https://www.nsnam.org/>.
- [13] *OMNEST simulation software.* <https://omnest.com/>.
- [14] *OMNeT++ discussion group.* <https://groups.google.com/forum/#!forum/omnetpp>.
- [15] *OMNeT++ tictoc tutorial.* <https://omnetpp.org/doc/omnetpp/tictoc-tutorial/>.
- [16] *Redbee-econotag.* <http://mc1322x.devl.org/>.
- [17] *Smart dust: Autonomous sensing and communication in a cubic millimeter.* <http://robotics.eecs.berkeley.edu/~pister/SmartDust/>.
- [18] *Texas instruments MSP430 datasheet: Mixed signal microcontroller.* <http://www.ti.com/lit/ds/slas723f/slas723f.pdf>.
- [19] *Tiny OS simulator.* <http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM>.

- [20] *Wasp mote - wireless sensor networks 802.15.4 zigbee mote.* <http://www.libelium.com/products/waspmote>.
- [21] *Wireless fidelity alliance (Wi-Fi).* <http://www.wi-fi.org/>.
- [22] Ariza A.: *Inetmanet framework for wireless sensor and ad-hoc networks using omnet++.* Last Access March, 2016. <https://github.com/aarizaq/inetmanet-2.0>.
- [23] Forster A. and Murphy AL.: *FROMS: A failure tolerant and mobility enabled multicast routing paradigm with reinforcement learning for wsns.* Ad Hoc Networks, 9:940–965, 2011.
- [24] Basu K. Agah A. and Das S.K.: *Security enforcement in wireless sensor networks: A framework based on non-cooperative games.* Pervasive Mobile Computing, 2:pp. 137–158, 2006.
- [25] ZigBee Alliance: <http://www.zigbee.org/>. Last Access March, 2016.
- [26] Sreekanth G. Arivudainambi D. and Balaji S.: *Energy efficient sensor scheduling for target coverage in wireless sensor network.* In *Wireless Communications, Networking and Applications, WCNA 2014*, December 2014.
- [27] Molina J. Barbancho J., León C. and Barbancho A.: *Giving neurons to sensors: Qos management in wireless sensors networks.* In *Proceedings of the IEEE conference on emerging technologies and factory automation ETFA*, September 2006. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4178222&tag=1.
- [28] Sá Silva J. Camilo T., Carreto C. and Boavida F.: *An energy-efficient ant-based routing algorithm for wireless sensor networks.* Ant Colony Optimization and Swarm Intelligence, 4150:49?59, 2006.
- [29] Huang C.F. and Tseng Y.C.: *The coverage problem in a wireless sensor network.* Mobile Networks and Applications, 10:519–528, 2005.
- [30] Wu T.K. Cheng R.H. and Yu C.W.: *A highly topology adaptable ad hoc routing protocol with complementary preemptive link breaking avoidance and path shortening mechanisms.* Wireless Networks, 16:1289–1311, 2010.
- [31] V. Minic D. Culibrk, D. Vukobratovic and M. Alonso Fernandez: *Sensing Technologies for Precision Irrigation.* Springer, 1st edition, 2014, ISBN 978-1-4614-8329-8.
- [32] Johnson D.B. and Maltz D.A.: *Dynamic source routing.* Ad-hoc wireless networks. Mobile Computing, 15:3?181, 1996.
- [33] Han R. Deng J. and Mishra S.: *Insens: Intrusion-tolerant routing for wireless sensor networks.* Computer Communications, 29:216–230, 2006.
- [34] Marot M. Diallo, C. and M. Becker: *Link quality and local load balancing routing mechanisms in wireless sensor networks.* In *Proceedings of the Sixth Advanced International Conference on Telecommunications (AICT)*, pages 306–315, May 2014. <http://ieeexplore.ieee.org/xpl/abstractKeywords.jsp?arnumber=5489814&tag=1>.

- [35] Wang D.J.: *Clustering mesh-like wireless sensor networks with an energy-efficient scheme*. International Journal of Sensor Networks, 7:199–206, 2010.
- [36] Sakkari D.S. and Basavaraju T.G.: *Gcct: A graph-based coverage and connectivity technique for enhanced quality of service in wsn*. Wireless Personal Communication, 85:1295–1315, 2015.
- [37] Gronvall B. Dunkels A. and Voigt T.: *Contiki a lightweight and flexible operating system for tiny networked sensors*. In *Proceedings of the 9th annual IEEE International Conference on local Computer Networks*, pages pp: 455–462, October 2004.
- [38] F.J. Estevez: *Daral simulation code for omnet++*. Last Access March, 2016. <https://github.com/fjestevez/DARP>.
- [39] González J. Estevez F-J., Rebel G. and Gloesekoetter P.: *Darp: Dynamic and adaptive radio protocol for wireless sensor networks*. Electronic Letters, 50:122–123, January 2014.
- [40] Arcos J.L. Fernandez-Marquez J.L. and Serugendo G.D.: *A decentralized approach for detecting dynamically changing diffuse event sources in noisy wsn environments*. Applied Artificial Intelligence, 26:376–397, 2012.
- [41] Kermajani H. and Gomez C.: *On the network convergence process in rpl over ieee 802.15.4 multihop networks: Improvement and trade-offs*. Sensors, 14:11993–12022, 2014.
- [42] Abid Ali M. Awais A. Sohail J. Hilal J., Anand P. and Mucheoul K.: *Dependability and reliability analysis of intra cluster routing technique*. Peer-to-Peer Networking and Applications, 8:838–850, 2015.
- [43] *IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks – wireless lan medium access control (mac) and physical layer (phy) specifications*. Standard, Institute of Electrical and Electronics Engineers Computer Society, New York, US, March 2012.
- [44] *IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks – wireless medium access control and physical layer specifications for wireless personal area networks (wpans)*. Standard, Institute of Electrical and Electronics Engineers Computer Society, New York, US, June 2005.
- [45] *IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks – wireless medium access control and physical layer specifications for high rate wireless personal area networks (wpans)*. Standard, Institute of Electrical and Electronics Engineers Computer Society, New York, US, September 2009.
- [46] *IEEE standard for local and metropolitan area networks – low-rate wireless personal area networks (lr-wpans)*. Standard, Institute of Electrical and Electronics Engineers Computer Society, New York, US, September 2011.

- [47] *ANSI/ISA-100.11a-2011 wireless systems for industrial automation: Process control and related applications.* Standard, International Society of Automation, Durham, US, 2011.
- [48] *Information technology – radio frequency identification for item management – parameters for active air interface communications at 433 mhz.* Standard, International Organization for Standardization, Geneva, CH, August 2004.
- [49] Choi J. and Lee C.: *Energy consumption and lifetime analysis in clustered multi-hop wireless sensor networks using the probabilistic cluster-head selection method.* EURASIP Journal on Wireless Communications and Networking, 2011.
- [50] Chang J.H. and Tassiulas L.: *Maximum lifetime routing in wireless sensor networks.* IEEE/ACM Transactions on Networking, 12(4), August 2014.
- [51] Baoju Z. Jiasong M., Wei W. and Wei S.: *An adaptive routing optimization and energy-balancing algorithm in zigbee hierarchical networks.* Journal on Wireless Communications and Networking, 43:11, 2014.
- [52] Hu Y. Johnson D. and Maltz D.: *The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4.* Rfc 4728, RFC Editor, February 2007. <http://www.rfc-editor.org/rfc/rfc4728.txt>, <http://www.rfc-editor.org/rfc/rfc4728.txt>.
- [53] Hui J.W. and Culler D.E.: *IPv6 in low-power wireless networks.* In *Proceedings of the IEEE*, volume 98 Issue: 11, pages 1865–1878, 2010.
- [54] Bagchi S. Khalil I. and Shroff N.: *Analysis and evaluation of secos, a protocol for energy efficient and secure communication.* Ad Hoc Networks, 5:360–391, 2007.
- [55] Calderoni L., Maio D., and Rovis S.: *Deploying a network of smart cameras for traffic monitoring on a "city kernel".* Expert Systems with Applications, 41:502–507, February 2014.
- [56] Kim K.T. Lee H.S. and Youn H.Y.: *A new cluster head selection scheme for long lifetime of wireless sensor networks.* Computational Science and its Applications - ICCSA 2006, 3983:519?528, 2006.
- [57] Polastre J. Szewczyk R. Whitehouse K. Woo A. Gay D. Hill J. Welsh M. Brewer E. Levis P., Madden S. and Culler D.: *Tinyos: An operating system for sensor networks.* http://www.dbis.ethz.ch/education/ss2007/tatbul/hotdms/papers/tinyos_chapter.pdf.
- [58] Tucker R. Lécué F., Tallevi-Diotallevi H.J. and Bicer V.: *Smart traffic analytics in the semantic web with star-city: Scenarios, system and lessons learned in dublin city.* Web Semantics: Science, Services and Agents on the World Wide Web, 27-28:26–23, 2014.

- [59] Saravanan M. and Madheswaran M.: *A hybrid optimized weighted minimum spanning tree for the shortest intrapath selection in wireless sensor network.* Mathematical Problems in Engineering, 2014:8, 2014.
- [60] Matin M.A. and Islam M.M.: *Wireless Sensor Networks - Technology and Protocols*, chapter 1, pages 3–24. InTech, 2012, ISBN 978-953-51-0735-4.
- [61] Cerqueira E. Loureiro A.A.F. Neto A. Machado K., Rosario D. and Neumann de Souza J.: *A routing protocol based on energy and link quality for internet of things applications.* Sensors, 13:1942–1964, 2013.
- [62] Sreenan C.J. Mercadal E., Vidueira C. and Borrell J.: *Improving the dynamism of mobile agent applications in wireless sensor networks through separate itineraries.* Computer communications, 36:1011–1023, 2013.
- [63] Gopalakrishnan S. Minhas M.R. and Leung V.C.M.: *Fuzzy algorithms for maximum lifetime routing in wireless sensor networks.* In *Proceedings of Global telecommunications conference (GLOBECOM)*, December 2008. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4697801>.
- [64] Mathew J. Mishra V. and Peradhan D.K.: *Fault-tolerant de-bruijn graph based multipurpose architecture and routing protocol for wireless sensor networks.* International Journal of Sensor Networks, 10:160–175, 2011.
- [65] Lugaric L. Morvaj B. and Krajkar S.: *Smart city - platform for emergent phenomena power system testbed simulator.* In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, pages 1–7, 2010.
- [66] Corson M.S. and Park V.D.: *Temporally ordered routing algorithm (tora) version 1: Functional specification* (1997). in *internet-draft*. <https://www.ietf.org/proceedings/52/I-D/draft-ietf-manet-tora-spec-04.txt>.
- [67] Mittal N. and Singh U.: *Distance-based residual energy-efficient stable election protocol for wsns.* Arabian Journal for Science and Engineering, 40:1637–1646, 2015.
- [68] Islam O. and Hussain S.: *An intelligent multi-hop routing for wireless sensor networks.* In *Proceedings of WI-IAT Workshops Web Intelligence and Intelligent Agent Technology Workshops*, December 2006. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4053244.
- [69] Belding Royer E. Perkins C. and Das S.: *Ad hoc on-demand distance vector (AODV) routing.* Rfc 3561, RFC Editor, July 2003. <http://www.rfc-editor.org/rfc/rfc3561.txt>, <http://www.rfc-editor.org/rfc/rfc3561.txt>.
- [70] HART Communication Protocol: <http://en.hartcomm.org/>. Last Access March, 2016.

- [71] Gloseskoetter P. Castillo Secilla J. M. Rebel G., Estevez F.J.: *Energy harvesting on human bodies*. In Roecker C. Holzinger A. and Ziefler M. (editors): *Smart Health, Open Problems and Future Challenges*, chapter 6, pages 125–159. Springer International, 2014.
- [72] Mangharam R. Rowe A. and Rajkumar R.: *RT-Link: A time-synchronized link protocol for energy-constrained multi-hop wireless networks*. In *Proceedings of the 3th Annual IEEE Communications Society Conference on Sensors, Mesh and Ad Hoc Communications and Networks*, 2006.
- [73] Kamouskos S. and Nass de Holanda T.: *Simulation of a smart grid city with software agents*. In *Third UKSim European Symposium on Computer Modeling and Simulation, 2009*, pages 424–429, 2009.
- [74] Preet Kaurab S. and Sharmacd M.: *Radially optimized zone-divided energy-aware wireless sensor networks (wsn) protocol using ba (bat algorithm)*. IETE Journal of Research, 61:170–179, 2015.
- [75] Soro S. and Heinzelman W.: *Prolonging the lifetime of wireless sensor networks via unequal clustering*. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2005*, ISBN 0-7695-2312-9.
- [76] Ruiz P.M. Sanchez J.A. and Stojmenovic I.: *Energy-efficient geographic multicast routing for sensor and actuator networks*. Computer Communications, 30:2519–2531, 2007.
- [77] Anuar N.B. Shamshirband S., Amini A. and Klah M.L.M.: *D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks*. Measurement, 55:212–216, 2014.
- [78] Shu Ching W. Shun-Sheng W. and Kuo Qin Y.: *Reaching trusted byzantine agreement in a cluster-based wireless sensor network*. Wireless Personal Communications, 78:1079–1094, 2014.
- [79] Alam S.M.N. and Haas Z.J.: *Coverage and connectivity in three-dimensional networks with random node deployment*. Ad Hoc Networks, 34:157–169, 2015.
- [80] Vardharajulu K. N. Sundareswaran P. and Rajesh R. S.: *Dech: Equally distributed cluster heads technique for clustering protocols in wsns*. Wireless Personal Communications, 84:137–151, 2015.
- [81] Clausen T. and Jacquet P.: *Optimized link state routing protocol (olsr)*. Rfc 3626, RFC Editor, October 2003. <http://www.rfc-editor.org/rfc/rfc3626.txt>, <http://www.rfc-editor.org/rfc/rfc3626.txt>.
- [82] Andrew S. Tanenbaum: *Computer Networks*. Prentice-Hall, Vrije University, Amsterdam, The Netherlands, 4th edition, 2002.
- [83] Guo M.Y. Guo S. Tang F.L., Tang C. and Yu S.: *Service-oriented wireless sensor networks and an energy-aware mesh routing algorithm*. Ad Hoc & Sensors Wireless Networks, 15:21–46, 2012.

- [84] A. Varga: *The OMNeT++ discrete event simulation systems.* In *Proceedings of the European Simulation Multiconference*, June 2001. <http://dl.acm.org/citation.cfm?id=1416290>.
- [85] Ze Ming S. Wei-Xian X., Qi-Ye Z. and Feng Z.: *A clustering routing protocol for wsn based on type-2 fuzzy logic and ant colony optimization.* Wireless Personal Communications, 84:1165–1196, 2015.
- [86] Wikipedia, 2016. https://es.wikipedia.org/wiki/Red_de_sensores.
- [87] Brandt A. Hui J. Kelsey R. Levis P. Pister K. Struik R. Vasseur J.P. Winter T., Thubert P. and Alexander R.: *RPL: IPv6 routing protocol for low-power and lossy networks.* Rfc 6550, RFC Editor, March 2012. <http://www.rfc-editor.org/rfc/rfc6550.txt>, <http://www.rfc-editor.org/rfc/rfc6550.txt>.
- [88] Llu X.: *A survey on clustering routing protocols in wireless sensor networks.* Sensors, 12:11113–11153, 2012.
- [89] Chen G.H. Ye M., Li C.F. and Wu J.: *An energy efficient clustering scheme in wireless sensor networks.* Ad Hoc Sensor Wireless Networks, 3:99–119, 2007.

DECLARACIÓN

Declaro que he desarrollado esta Tesis Doctoral, bajo la supervisión de los directores de Tesis y, por tanto, asumo la autoría de todo lo descrito en el presente documento.

También declaro que el trabajo realizado es inédito (salvo por las publicaciones científicas obtenidas y mencionadas en el apartado de PRODUCCIÓN CIENTÍFICA del Capítulo 9) y no es plagio ni total ni parcial de ninguna otra investigación realizada por otras personas.

Afirmo que todos los datos expuestos en esta investigación no han sido falseados y que cualquier error que pudiera existir en el documento, no ha sido introducido conscientemente.

Granada, Abril 2016

Francisco José Estévez Ortiz

