

# **Redes inalámbricas de sensores: una nueva arquitectura eficiente y robusta basada en jerarquía dinámica de grupos**

Juan Vicente Capella Hernández

EDITORIAL  
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA



DEPARTAMENTO DE  
INFORMÁTICA DE SISTEMAS Y COMPUTADORES

TESIS DOCTORAL

*Redes inalámbricas de sensores:  
Una nueva arquitectura eficiente y robusta  
basada en jerarquía dinámica de grupos*

AUTOR:

Juan Vicente Capella Hernández

DIRECTOR:

Dr. Alberto Bonastre Pina

Valencia, abril de 2010



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



*Esta editorial es miembro de la UNE, lo que garantiza la difusión y comercialización de sus publicaciones a nivel nacional e internacional.*

© Juan Vicente Capella Hernández, 2011

Primera edición, 2011

© de la presente edición:  
Editorial Universitat Politècnica de València  
[www.editorial.upv.es](http://www.editorial.upv.es)

ISBN: 978-84-695-1042-1

Ref. editorial: 5513

Queda prohibida la reproducción, distribución, comercialización, transformación, y en general, cualquier otra forma de explotación, por cualquier procedimiento, de todo o parte de los contenidos de esta obra sin autorización expresa y por escrito de sus autores.

*A los que quiero.*

*“La ciencia más útil  
es aquella cuyo fruto es el  
más comunicable”*

Leonardo Da Vinci

# Resumen

Las Redes Inalámbricas de Sensores (RIS) gozan de un gran auge hoy en día, habiendo sido identificadas como una de las tecnologías más prometedoras por diversos analistas tecnológicos y revistas especializadas, debido a que dan respuesta a las exigencias actuales referentes al establecimiento de redes que cubran necesidades de comunicación de forma flexible -en tiempo y espacio- y autónoma -autoconfiguración e independencia de una estructura fija-. La posibilidad de implementar dispositivos de bajo coste y elevada duración capaces de obtener información del entorno y reenviarla de forma inalámbrica a un centro de coordinación ofrece posibilidades inimaginables en multitud de aplicaciones.

En la mayoría de las aplicaciones se pretende que los nodos no requieran mantenimiento, explotando el concepto de nodos de “usar y tirar”, puesto que una vez desplegados no son recuperables. En este entorno, los mayores desafíos se encuentran en minimizar dos factores fundamentales: coste y consumo, maximizando el tiempo de servicio.

En los últimos años se han propuesto numerosos protocolos para RIS. Sin embargo, la mayoría de estas propuestas, o bien realizan asunciones poco realistas, o bien resultan poco escalables, lo cual se traduce en aproximaciones poco prácticas.

Por otro lado, hasta la fecha se han obviado características deseables como tolerancia a fallos, seguridad y acotación de tiempos en las comunicaciones, generalmente no consideradas en la inmensa mayoría de los protocolos existentes, y cuando lo son, se obtienen sacrificando la eficiencia energética.

En la presente tesis se ha propuesto una novedosa solución integral orientada principalmente a reducir el consumo de energía. La nueva arquitectura denominada EDETA (*Energy-efficient aDaptive hiErarchical and robuST Architecture*) es además escalable, apropiada tanto para RIS homogéneas como heterogéneas –cada vez con mayor auge–, auto-configurable, soporta de forma transparente múltiples sumideros, y proporciona características como tolerancia a fallos y tiempos acotados, sin degradar las prestaciones de la red.

La arquitectura propuesta está basada en una jerarquía de dos niveles, el nivel inferior basado en clústeres, gobernado por un protocolo interno del clúster denominado *Intra-Cluster-Communication* y el nivel superior formado por un árbol dinámico de nodos líderes de clúster que ejecutan un protocolo entre clústeres denominado *Inter-Cluster-Routing*.

La evaluación de los mecanismos se ha realizado mediante una doble vía: implementación real y simulación, demostrándose que los mecanismos propuestos consiguen aumentar considerablemente el tiempo de vida de la red, a la vez que proporcionan adicionalmente robustez en las comunicaciones, tolerancia a fallos y tiempos de respuesta acotados. Esta arquitectura ha sido aplicada con éxito a sistemas reales en diversos campos.

# Abstract

Wireless Sensor Networks (WSN) have been identified as one of the most promising technologies for various technology analysts and magazines, because they satisfy the current requirements concerning the establishment of networks that cover communication needs in a flexible (in space and time), autonomous (self-configuration), and being independent of a fixed structure. The ability to deploy low-cost devices with high operation time without the need of maintenance, which are able to obtain environment information and transmit it wirelessly to a coordination center offers unimaginable possibilities in many applications.

In most applications it is intended that the nodes do not require maintenance, exploiting the concept of throwaway nodes, because once deployed they are not recoverable. In this environment, major challenges are to minimize two main factors: cost and consumption, maximizing the time of service.

In recent years many protocols have been proposed for WSN. However, most of these proposals, or make unrealistic assumptions or are not very scalable, which translates into non-practical approaches.

Furthermore, desirable characteristics, such as fault tolerance, safety and marking of time in communication, have generally been ignored, so they are not considered in the most of existing protocols, and when they are, the sacrifice of energy efficiency is required to be implemented.

In this thesis an innovative solution is proposed, aimed primarily at reducing energy consumption. The new architecture called EDETA (*Energy-efficient aDaptive hiErarchical and robusT Architecture*) is also scalable, suitable for both homogeneous WSN's and heterogeneous ones (increasingly employed), able to self-configure, which supports multiple sinks in a transparent manner, and provides features such as fault tolerance and bounded time responses, without degradation of the network performance.

The proposed architecture is based on a two-level hierarchy. The lower level is based on clusters, governed by an internal protocol called *Intra-Cluster Communication*, while the upper level consists of a dynamic tree node of cluster leaders, the so called *Inter-Cluster Routing* protocol.

The evaluation of the mechanisms has been performed through two ways: implementation and simulation, showing that the proposed mechanisms significantly increases the lifetime of the network, while providing additional robust communications, fault tolerance and time bounded response. This architecture has been successfully applied to real systems in different application fields.

# Resum

Les xarxes sense fil de sensors (XSFS) gaudeixen d'un gran auge avui dia, havent estat identificades com una de les tecnologies més prometedores per diversos analistes tecnològics i revistes especialitzades, degut al fet que donen resposta a les exigències actuals referents a l'establiment de xarxes que cobrisquen necessitats de comunicació de forma flexible -en temps i espai- i autònoma -autoconfiguració i independència d'una estructura fixa-. La possibilitat d'implementar dispositius de baix cost i elevada duració capaços d'obtenir informació de l'entorn i reenviar-la sense fils a un centre de coordinació ofereix possibilitats inimaginables en multitud d'aplicacions.

En la majoria de les aplicacions es pretén que els nodes no requerisquen manteniment, explotant el concepte de nodes d'usar i llençar, ja que una vegada desplegats no són recuperables. En aquest entorn, els majors desafiaments es troben a minimitzar dos factors fonamentals: cost i consum, maximitzant el temps de servei.

En els últims anys s'han proposat nombrosos protocols per a XSFS. No obstant això, la majoria d'aquestes propostes, o bé realitzen assumpcions poc realistes, o bé resulten poc escalables, la qual cosa es tradueix en aproximacions poc pràctiques.

D'altra banda, fins a la data s'han obviat característiques desitjables com tolerància a fallades, seguretat i acotament de temps en les comunicacions, generalment no considerades en la immensa majoria dels protocols existents, i quan ho són, s'obtenen sacrificant l'eficiència energètica.

En la present tesi s'ha proposat una nova solució integral orientada principalment a reduir el consum d'energia. La nova arquitectura denominada EDETA (*Energy-efficient aDaptative hiErarchical and robusT Architecture*) és a més escalable, apropiada tant per a XSFS homogènies com heterogènies -cada vegada amb major auge-, auto-configurable, suporta de forma transparent múltiples estacions base, i proporciona característiques com tolerància a fallades i temps acotats, sense degradar les prestacions de la xarxa.

L'arquitectura proposada està basada en una jerarquia de dos nivells, el nivell inferior basat en clústers, governat per un protocol intern del clúster denominat *Intra-Cluster-Communication* i el nivell superior format per un arbre dinàmic de nodes líders de clúster que executen un protocol entre clústers denominat *Inter-Cluster-Routing*.

L'avaluació dels mecanismes s'ha realitzat mitjançant una doble via: implementació real i simulació, demostrant-se que els mecanismes proposats aconsegueixen augmentar considerablement el temps de vida de la xarxa, alhora que proporcionen addicionalment robustesa en les comunicacions, tolerància a fallades i temps de resposta acotats. Aquesta arquitectura ha estat aplicada amb èxit a sistemes reals en diversos camps.

# Índice General

<b>Resumen</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Resum</b>	<b>ix</b>
<b>1 Introducción</b>	<b>27</b>
1.1 Redes inalámbricas de sensores.....	27
1.2 Línea de investigación.....	31
1.3 Objetivos y aportaciones de la tesis.....	33
1.4 Desarrollo de la tesis.....	35
<b>2 Estado del arte</b>	<b>37</b>
2.1 Clasificación de las estrategias de enrutamiento en WSN.....	37
2.2 Redes planas.....	40
2.2.1 <i>Sensor Protocols for Information via Negotiation (SPIN)</i> .....	40
2.2.2 Difusión Directa.....	42
2.2.3 <i>Rumor routing</i> .....	44
2.2.4 <i>Minimum Cost Forwarding Algorithm (MCFA)</i> .....	45
2.2.5 <i>Gradient-Based Routing (GBR)</i> .....	46
2.2.6 <i>Information-driven sensor querying (IDSQ)</i> y <i>CADR</i> .....	46
2.2.7 <i>COUGAR</i> .....	47
2.2.8 <i>Active Query forwarding in sensor networks (ACQUIRE)</i> .....	48
2.2.9 <i>Energy Aware Routing</i> .....	49
2.2.10 <i>Random Walks Routing Protocols</i> .....	50
2.2.11 Conclusiones sobre redes planas.....	52

2.3 Redes jerárquicas.....	53
2.3.1 <i>Low Energy Adaptive Clustering Hierarchy</i> (LEACH).....	53
2.3.2 <i>Low Energy Adaptive Clustering Hierarchy- Centralized</i> (LEACH-C).....	57
2.3.3 <i>Power-Efficient Gathering in Sensor Information Systems</i> (PEGASIS).....	60
2.3.4 <i>Threshold-sensitive Energy Efficient Protocols</i> (TEEN y APTEEN).....	63
2.3.5 <i>Small Minimum Energy Communication Network</i> (MECN).....	65
2.3.6 <i>Self Organizing Protocol</i> (SOP).....	66
2.3.7 <i>Sensor Aggregates Routing</i> .....	70
2.3.8 <i>Virtual Grid Architecture Routing</i> (VGA).....	70
2.3.9 <i>Hierarchical Power-aware Routing</i> (HPAR).....	71
2.3.10 <i>Two-Tier Data Dissemination</i> (TTDD).....	72
2.3.11 <i>Energy-efficient Distributed Dynamic Diffusion routing algorithm</i> (e3D).....	74
2.3.12 <i>Energy efficient clustering routing</i> (EECR).....	77
2.3.13 IMRAFRA.....	79
2.3.14 <i>Uniformly Distributed Adaptive Clustering Hierarchy routing protocol</i> .....	82
2.3.15 <i>Tree-Clustered Data Gathering Protocol</i> (TCDGP).....	87
2.3.16 <i>Energy-Efficient Distributed Unequal Clustering</i> (EEDUC).....	90
2.3.17 HTECRP.....	94
2.3.18 <i>Hybrid Energy-efficient Tree-based Optimized Routing Protocol</i> (HECTOR).....	97
2.3.19 <i>Energy-Balancing Unequal Clustering Protocol</i> (EB-UCP).....	102
2.3.20 <i>Grid-clustering Routing Protocol</i> (GROUP).....	105
2.3.21 <i>Base-station Controlled Dynamic Clustering Protocol</i> (BCDCP).....	108
2.3.22 <i>Dynamic Minimal Spanning Tree Routing Protocol</i> (DMSTRP).....	110
2.3.23 <i>Power Efficiency Routing Protocol</i> (PER).....	113
2.3.24 <i>Distributive Energy Efficient Adaptive Clustering</i> (DEEAC).....	116
2.4 Redes basadas en localización.....	119
2.4.1 <i>Geographic Adaptive Fidelity</i> (GAF).....	119

2.4.2 <i>Geographic and Energy Aware Routing (GEAR)</i> .....	123
2.5 Conclusiones.....	126
<b>3 Estrategias de optimización del consumo en WSN</b>	<b>129</b>
3.1 Enfoque para optimización del consumo de energía.....	129
3.2 Selección de las estrategias más adecuadas.....	131
3.3 Principios para los mecanismos propuestos y comparación.....	132
<b>4 Protocolo de encaminamiento EDETA</b>	<b>137</b>
4.1 Introducción.....	137
4.2 Arquitectura propuesta.....	138
4.2.1 Tipos de nodo.....	139
4.2.1.1 Nodo sumidero.....	139
4.2.1.2 Nodo sensor.....	140
4.2.2 Mensajes EDETA.....	141
4.2.2.1 Notificación rol NL (LEADER).....	141
4.2.2.2 Petición de unión al clúster (JOIN-REQ).....	142
4.2.2.3 Respuesta a la petición de unión al clúster (JOIN-REPLY).....	143
4.2.2.4 Necesidad de Nodo Líder (NEED_LEADER).....	143
4.2.2.5 Petición asociación con nodo padre árbol (ASSOC-REQ).....	144
4.2.2.6 Respuesta petición de asoc. con nodo padre (ASSOC-REPLY).....	144
4.2.2.7 Inf. planificación descendiente a nodo padre en árbol (DINFO).....	144
4.2.2.8 Asig. de planif. y dirección lógica desde nodo padre árbol (SICI).....	145
4.2.2.9 Notificación nodo sustituto (SN).....	145
4.2.2.10 Fallo nodo sumidero (SINK-FAILURE).....	146
4.2.2.11 POLL {intra-clúster   inter-clúster}.....	146
4.2.2.12 Transmisión de datos {intra-clúster   inter-clúster} (DATA).....	147
4.2.2.13 Reconocimiento {intra-clúster   inter-clúster} (ACK).....	147
4.3 Operación del protocolo.....	147

4.3.1 Fase de inicialización.....	149
4.3.1.1 Elección de rol.....	149
4.3.1.2 Notificación y formación del árbol.....	151
4.3.1.3 Establecimiento y planificación de los clústeres.....	153
4.3.1.4 Planificación del árbol.....	154
4.3.2 Fase de operación.....	155
4.3.2.1 Protocolo de comunicación <i>intra-cluster</i> .....	155
4.3.2.2 Protocolo de comunicación <i>inter-cluster dynamic tree</i> .....	158
4.3.2.3 Duración de los clústeres.....	161
4.4 Robustez del protocolo.....	161
4.4.1 Fallo de un nodo líder.....	162
4.4.2 Fallo de un nodo hoja.....	163
4.4.3 Fallo de nodo sumidero.....	163
4.4.4 Nodos huérfanos.....	164
4.5 Conclusiones.....	164
<b>5 Experimentación</b>	<b>167</b>
5.1 Metodología de evaluación.....	167
5.1.1 Proceso de simulación.....	167
5.1.2 El simulador NS-2.....	168
5.1.3 Modelo de energía.....	186
5.1.4 Metodología.....	198
5.2 Evaluación de prestaciones.....	199
5.2.1 Parámetros y escenarios.....	199
5.2.2 Impacto del tráfico de red.....	200
5.2.3 Impacto del número de nodos.....	200
5.2.4 Impacto del área de la red.....	201
5.2.5 Comparativa de EDETA, LEACH y ESTRELLA.....	202

5.3 Campos aplicación de las técnicas propuestas.....	205
5.3.1 Plataforma HW para la implementación de EDETA y mediciones reales.....	206
5.3.1.1 Cálculos estimación vida batería.....	207
5.3.2 Aplicación de la plataforma hardware.....	209
5.4 Estudio del impacto del número de sumideros y su ubicación.....	210
5.4.1 Sumideros en escenario grid – esférico.....	211
5.4.2 Sumideros en escenario en forma de camino.....	215
5.4.3 Conclusiones.....	217
5.5 Conclusiones finales.....	219
<b>6 Conclusiones, publicaciones y trabajo futuro</b>	<b>221</b>
6.1 Conclusiones.....	221
6.2 Publicaciones relacionadas con la tesis.....	224
6.3 Trabajo futuro.....	226
<b>Bibliografía</b>	<b>229</b>

## Índice de Figuras

2.1.	Componentes de un nodo sensor.....	38
2.2.	Taxonomía de los protocolos de enrutamiento en WSN.....	39
2.3.	Problema de la implosión (izquierda) y overlap (derecha).....	41
2.4.	Negociación en SPIN.....	41
2.5.	(a) Propagación del interés, (b) configuración inicial de los gradientes y (c) entrega de la información mediante el camino reforzado.....	43
2.6.	Operación protocolo RR.....	44
2.7.	CADR para N = 200 nodos. De izquierda a derecha: $\gamma = 1$ , $\gamma = 0.2$ , $\gamma = 0.0$ .....	47
2.8.	Operación protocolo ACQUIRE.....	48
2.9.	Energía consumida ( $\mu\text{J}$ ) para <i>Energy Aware</i> y Difusión Directa de izquierda a derecha respectivamente .....	50
2.10.	Malla cuadrada de tamaño N x N. Los paquetes se inyectan en el nodo fuente situado en [0,0] y deben viajar salto a salto hasta el nodo destino [N-1,N-1] .....	51
2.11.	Etapas de expansión y compresión (izquierda) y probabilidades de retransmisión (derecha) .....	52
2.12.	Mensajes involucrados en el proceso de asociación al clúster.....	54
2.13.	Distribución de los clústeres en la red. CHs (rojo) y los nodos hoja junto con sus regiones de acción .....	54
2.14.	Estructura del <i>frame</i> TDMA donde se aprecia un flujo de datos dividido en <i>frames</i> y estos <i>frames</i> divididos en slots de tiempo.....	55
2.15.	Asignación de ranuras a los nodos en la fase de operación.....	55
2.16.	Evolución temporal operacional de LEACH donde se aprecia como la ronda se divide en <i>frames</i> .....	56

2.17.	Energía total disipada (izquierda) y número de nodos que siguen vivos (derecha) en función del número de rondas para una red con $N = 100$ nodos, distribuidos de forma aleatoria .....	56
2.18.	(a) Cantidad total de información recibida en el sumidero en función del tiempo y (b) cantidad total de información recibida en el sumidero en función de la energía. Cada nodo comienza con 2 J de energía.....	58
2.19.	(a) Número de nodos vivos en función del tiempo y (b) número de nodos vivos en función de la cantidad de datos entregada al sumidero. Cada nodo comienza con 2 J de energía.....	59
2.20.	Encadenamiento en PEGASIS.....	61
2.21.	Esquema basado en cadenas de 3-niveles para una red de sensores con nodos sin CDMA .....	63
2.22.	<i>Clustering</i> jerárquico en TEEN.....	64
2.23.	Línea operacional para (a) TEEN y (b) APTEEN.....	65
2.24 .	Subred inicial computada por MECN y SMECN.....	66
2.25.	Sistema auto-organizativo con sensores routers, nodos especializados y nodos sumidero .....	68
2.26.	Dos fuentes de señal colocadas en un mismo punto (izquierda) y celdas de Voronoi para $N = 200$ nodos aleatoriamente distribuidos .....	70
2.27.	Distribución regular con la localización de LAs. Un subconjunto de éstos, los MAs, son seleccionados de manera subóptima .....	71
2.28.	Disposición nodos en TTDD. Nodo fuente B. Sumidero S.....	72
2.29.	Consulta <i>Two-Tier</i> y posterior retransmisión entre la fuente A y los sumideros S1 y S2 .....	73
2.30.	Tiempo de vida de los nodos para Comunicación Directa, Difusión Básica y Clustering Aleatorio respectivamente .....	75
2.31.	Tiempo de vida de los nodos aplicando Difusión e3D.....	75
2.32.	Tiempo de vida de los nodos para <i>clustering</i> ideal.....	76
2.33 .	Composición de una ronda.....	78
2.34.	Número de nodos vivos, mensajes recibidos por el sumidero y disipación energética media .....	78
2.35.	Encaminamiento inter-clúster en IMRAFRA.....	81
2.36.	Limitación del ámbito de retransmisión empleando ángulo de restricción.....	81
2.37.	Transmisión de los paquetes intra-clúster y restricciones indicadas por la desigualdad .....	82

2.38.	Número de nodos vivos para el área de (a) 500x500m y (b) 1000x1000m.....	85
2.39.	Energía total disipada para el área de 1000x1000m.....	85
2.40.	Consumo energético en la fase de configuración para diferentes tamaños de red .....	86
2.41.	Diagrama de flujo de TCDGP.....	88
2.42.	Agregación en TCDGP.....	89
2.43.	Comparación en número de rondas con Direct Diffusion, LEACH y PEGASIS mediante FND .....	90
2.44.	Visión general del mecanismo de comunicación en EEUC.....	91
2.45.	Fase de clusterización en EEDUC.....	92
2.46.	Energía consumida por los CHs.....	93
2.47 .	Energía total restante en la red.....	94
2.48.	Número de paquetes perdidos en función del número de eventos en el protocolo .....	97
2.49.	Etiquetado en HECTOR.....	99
2.50.	Topología homogénea.....	100
2.51.	Evaluación prot. HECTOR: Vcost=éxito (a) 3 <i>landmarks</i> y (b) 5 <i>landmarks</i> .....	101
2.52.	Evaluación prot. HECTOR: Vcost=fallo (a) 3 <i>landmarks</i> y (b) 5 <i>landmarks</i> .....	101
2.53.	El área circular consistente en coronas.....	103
2.54.	Algoritmo de transmisión inter-clúster.....	104
2.55.	Comparación de rendimiento para (a) $\alpha = 0.1$ , (b) $\alpha = 0.3$ , (c) $\alpha = 0.7$ y (d) $\alpha = 1.0$ .....	105
2.56.	Ejemplo de clúster en malla.....	106
2.57.	<i>Forwarding</i> de datos durante la construcción de la malla.....	107
2.58.	Consumo de energía y retardo medio entre GROUP y LEACH.....	107
2.59.	Proceso del algoritmo en el protocolo para las fases de Configuración y Comunicación.....	109
2.60.	Disipación energética media en comparación con otras estrategias basadas en clustering .....	110
2.61.	Cambio en la topología básica de (a) club en LEACH y BCDCP a (b) MST en DMSTRP .....	110
2.62 .	Tiempo de vida de la red para distintos tamaños en la red de sensores.....	112

2.63.	El CH y nodos sensores en el clúster. Las líneas punteadas denotan los anillos de comunicación .....	114
2.64.	Congestión alrededor de un agujero.....	114
2.65.	Método circular para solventar el problema de la congestión alrededor de los agujeros en la red .....	115
2.66 .	Normalización del consumo de energía total de PER y DD.....	115
2.67 .	Energía total disipada en función del tiempo para LEACH y DEEAC.....	117
2.68.	(a) Fracción de CHs en puntos calientes y (b) fracción de la energía disipada por los nodos que pertenecen a puntos calientes .....	118
2.69.	Número de nodos vivos en función del tiempo.....	118
2.70 .	Malla virtual y representantes en GAF.....	119
2.71.	Transiciones entre estados en GAF.....	120
2.72.	Comparación en cuanto al tiempo de vida de la red. GAF vs. AODV a velocidades bajas de los nodos y varios tiempos de pausa .....	122
2.73.	Aprendizaje de las rutas alrededor de los agujeros.....	124
2.74.	<i>Forwarding</i> geográfico recursivo.....	124
4.1.	Arquitectura propuesta.....	139
4.2.	Estructura nodo sumidero.....	140
4.3.	Estructura nodo sensor.....	140
4.4.	Operación protocolo EDETA.....	149
4.5.	Diagrama de estados NL: procedimiento de formación árbol.....	152
4.6.	Diagrama de estados nodo normal.....	157
4.7.	Diagrama de estados NL: inter-clúster routing.....	158
4.8.	Diagrama de estados general Nodo Líder.....	159
4.9.	Diagrama de estados NL inter-clúster routing: Reemplazar padre.....	161
5.1.	Jerarquía de clases parcial de NS-2.....	169
5.2.	Clase <i>Classifier</i> .....	170
5.3.	Diagrama de colaboración de la clase <i>Classifier</i> .....	171
5.4.	Diagrama de colaboración de la clase <i>Address</i> .....	171
5.5.	Diagrama de colaboración de la clase <i>MultiCast</i> .....	172
5.6.	Diagrama de colaboración de la clase <i>MultiPath</i> .....	172

5.7.	Diagrama de colaboración de la clase <i>Hash</i> .....	173
5.8.	Diagrama de colaboración de la clase <i>Connector</i> .....	174
5.9.	Diagrama de colaboración de la clase <i>LinkDelay</i> .....	174
5.10.	Diagrama de colaboración de la clase <i>Mac</i> .....	176
5.11.	Diagrama de colaboración de la clase <i>Queue</i> .....	177
5.12.	Diagrama de colaboración de la clase <i>Trace</i> .....	177
5.13.	Estructura de un nodo <i>unicast</i> .....	178
5.14.	Estructura de un nodo <i>multicast</i> .....	178
5.15.	Diagrama de colaboración de la clase LL (enlace de datos).....	179
5.16.	Ejemplo de flujo.....	180
5.17.	Diagrama de colaboración de la clase <i>MobileNode</i> .....	180
5.18.	Diagrama de bloques <i>MobileNode</i> bajo la extensión <i>wireless</i> .....	181
5.19.	Representación gráfica <i>Free Space Model</i> .....	184
5.20.	Representación gráfica <i>Shadowing Model</i> .....	185
5.21.	DTE incluyendo consumos entre transiciones y tiempo de transición (2v). Chip radio CC2420.....	188
5.22.	Porcentaje de cambio en la productividad (a) y bytes-por-Julio disipado (b) cuando se permanece en Shutdown en contrapartida de no hacerlo .....	190
5.23.	Línea de tiempo operacional de EDETA para NL, diferenciando sus operaciones y fases de la comunicación .....	193
5.24.	DTE parcial nodo líder.....	194
5.25.	Línea de tiempo operacional para el NL.....	194
5.26 .	Línea de tiempo operacional para el NL desglosado en función de la fase.....	194
5.27.	Línea de tiempo operacional de EDETA para el nodo normal, diferenciando sus operaciones y fases de la comunicación .....	195
5.28 .	Línea de tiempo operacional para el nodo normal.....	196
5.29.	DTE parcial para los nodos normales o nodos hoja.....	196
5.30.	DTE parcial del inter-cluster routing de los nodos líderes.....	197
5.31.	Estudio comparativo del consumo de energía medio de los protocolos de encaminamiento en función de la tasa de envío.....	200

5.32.	Estudio comparativo del consumo de energía medio de los diferentes protocolos de encaminamiento en función del número de nodos .....	201
5.33.	Estudio comparativo del consumo de energía medio de los diferentes protocolos de encaminamiento en función del diámetro de la red: 100x100m, 200x200m, 500x500m, 1000x100m.....	202
5.34 .	Nodos vivos respecto del tiempo. (a) 100 nodos, 200x200 m. (b) 200 nodos, 500x500 m.....	202
5.35.	Consumo de energía en los primeros 1100 segundos. (a) 100 nodos, 200x200 m. (b) 200 nodos, 500x500 m.....	203
5.36.	Consumo de energía en los primeros 1200 segundos. 100 nodos en áreas desde 100 a 260 metros de diámetro .....	203
5.37.	% Paquetes perdidos en LEACH y EDETA.....	204
5.38 .	Diversas plataformas evaluadas para la implementación de EDETA.....	205
5.39.	(a) Vista componentes PCB del nodo y con la batería instalada. (b) Nodo completo insertado en la madera.....	209
5.40.	Vista interna del nodo sumidero.....	210
5.41.	Disposición de los sumideros en las pruebas para 1 SINK en escenario grid-esférico.....	211
5.42.	Impacto de la posición del SINK. 1 SINK escenario grid-esférico.....	212
5.43.	Disposición de los sumideros en las pruebas para 2 SINK en escenario grid-esférico .....	212
5.44.	Impacto de la posición del SINK. 2 SINKs escenario grid-esférico.....	213
5.45.	Disposición de los sumideros en las pruebas para 3 SINK en escenario grid-esférico.....	213
5.46.	Impacto de la posición del SINK. 3 SINKs escenario grid-esférico.....	214
5.47.	Disposición de los sumideros en las pruebas para 4 SINK en escenario grid-esférico .....	214
5.48 .	Impacto de la posición del SINK. 4 SINKs escenario grid-esférico.....	215
5.49.	Disposición de los sumideros en las pruebas para 1 SINK. Escenario forma de camino .....	215
5.50 .	Impacto de la posición del SINK. 1 SINK escenario forma de camino.....	215
5.51.	Disposición de los sumideros en las pruebas para 2 SINKs. Escenario forma de camino .....	216
5.52 .	Impacto de la posición del SINK. 2 SINKs escenario forma de camino.....	216

5.53.	Disposición de los sumideros en las pruebas para 3 SINK. Escenario forma de camino .....	216
5.54 .	Impacto de la posición del SINK. 3 SINK escenario forma de camino.....	217
5.55.	Disposición de los sumideros en las pruebas para 4 SINKs. Escenario forma de camino .....	217
5.56.	Impacto del número de sumideros sobre el consumo. Escenario grid-esférico .....	218
5.57 .	Impacto del número de sumideros sobre la latencia. Escenario grid-esférico .....	218
5.58.	Impacto del número de sumideros sobre el consumo. Escenario forma de camino .....	218
5.59.	Impacto del número de sumideros sobre la latencia. Escenario forma de camino .....	219

## Índice de Tablas

2.1.	Parámetros de los experimentos.....	89
2.2.	Encaminamiento jerárquico vs. plano.....	126
2.3.	Clasificación y comparación de protocolos de enrutamiento en WSN.....	127
4.1.	Mensajes Protocolo EDETA.....	141
5.1.	Consumo corriente medido y declarado en hoja de especificaciones VDD=3.3V.....	207
5.2.	Requisitos de energía para cada módulo en la configuración en estrella. VDD=3.3V.....	207
5.3.	Requisitos de energía para cada módulo para EDETA. VDD=3.3V.....	208
5.4.	Duración estimada baterías para la configuración en estrella.....	208
5.5.	Duración estimada baterías para la implementación basada en EDETA.....	209

# Capítulo 1

## Introducción

### 1.1 Redes inalámbricas de sensores

En las últimas décadas hemos asistido a un explosivo crecimiento de las redes de computadores y en concreto de las comunicaciones inalámbricas, propiciado por los continuos avances tecnológicos. Así, han aparecido circuitos electrónicos cada vez más pequeños, potentes y de menor coste, permitiendo también en esta línea, importantes avances en el campo de los transductores. Todo ello permite el desarrollo de nuevos dispositivos para la detección y medida de cualquier magnitud de forma sencilla y con gran precisión, siendo estos dispositivos de pequeño tamaño y bajo coste.

Estos factores han permitido el despegue del campo de investigación de las Redes Inalámbricas de Sensores (RIS), conocidas en inglés como *Wireless Sensor Networks* (WSN), que han sido identificadas como una de las tecnologías más prometedoras por diversos analistas tecnológicos y revistas especializadas, entre las que pueden citarse el observatorio tecnológico del MIT [MIT03] y la revista Dinero [DINERO04]. La posibilidad de implementar dispositivos de bajo coste y elevada duración sin mantenimiento capaces de obtener información del entorno y reenviarla de forma inalámbrica a un centro de coordinación ofrece posibilidades inimaginables en multitud de aplicaciones. Las RIS están siendo aplicadas con éxito a sistemas de automoción, aplicaciones industriales, aviónica, entornos inteligentes, identificación de productos, domótica y seguridad, control de consumo energético, estudio de

invernaderos, monitorización del medio ambiente, y un sinnúmero de nuevas aplicaciones [BOUKER09].

La principal innovación consiste en sustituir sensores de elevada complejidad – y por tanto caros y limitados en número – y su infraestructura de comunicaciones asociada por un conjunto mucho mayor de dispositivos sensores de más sencillos – que resultan considerablemente más baratos – que actúan como sensores, obteniendo magnitudes físicas del entorno, pero que además dan soporte a las comunicaciones de otros nodos como elementos de infraestructura.

La característica principal de las RIS es su capacidad de organización automática. Así, es posible realizar despliegues de sensores en lugares de difícil acceso, o incluso en el interior de zonas afectadas por catástrofes, mediante medios aéreos. Gracias a esta característica es posible mantener la red en funcionamiento incluso cuando algunos nodos se encuentran fuera de servicio, debido a fallos, agotamiento baterías, modo bajo consumo, etc. En estos casos la red es capaz de reorganizarse y continuar funcionando.

Las RIS se basan en dotar a cada nodo de enlaces de radio de baja potencia, de tal modo que el área de cobertura es relativamente pequeña. De esta forma se consigue economizar de forma significativa el consumo de potencia, mientras que, en cambio, es necesario proporcionar mecanismos de encaminamiento con múltiples saltos, que permitan la comunicación con los nodos más alejados. Otra ventaja es la reutilización de frecuencias, ya que dos nodos con áreas de cobertura disjuntas podrán emplear la misma banda de transmisión. Si la densidad de nodos es lo suficientemente grande, este mecanismo permite establecer múltiples rutas para cada destino posible, permitiendo la implementación de técnicas de tolerancia a fallos contemplando rutas redundantes –y de gestión global de energía– empleando rutas alternativas para balancear el consumo entre nodos.

Otra ventaja significativa consiste en la posibilidad de plantear mecanismos de cooperación entre nodos, bien mediante distribución de funciones por ejemplo realizando cada nodo una labor complementaria a sus vecinos o bien mediante procesos redundantes donde cada nodo supervisa el comportamiento del vecino.

Sin embargo, estas capacidades vienen limitadas por dos factores fundamentales: coste y consumo. Así, en la mayoría de las aplicaciones se pretende que los nodos no requieran mantenimiento, explotando el concepto de nodos de “usar y tirar”, puesto que una vez desplegados no son recuperables. Para ello se requiere que su coste sea mínimo, y que realicen un aprovechamiento óptimo de la energía, maximizando el tiempo de servicio.

El diseño de una red de sensores está motivado o influido por uno o más de los siguientes requisitos:

- Despliegue masivo y aleatorio: La mayoría de las RIS contienen un número muy grande de sensores (del orden de cientos o millares, como poco), que pueden diseminarse de forma aleatoria en las áreas de interés o ser lanzadas desde un avión sobre terrenos inaccesibles o peligrosos. El sistema debe realizar una configuración automática antes de llevar a cabo las tareas de sensorización requeridas.

- Redundancia de la información: la alta densidad de nodos sensores hace que los datos obtenidos en uno de ellos sean redundantes con respecto a otros nodos de su entorno.
- Limitación de recursos. El diseño e implementación de RIS debe tener en cuenta cuatro limitaciones de recursos: energía, potencia de cálculo, memoria y ancho de banda. Además de estar generalmente limitados en tamaño, los sensores dependerán de sus baterías y de la energía que puedan extraer del entorno para su funcionamiento. Por ello (y por economizar costes), los nodos habitualmente están limitados en memoria, potencia de cálculo y ancho de banda.
- Topología y entornos dinámicos. Las condiciones en las cuales se despliega una red de sensores no es fija, sino que pueden existir movimiento de sensores e incluso desaparición u adición de nodos. Las RIS deben ser capaces de reconfigurarse de forma autónoma frente a estos avatares.
- Medio de transmisión poco fiable: La utilización de comunicaciones inalámbricas presenta una tasa de errores considerablemente mayor que las comunicaciones cableadas.
- Diversidad de aplicaciones: debido a las propiedades de las RIS, donde en cada una de ellas pueden primar más unas características que otras (robustez, consumo, QoS, alcance, etc.)
- Seguridad y privacidad: estos factores son especialmente importantes en aplicaciones militares o de vigilancia. Por tanto, los ataques de denegación de servicio, intrusión o manipulación de datos en estas aplicaciones deben estar previstos.

Uno de los aspectos que más impacto causa en las prestaciones de las redes inalámbricas de sensores es el conjunto de protocolos empleados en cada uno de sus niveles. Por ello, este es un frente de investigación aún abierto: a pesar de la existencia de protocolos estándar (IEEE 802.15.4) no dejan de surgir otras propuestas más eficientes para muchas aplicaciones en las que se requiere un amplio periodo de funcionamiento con fuentes de energía muy limitadas (bajo coste de los nodos), siendo por tanto la eficiencia energética uno de los principales objetivos de esta tesis. Sin embargo, este trabajo no descuida otras características cada vez más demandadas en nuevas aplicaciones de las RIS como son escalabilidad, flexibilidad, tolerancia a fallos, seguridad y acotación de tiempos en las comunicaciones [CAPELLA05].

En cuanto a tecnologías para la implementación de RIS, en este momento se contemplan las siguientes:

- WiFi: El estándar de comunicaciones IEEE 802.11 [WIFI09] está ampliamente extendido en comunicación entre computadores. Es una tecnología de elevado ancho de banda.
- BlueTooth: El estándar Bluetooth [BLUET09] posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia. Es un estándar ampliamente empleado en telefonía móvil.

- 802.15.4 /Zigbee: 802.15.4 [IEEE15.4-06] ha sido diseñado específicamente para el desarrollo de redes de sensores inalámbricas. Zigbee [ZIGBEE07] está orientado a proporcionar dispositivos de un bajo coste, así como reducido consumo de energía y gran simplicidad hardware y software.
- Tecnologías inalámbricas sub-Ghz: se trata de tecnologías también diseñadas específicamente para redes de sensores, que operan en bandas inferiores al GHz, típicamente sobre los cuatrocientos y ochocientos megahercios, y cuyo primordial objetivo es el bajo consumo tanto en transmisión como en recepción. La gran ventaja de estas tecnologías frente a las ubicadas en los 2.4Ghz es que tienen un mayor radio de alcance, les afecta menos las inclemencias meteorológicas, y tienen generalmente menos interferencias de otros dispositivos inalámbricos en dichas bandas que en la de 2.4 Ghz.

En la presente tesis se han evaluado las anteriores tecnologías, decantándose la balanza hacia tecnologías sub-Ghz y 802.15.4, principalmente por motivos de consumo de energía, tal como se recoge en el capítulo 2 (Estado del arte).

En cuanto a campos de aplicación, actualmente se contempla la utilización de las tecnologías basadas en las RIS en:

- Aplicaciones militares: Sistemas C4ISR (Command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting), tareas de reconocimiento y alerta temprana.
- Aplicaciones medioambientales: Monitorización de hábitat, detección de inundaciones, incendios, etc.
- Edificios: Sistemas de climatización más eficientes, localización de personas u objetos, aplicaciones domóticas y de seguridad, etc.
- Sistemas de emergencia.
- Ciencias de la salud: Seguimiento y monitorización de las constantes biomédicas.
- Hogar: Alarmas técnicas y de seguridad (inundación, gas, incendio, intrusión), aplicaciones domóticas, etc.
- Exploración científica: Entornos lugares con inaccesibles o peligrosos (simas oceánicas, volcanes, etc.).

Multitud de investigadores centran sus esfuerzos en este campo tan prometedor. Como ejemplo se pueden citar algunos de los proyectos de investigación más representativos llevados a cabo en la Unión Europea:

- Awake (UE) [AWAKE ]: Es un sistema para la vigilancia de conductores de vehículos en el tráfico: cansancio, falta de atención, etc.

- WiSeNts (UE) [WISENTS]: Cooperación de sistemas empotrados de redes de sensores inalámbricas.
- EYES (UE) [EYES]: Redes de sensores autoconfigurables, colaborativas y eficientes energéticamente.
- Hogthrob (Dinamarca) [HOGTHRO]: Redes de Sensores para la monitorización del estado de la nieve.
- CSOnet Project (Francia) [CSONET]: Redes inalámbricas de sensores para monitorización y control de cloacas en área metropolitana.
- RUNES (UE) [RUNES]: Redes de Sensores autoconfigurables.
- SECURE TIME SYNCHRONIZATION IN SENSOR NETWORKS (Dinamarca) [SECURET]: Proyecto de investigación centrado en la sincronización de redes de sensores.
- SPOT (Dinamarca) [SPOT]: Localización de nodos en redes de sensores inalámbricas.
- DTS/SN (Suecia) [DTSSN]: Delay Tolerant Networks / Sensor Networks.
- ADA [ADAIST]: Redes de sensores distribuidas para la monitorización de la calidad del aire.

## 1.2 Línea de investigación

El Grupo de Sistemas Tolerantes a Fallos (GSTF) es cofundador del Instituto Universitario de Aplicaciones de las Tecnologías de la Información y de las Comunicaciones Avanzadas (ITACA), dentro de la Universidad Politécnica de Valencia (UPV), contando en la actualidad con 26 miembros.

Sus principales áreas de investigación son:

- Diseño, modelado y evaluación de Sistemas Tolerantes a Fallos (STF).
- Desarrollo de sistemas de inyección de fallos para la validación de STF.
- Desarrollo y programación de sistemas empotrados basados en microcontroladores.
- Redes industriales, redes inalámbricas de sensores y redes de sensores subacuáticas.
- Evaluación y mejora de la seguridad y la confiabilidad en redes de computadores inalámbricas y cableadas.

En el año 2000 el GSTF realizó el proyecto DPI 2000-0618 subvencionado por la CICYT, cuyo título fue “Desarrollo y análisis de aplicaciones distribuidas tolerantes a fallos basadas en *Time-Triggered Architecture* para entornos de automoción”. El autor de la presente tesis comenzó su actividad investigadora dentro de este proyecto, estudiando los aspectos de tolerancia a

fallos y tiempo real de las redes locales industriales para la implementación de aplicaciones distribuidas de alta confiabilidad.

Al término del mismo, tras su incorporación como profesor al Departamento de Informática de Sistemas y Computadores (DISCA) de la UPV, el trabajo de investigación del autor, de acuerdo con la línea docente correspondiente a informática industrial, redes de computadores y redes de área local e interconexión de redes, continuó en temas ligados a la problemática que conlleva el uso de las redes inalámbricas como base para la implementación de sistemas distribuidos para monitorización y control industrial, y posteriormente de forma más específica en redes inalámbricas de sensores para dotarlas de las características mencionadas. Dentro de esta línea de trabajo el autor ha participado y está participando en los siguientes proyectos de investigación, donde se han venido desarrollando y aplicando las técnicas y mecanismos propuestos en la presente tesis:

- “Análisis y evaluación de la integración de sistemas de visión sobre redes de área local industrial”. UPV. Ayudas para grupos interdisciplinares del programa incentivo a la investigación. Periodo 2002-2003.
- “Estudio e implementación de un sistema integral para la gestión y control de cargas eléctricas residenciales: Una oportunidad para la participación activa de usuarios finales en la disminución de picos de demanda y el uso eficiente de la energía”. Fundación Séneca – Murcia. Periodo 2006-2007.
- “Desarrollo de un sistema de protección forestal mediante redes de sensores inalámbricos (SENIN)”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- “Monitorización medioambiental mediante redes de sensores inalámbricos (MARESIN)”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- “Detección precoz del picudo rojo mediante redes de sensores inalámbricas”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- “Desarrollo de un sistema único de protección perimetral escalable y robusto”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- “Diseño, Evaluación e Implementación de Una Red Subacuática de Sensores”. Ministerio de Educación y Ciencia (CICYT DPI2007-66796-C03-01). Periodo 2007-2010.
- “Investigación básica fundamental sobre tecnologías constitutivas de un sistema de Red Inalámbrica de Sensores y su aplicación para el desarrollo de una Plataforma de Redes Inalámbricas de Sensores”. Ministerio de Educación y Ciencia (PET2008-0011). Periodo 2009-2011.
- “Wireless smart fire alarm system”. Programa Europeo EUREKA (Subprograma AVANZA I+D) TSI-020400-2008-180. 2009-2011.

### 1.3 Objetivos y aportaciones de la tesis

Fruto de la participación en los proyectos de investigación expuestos, se planteó el trabajo de investigación plasmado en la presente tesis, cuyos principales objetivos y aportaciones han sido:

1. Realizar un estudio de las tecnologías apropiadas para la implementación de redes de sensores inalámbricas, sus características y limitaciones, evaluación de plataformas hardware, medición de consumos, estudios de interferencia entre diferentes tecnologías inalámbricas, impacto de la ubicación de múltiples sumideros en una red, así como otros aspectos interesantes, de cara a plantear protocolos de nivel superior adecuados que permitan obtener las máximas prestaciones y mínimo consumo energético.
2. Analizar en profundidad los protocolos de encaminamiento existentes para redes inalámbricas de sensores, y las prestaciones que ofrecen a la hora de implementar una aplicación real. Las técnicas de enrutamiento se han estudiado observando aspectos como eficiencia energética, escalabilidad, tolerancia a fallos y tiempo real, comparando unas con otras para analizar las ventajas y desventajas de cada técnica.

Tomando las características anteriormente mencionadas y las conclusiones obtenidas, se han establecido y evaluado los parámetros de diseño deseables en un protocolo de enrutamiento para redes inalámbricas de sensores.

3. Diseñar, implementar y verificar el funcionamiento de un nuevo protocolo de enrutamiento para redes inalámbricas de sensores que responda a las características requeridas como son eficiencia energética, escalabilidad, flexibilidad, tolerancia a fallos, seguridad y acotación de tiempos en las comunicaciones, puesto que no es posible encontrar reunidas todas ellas en una misma arquitectura o protocolo actual. La verificación del mismo se ha realizado mediante dos vías complementarias: simulación e implementación. Para la simulación se ha desarrollado un modelo completo para el simulador NS-2 [NS09] ampliamente utilizado en la comunidad científica, que se ha validado mediante experimentación con nodos reales. Se han ajustado los parámetros y obtenido las expresiones para los umbrales. Por otro lado, también se ha verificado mediante implementación a través de su aplicación en diversos proyectos. Los requisitos necesarios se pueden resumir en los siguientes puntos:

- a) Prestaciones: Las prestaciones de los protocolos estudiados resultan generalmente insuficientes para los requisitos de gran parte de las aplicaciones planteadas en varios sentidos:
  - Consumo energético: Resulta un factor clave en las redes de sensores inalámbricas, dado que se caracterizan por nodos de bajo coste con grandes limitaciones de energía, en los que, en una gran parte de las aplicaciones, la vida del nodo termina cuando se agota la batería. El

protocolo propuesto saca el máximo partido a la fuente de energía de los nodos, reparte equánimamente la carga para maximizar la vida de la red, y es capaz de soportar redes heterogéneas, donde los nodos no son todos iguales, y donde por tanto se permite – y de hecho resulta más eficiente – cargar más a los nodos con más recursos, con el objetivo siempre de aprovechar las diferentes capacidades de cada uno al máximo y prolongar la vida de la red, cuestiones poco logradas en otras aproximaciones estudiadas.

- Sobrecarga: La sobrecarga, o proporción de información de control que no lleva datos útiles, que se ha observado en los protocolos estudiados resulta enorme, tanto para inicializar la red como para mantener y descubrir rutas, etc., puesto que es necesario el envío y difusión de gran cantidad de mensajes en la red, con el impacto que esto tiene en las prestaciones y en el consumo, siendo necesario por otro lado el proceso y almacenamiento de grandes cantidades de información en los nodos de la red.
  - Coste de implementación: Muchos de los protocolos estudiados requieren de los nodos una potencia de cálculo apreciable, procesadores dedicados a comunicaciones o grandes cantidades de memoria, lo cual repercute negativamente en el coste y consumo de los mismos. Resulta necesario que el nuevo protocolo requiera un mínimo de recursos y que, por tanto, sea adecuado incluso para microcontroladores de baja potencia, reduciéndose en el protocolo propuesto la complejidad de forma muy significativa.
- b) Escalabilidad: La escalabilidad resulta otro factor clave en las redes inalámbricas de sensores, que no se haya bien resuelto en la mayoría de protocolos estudiados, debido a sus hipótesis de partida (por ejemplo, los protocolos basados en *clusters* asumen habitualmente que los *cluster heads* están en cobertura del sumidero, y en general se asume un solo sumidero en la red). Todo ello los hace poco escalables dado que al aumentar el número de nodos o el diámetro de la red las comunicaciones se vuelven muy costosas – en cuanto a energía necesaria –, con una sobrecarga que va creciendo exponencialmente y con un cuello de botella consistente en un único sumidero que puede estrangular la red.
- c) Flexibilidad: los protocolos estudiados están pensados para redes homogéneas, donde todos los nodos son iguales y poseen idénticas capacidades, el protocolo propuesto se adaptará y optimizará el uso de recursos tanto en redes homogéneas como en redes heterogéneas, muy en auge, dado que pueden permitir configuraciones más óptimas, todo ello de forma automática. Además el protocolo es adecuado tanto para sistemas que requieran una baja frecuencia de muestreo de los sensores como para aquellos que necesiten una frecuencia elevada, e incluso para aplicaciones que requieran alarmas. Por otro lado, los protocolos estudiados tampoco dan soporte a esquemas de comunicación bidireccional (entre

sumidero y nodos) ni contemplan la comunicación entre nodos de la red (*peer-to-peer*), mientras que el protocolo propuesto permite estos perfiles de comunicación de forma sencilla y sin afectar al consumo ni a las cotas temporales de las comunicaciones hacia el sumidero. Además, toda la configuración de rutas, minimización del consumo, adaptación a posibles fallos, etc. se realiza de forma dinámica, distribuida y de forma totalmente automática.

- d) Tolerancia a fallos: generalmente los protocolos para redes de sensores inalámbricas no contemplan mecanismos específicos de tolerancia a fallos, al entender que la propia replicación de nodos que supone una red de sensores en sí es suficiente. El protocolo propuesto sí presenta mecanismos de tolerancia a fallos de forma que un fallo en un nodo no suponga tener que perder tiempo y energía en reconfigurar de nuevo toda la red (aunque dicho nodo ejerza un papel relevante en la red, como puede ser nodo líder o sumidero), ni pérdida de funcionalidad, y todo ello afectando lo mínimo posible al consumo de energía.
- e) Seguridad: es un aspecto de creciente importancia pero no contemplado generalmente en las arquitecturas para WSN. El protocolo propuesto contempla la seguridad en la red, estableciendo mecanismos para la autenticación de los nodos y transmisión de la información de forma segura, proporcionando confidencialidad e integridad en las comunicaciones.
- f) Acotación de tiempos en las comunicaciones: la gran mayoría de los protocolos estudiados no permiten ningún tipo de acotación de tiempos en las comunicaciones, siendo por tanto imposible dar una cota del tiempo necesario para que una trama generada por un nodo alcance el sumidero. Algunas aproximaciones lo contemplan, pagando el precio de una sobrecarga y un consumo considerablemente mayor. Es un objetivo del protocolo propuesto proporcionar comunicaciones acotadas en el tiempo sin penalizar en sobrecarga ni mayor consumo en todo el proceso de comunicación.

Para evaluar los mecanismos propuestos se utilizará una metodología basada en modelos de simulación, comparando la bondad de los mecanismos propuestos con otros ya existentes. De cara a obtener resultados fiables, se utilizará la herramienta de simulación *Network Simulator, ns-2*. Este simulador es uno de los más empleados en el área de investigación de las redes inalámbricas de sensores, lo cual está documentado en multitud de artículos referenciados.

## 1.4 Desarrollo de la tesis

La presente tesis se estructura en 6 capítulos en los que se presenta el trabajo realizado y se desarrollan las aportaciones antes citadas. En el capítulo 2 se refleja el resultado del estudio a fondo realizado sobre los protocolos de encaminamiento existentes para redes inalámbricas de sensores, poniendo de manifiesto para cada aproximación sus ventajas e inconvenientes, así como sus características de ahorro energético, escalabilidad, tolerancia a fallos y acotación

de tiempos de respuesta, lo cual proporciona finalmente un conocimiento global de las técnicas más empleadas hasta la actualidad, y que ha supuesto la propuesta de un nuevo protocolo que cumpla con todos los objetivos propuestos en el punto anterior.

En el capítulo 3 se presenta un estudio de las estrategias más adecuadas para la mejora del consumo energético en redes inalámbricas de sensores. Los resultados de este estudio marcarán las directrices de diseño de las diferentes aportaciones de la presente tesis, planteándose ya a lo largo del mismo, propuestas innovadoras que sirvieron de base para la arquitectura propuesta en el capítulo 4.

El capítulo 4 describe el protocolo de encaminamiento denominado *“Energy-efficient Adaptive hierarchical and robust Architecture”* (EETA). EETA es una de las principales aportaciones del presente trabajo, integrando aspectos de eficiencia energética, escalabilidad, flexibilidad, tolerancia a fallos y acotación de tiempos en las comunicaciones, constituyendo un nuevo paradigma en las RIS. El capítulo también presenta la evaluación del protocolo utilizando un modelo de simulación implementado y validado con datos reales a tal efecto, comparándose las prestaciones del mismo frente a otras propuestas, obteniéndose resultados muy satisfactorios. Se ha evaluado el impacto de parámetros tales como el número de nodos, el diámetro de la red y la carga utilizada. Los índices de prestaciones estudiados son el consumo energético, duración de la red y porcentaje de paquetes perdidos. Finalmente también se estudió la estabilidad del protocolo y su funcionamiento en presencia de fallos.

En el capítulo 5 se presenta la metodología seguida en la fase de experimentación, y se introduce el simulador NS-2 utilizado para evaluar las técnicas propuestas en la presente tesis, presentándose su estructura y funcionamiento. Asimismo se presentan algunas de las implementaciones realizadas para ensayar el nuevo protocolo, describiendo los escenarios y los resultados obtenidos en diversos campos.

Finalmente, el capítulo 6 presenta las conclusiones, aportaciones y publicaciones derivadas de la presente tesis, y se exponen las líneas de trabajo futuras.

## Capítulo 2

### Estado del arte

#### 2.1 Clasificación de las estrategias de enrutamiento en WSN

El gran auge de las redes de sensores inalámbricas tiene sus pilares básicos en su gran abanico de posibilidades, debido entre otros factores al bajo coste y miniaturización de los sensores y sistemas basados en microcontrolador actuales. Un nodo sensor generalmente es un sistema basado en microcontrolador que incorpora un módulo de Radio Frecuencia (RF), una unidad de sensorización, encargada de recoger los datos del entorno, todo ello alimentado mediante una batería. Como se ve detalladamente en la *figura 2.1*, estos nodos se pueden mejorar con *hardware* de posicionamiento (por ejemplo, *GPS*) o dotándolos de capacidad de movimiento, todo ello a costa de un mayor consumo.

La estrategia de crear sensores sofisticados con *hardware* adicional para implementar multitud de tareas diferentes es contraproducente, ya que, a pesar de dotar al nodo de gran versatilidad, conlleva un mayor consumo. Siguiendo esta línea, los sensores resultantes serán caros, requerirían un mantenimiento continuo (por ejemplo, para reemplazar las baterías) o alimentación continua, obligando a la instalación de cableado adicional. Este tipo de sensores, si bien podrían resultar de aplicación en algunos entornos, presentan evidentes desventajas, no justificándose en este caso, salvo excepciones puntuales, la utilización de comunicaciones inalámbricas.

Sin embargo, la tendencia actual consiste en la sustitución de estos sensores complejos por multitud de pequeños sensores simples, siguiendo el principio fundamental de reducir los costes. La implementación de nodos de bajo coste – y por tanto sencillos – es un primer factor de economía, mientras que la eliminación del mantenimiento – con los costes de mano de obra que ello conlleva – resulta un segundo factor decisivo. Mediante esta estrategia, el mantenimiento se reduce a repoblar el área monitorizada con nuevos nodos que se integren en la red actual para sustituir aquellos que han dejado de funcionar, bien por avería, accidente, etc. como por agotamiento de sus baterías.

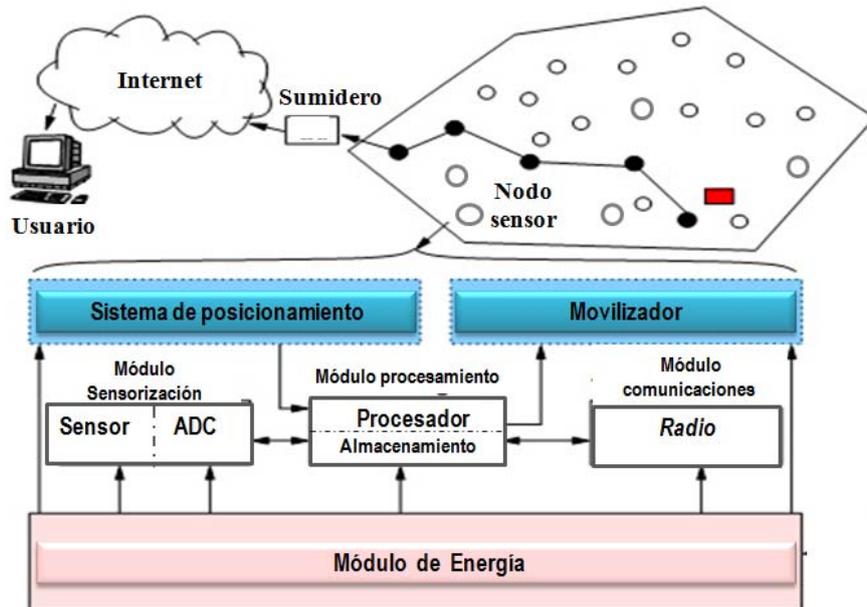


Figura 2.1. Componentes de un nodo sensor

Buscando este objetivo, resulta primordial gestionar correctamente el consumo de los nodos sensores, para así equilibrar las prestaciones del sensor con su consumo.

Uno de los factores que más influye en la limitación de consumo es el algoritmo de encaminamiento empleado.

En los últimos años se han desarrollado multitud de técnicas en esta dirección, proponiendo diversas alternativas sobre qué tipo de topología de red es la óptima para minimizar el consumo de energía y, en consecuencia, maximizar el tiempo de vida de la red.

De la mano de estas propuestas han aparecido multitud de protocolos, cada uno centrado en maximizar diferentes parámetros de la red. En [ALKARAK00] se propone una clasificación, a nivel de estructura de red y de operación del protocolo, que presenta una visión global de las diferentes técnicas de encaminamiento que se contemplan en la actualidad, y que podemos observar en la figura 2.2.

Basándonos en la forma de operar del protocolo, se distinguen cinco categorías:

1. **Basados en Negociación** (*Negotiation Based*), donde empleando una serie de mensajes de negociación se pretende eliminar duplicados en la información y prevenir que datos redundantes se envíen al siguiente nodo o al sumidero.
2. **Basados en Multiruta** (*Multi-Path Based*), donde se usan múltiples caminos en lugar de un único camino con el fin de mejorar el rendimiento.

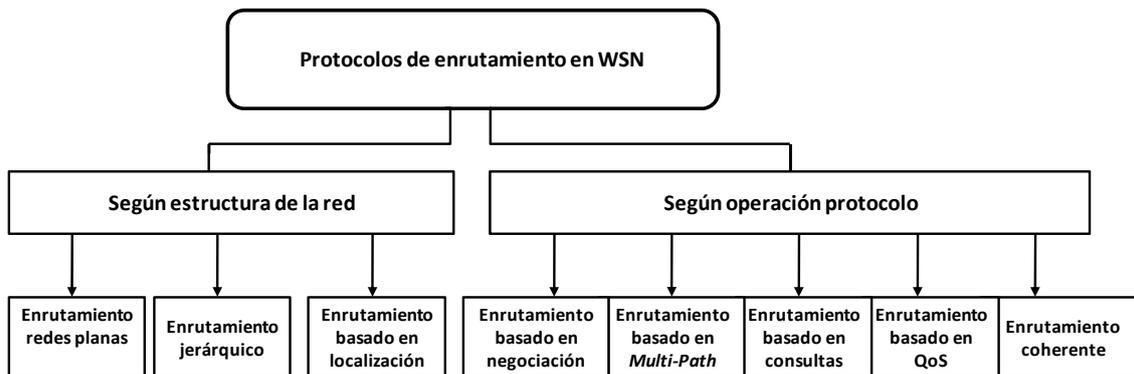


Figura 2.2. Taxonomía de los protocolos de enrutamiento en WSN

3. **Basados en Consultas** (*Query Based*), donde los nodos destinatarios propagan la consulta de información (tarea de sensorización) desde un nodo hacia la red y cuando se encuentra un nodo que posee dicha información, éste responde a la consulta enviando los datos al que inició la consulta.
4. **Basados en Calidad de Servicio** (*QoS Based*), donde la red debe satisfacer ciertas métricas de *QoS*, como *delay*, energía, ancho de banda, cuando envía datos al sumidero, manteniendo de esta forma la red balanceada en cuanto a consumo de energía y calidad de la información.
5. **Basados en Coherencia** (*Coherent Based*), donde la información es enviada después de un mínimo procesado a los nodos encargados de la agregación. El procesamiento en coherencia es una estrategia típica para elaborar algoritmos de enrutamiento eficientemente energéticos.

Basándonos en la **estructura de la red**, tenemos principalmente tres tipos de redes:

1. **Redes Planas** (*Flat Networks*), en las que todos los nodos desempeñan el mismo papel. En este tipo de redes la labor de sensorizar es realizada en colaboración.
2. **Redes Jerárquicas** (*Hierarchical Networks*), en las que existen nodos con distintos tipos de rol. Aquí se establecen diversos niveles en la red, en función del papel de los nodos.

3. **Encaminamiento basado en Localización** (*Location-based routing*), donde cada nodo dispone de un sistema que permite conocer la posición exacta del resto de nodos, y emplea esta información para la transmisión de datos.

A continuación pasamos a detallar los protocolos más destacados de cada rama según esta clasificación.

## 2.2 Redes planas

La estructura de red subyacente puede tener un papel muy importante en la operación del protocolo. La primera aproximación a la hora de diseñar protocolos de enrutamiento suele asumir que todos los nodos de la red son iguales, sin distinciones y considerando que todos van a realizar las mismas tareas. Este enfoque produce redes planas, donde todos los nodos realizan las mismas labores de enrutamiento y procesamiento, sin aplicar diferentes roles a los sensores y colaborando conjuntamente en las tareas de sensorización.

En este caso, en muchos sistemas resulta poco apropiado asignar un identificador global a cada nodo, debido al gran número de éstos en la red. Esta consideración ha llevado a la implantación de encaminamiento centralizado, donde la Estación Base (*Base Station*) ó sumidero (*Sink*) envía consultas a ciertas regiones y espera los datos enviados por los sensores. Los datos se negocian vía consultas (*queries*), por lo que se requiere mayoritariamente algún esquema de direccionamiento mediante atributos para especificar las propiedades de esos datos.

En los inicios del estudio en encaminamiento centralizado, los protocolos *SPIN* y Difusión Directa mantenían el ahorro energético mediante negociación y eliminación de datos redundantes. Estos dos protocolos, que se exponen a continuación, motivaron el diseño de muchos otros que siguen un concepto similar.

### 2.2.1 *Sensor Protocols for Information via Negotiation (SPIN)*

En *SPIN* [KULIK99] se disemina toda la información de cada nodo al resto de nodos de la red asumiendo de esta forma que todos los nodos son estaciones base en potencia. Esto permite al usuario final realizar consultas y recibir la información inmediatamente. Los protocolos de esta familia se aprovechan de la propiedad de que los nodos que se encuentran cercanos unos a otros poseen datos similares y, por lo tanto, únicamente se necesita diseminar la información que los demás nodos no conocen.

El concepto de inundación o *flooding* en si mismo tiene dos grandes problemas como podemos apreciar en la figura 2.3; el problema de la implosión, que provoca que se envíen varias copias de los datos a los nodos malgastando recursos, y el problema de la superposición (*overlap*), provocado por el hecho de que zonas colindantes son cubiertas por diversos nodos, y por tanto se transmite información replicada sobre dichas zonas a los demás nodos.

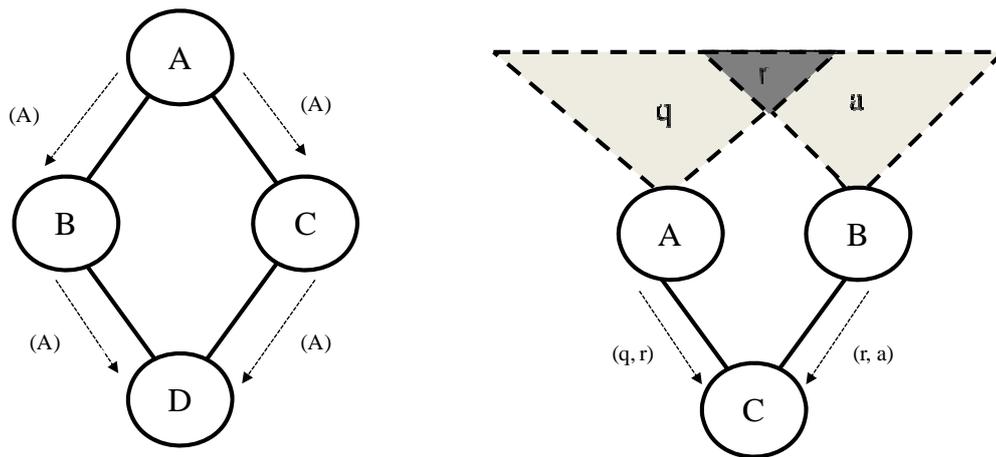


Figura 2.3. Problema de la implosión (izquierda) y *overlap* (derecha)

Para paliar los efectos de estos problemas, los nodos en *SPIN* asignan un nombre de alto nivel a la información que recogen (*meta-datos*) para describirla totalmente y negocian los unos con los otros con estos *metadatos* antes de transmitirlos. Con esta negociación se asegura que únicamente se enviará información útil (figura 2.4).

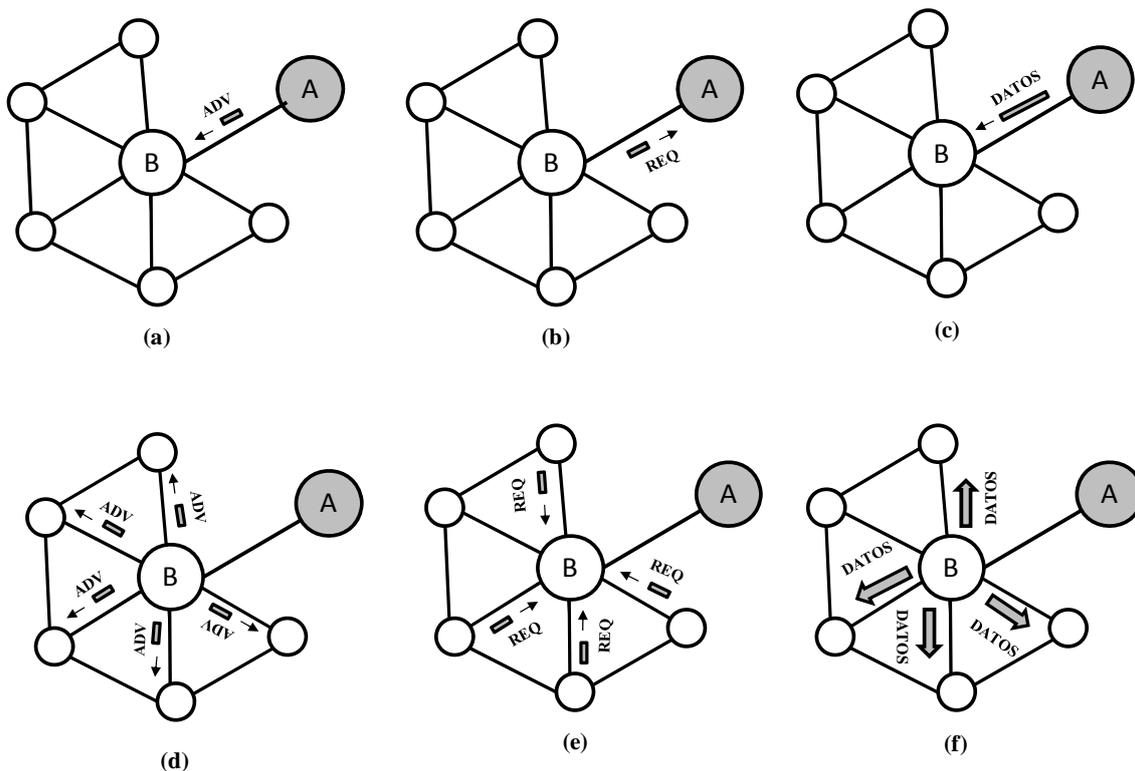


Figura 2.4. Negociación en *SPIN*: (a) El nodo A comienza advirtiéndole al nodo B de que posee información para compartir. (b) El nodo B responde enviando al nodo A una petición. (c) Se reciben los datos, (d) el nodo B envía advertencias a sus vecinos, (e) dichos vecinos envían peticiones de vuelta a B, (f) a lo que B contesta enviándoles el dato

En líneas generales, los protocolos de la familia de *SPIN* se apoyan en dos ideas básicas:

1. Los sensores operan más eficientemente y conservan mejor la energía enviando información descriptiva sobre los datos sensorizados en lugar de enviar los datos mismos.
2. La difusión (*flooding*) convencional malgasta mucha energía y ancho de banda cuando envía copias innecesarias de los datos, como consecuencia de una mala gestión de los sensores en solapamiento (*overlapping*). Seleccionando un nodo aleatorio al que mandarle la información en lugar de realizar difusión, se soluciona el problema en primera instancia, sin embargo eleva significativamente las latencias a la hora de propagar la información entre los nodos de la red.

Una de las ventajas de *SPIN* es que los cambios en la topología son localizados, es decir, debido a que cada nodo solo necesita saber la posición de su vecino directo en *single-hop*, los cambios no son costosos. El ahorro energético en comparación con *flooding* clásico es considerable. Sin embargo, *SPIN* posee varias desventajas muy a tener en cuenta:

- Las advertencias de envío que realiza *SPIN* no aseguran la entrega de la información. Para comprender esto, supóngase una aplicación de detección de intrusiones donde los datos deben entregarse de forma fiable en intervalos de tiempo constantes. Si se asume que los nodos interesados en la información están lejos del nodo fuente y que los nodos entre fuente y destino no están interesados en la información, dicha información nunca será entregada a su destinatario, lo que provoca una deficiencia en la capacidad de difusión (*QoS*) de *SPIN*.
- Debido a la decisión forzada de aplicar negociación para evitar los problemas de *flooding* y *overlap* clásicos, se impone una sobrecarga sustancial en los envíos debido a la negociación previa realizada por los nodos y, por lo tanto, esto afecta al tiempo de vida de la red que, aún siendo competente en comparación con el *flooding* clásico, no resulta competitiva en los términos en los que está basado este trabajo.
- Además, dicha negociación, provoca que el algoritmo carezca de escalabilidad, si no se imponen mecanismos de eficiencia en su envío, debido a que cuando se incrementa el número de nodos de la red el número de vecinos por nodo, y por tanto el número de mensajes, se multiplica, agravando el problema.

### 2.2.2 Difusión Directa

Difusión Directa (*Directed Diffusion* [YE01]) es un paradigma centrado en los datos y orientado a la aplicación en el sentido en que toda la información generada por los sensores se nombra mediante pares de *atributo-valor*. Se disemina una tarea de sensorización por la red de

sensores como “interés” de cierto dato nombrado. Este interés describe la tarea a realizar en la red y se difunde por la misma salto a salto, realizando cada nodo una difusión a sus vecinos. Esta difusión configura *gradientes* en la red, los cuales designan un esquema de eventos, esto es, una concordancia *interés-dato*. En líneas generales, un gradiente especifica el valor de un atributo y su dirección. La fuerza del gradiente puede ser diferente dependiendo de los vecinos, lo que conlleva posibles diferencias en los flujos de información.

En esta etapa los bucles no se comprueban pero posteriormente se eliminarán. Los eventos irán fluyendo a lo largo de los generadores de interés por múltiples caminos. El algoritmo posteriormente refuerza uno ó un conjunto de estos caminos, como se puede observar en la figura 2.5.

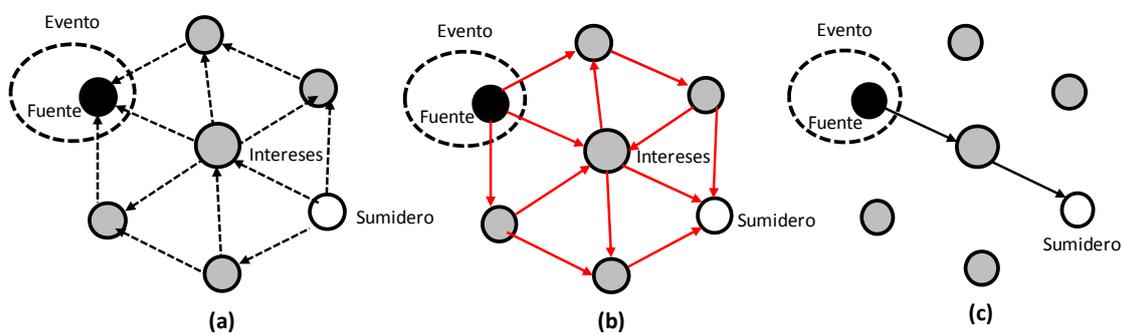


Figura 2.5. (a) Propagación del interés, (b) configuración inicial de los gradientes y (c) entrega de la información mediante el camino reforzado

La idea principal del paradigma es combinar los datos procedentes de diferentes fuentes para eliminar la redundancia, minimizando de esta forma el número de transmisiones y con ello aumentando el ahorro energético, ya que se seleccionan empíricamente los mejores caminos para la difusión.

Difusión Directa difiere de *SPIN* en dos aspectos; primero, Difusión Directa se centra en peticiones de información bajo demanda, es decir, el sumidero envía consultas a los nodos por medio de inundación (*flooding*). En *SPIN*, sin embargo, los sensores advierten la disponibilidad de información a los demás, permitiendo de esta forma que sean los nodos vecinos los que muestren interés por los datos obtenidos. Segundo, toda la información en Difusión Directa es vecino a vecino, donde cada nodo tiene la capacidad de realizar agregación de la información.. De forma contraria a *SPIN*, en Difusión Directa no hay necesidad de mantener una topología de red global.

Por el contrario, este mecanismo también presenta desventajas claras:

- Difusión Directa no es aplicable a aplicaciones que requieran una continua entrega de datos al sumidero, puesto que el modelo basado en consultas por demanda no ayuda a este propósito. El continuo *flooding* al que se vería sometida la red introduciría una sobrecarga más que notable. De hecho, en el propio trabajo de investigación

referenciado, los autores consiguen los resultados en entornos lejanos a la alta carga, es decir, en entornos sin congestión y con poca asiduidad de consultas, ya que no se disponen de mecanismos para evitar la sobrecarga por inundación y por tanto aumentar el tiempo de vida de la red al retrasar el agotamiento de la batería.

- Esta inundación provoca que el protocolo no sea escalable, alejándolo de toda viabilidad cuando se requiere que tome el control de redes con gran cantidad de nodos.
- Como consecuencia también del empleo de *flooding* en la red, no es posible garantizar ningún tipo de calidad en el servicio de entrega ya que la latencia y las pérdidas de paquetes resultan incontrolables. Por ello, el autor considera que en lo referente a *QoS* este protocolo no aporta nada.
- Además, la elección de los caminos se realiza de forma poco optimizada y la difusión es estable solo en determinados escenarios, luego dista mucho de una buena solución.

### 2.2.3 Rumor routing

*Rumor routing* [BRAGINS02] es una variación de Difusión Directa y fue concebido para entornos en los cuales el encaminamiento geográfico no es posible, bien porque no se dispongan de sistemas de coordinación o bien porque el fenómeno de interés no está geográficamente correlacionado.

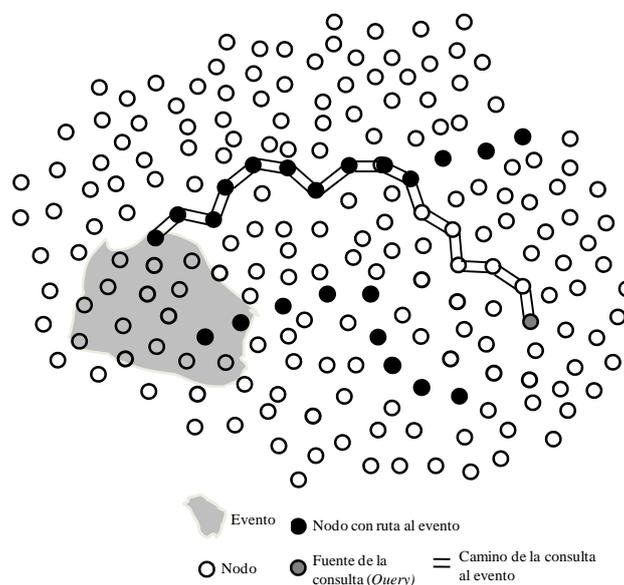


Figura 2.6. Operación protocolo RR: La consulta se origina desde el nodo fuente (gris) y busca un camino hacia el evento. Tan pronto encuentre un nodo correspondiente a dicho camino, será redireccionada directamente hacia el evento

En general, Difusión Directa emplea *flooding* para inyectar la consulta en la red y difundir las tareas. Sin embargo, en algunos casos solo hay una pequeña cantidad de información

requerida por parte de los nodos y por ello el uso de inundación es desproporcionado. *Rumor routing* propone una solución lógica y de compromiso entre el *flooding* de consultas y el *flooding* de notificaciones de eventos. La idea es crear caminos que lleven a cada evento para de esta forma, cuando se genera una consulta, ésta se manda y retransmite aleatoriamente hasta que encuentra el camino del evento, en vez de emplear inundación para alcanzar dicho evento. En la *figura 2.6* observamos un ejemplo de cómo *RR* encuentra dicho camino.

En caso de que no se encuentre el camino hacia el evento, en última instancia el nodo decidirá usar inundación. Bajo ciertas condiciones, se puede emplear una solución híbrida para mejorar el rendimiento en términos energéticos, pero el uso de inundación, aunque sea en menor medida, acarrea las consecuencias que ya se vieron en puntos anteriores.

Los resultados de las simulaciones muestran que *Rumor routing* puede lograr un ahorro energético significativo cuando se compara con *flooding* de eventos, obteniendo también una mínima tolerancia a fallos en los nodos, ya que si uno o varios nodos pertenecientes a un camino hacia un determinado evento cayeran, existiría al menos otro camino en la dirección del mismo evento. Sin embargo, *Rumor routing* se comporta bien solo cuando el número de eventos es pequeño. Para un gran número de ellos, el coste de mantener las tablas de eventos en cada nodo llega a ser impracticable si no son de interés para el sumidero.

#### 2.2.4 Minimum Cost Forwarding Algorithm (MCFA)

El algoritmo *MCFA* [YE01] explota el hecho de que la dirección del encaminamiento es siempre conocida. Cada nodo mantiene una estimación del camino de menor coste hasta el sumidero. Cada mensaje que debe ser enrutado por el nodo es enviado mediante *broadcast* a sus vecinos. Cuando un nodo recibe el mensaje, comprueba si se encuentra en el camino de menor coste entre el nodo emisor y el sumidero. Si es el caso, lo reenvía mediante difusión de nuevo a sus vecinos. Este proceso se repite hasta que se alcanza el sumidero.

La comprobación de si se está o no en el camino de menor coste se realiza de la siguiente manera:

1. El sumidero realiza la difusión de un mensaje con el coste a 0 mientras que cada nodo inicializa su menor coste hasta la estación base a  $\infty$ .
2. Una vez recibido el mensaje originado por el sumidero, cada nodo comprueba si la estimación en el mensaje más el coste del *link* por el cual ha recibido ese mensaje es menor que la estimación actual. De ser así, se actualiza tanto la estimación del mensaje del sumidero como la estimación actual para el nodo.
3. Si el mensaje *broadcast* se ha modificado, se reenvía. Si no, se purga y no se realiza ninguna acción posterior.

Este procedimiento revela dos desventajas de *MCFA*:

- Un nodo en particular recibirá múltiples actualizaciones. Para evitar esto, se propone un algoritmo de *backoff* para retrasar el envío de las actualizaciones y acotarlas pero no eliminarlas.
- Acotando estas actualizaciones de cierta manera se hace el protocolo más escalable, pero llegado un cierto punto no muy lejano, la escalabilidad se volvería a romper. Cambiar el enfoque es primordial para obtener una buena escalabilidad.
- Debido a la forma en la que se producen las actualizaciones, los nodos más lejanos al sumidero recibirán más actualizaciones que los que están más cerca, lo que provoca una sobrecarga tanto energéticamente en los nodos como de mensajes en la red.
- Dicha sobrecarga implica congestión en determinadas circunstancias, lo que conlleva problemas de latencia y la pérdida de una mínima calidad de servicio (*QoS*).

### 2.2.5 Gradient-Based Routing (GBR)

*Gradient-Based Routing* [SCHURGO1] es una nueva variante de Difusión Directa. La idea clave subyacente en *GBR* es memorizar el número de saltos cuando el mensaje es difundido en la red. De esta forma, cada nodo calcula un parámetro conocido como la “altura” del nodo, que es el mínimo número de saltos hasta alcanzar el sumidero. La diferencia entre su propia *altura* y la de sus vecinos es lo que se llama gradiente del enlace. Los paquetes son retransmitidos por el enlace con el mayor gradiente. *GBR* usa algunas técnicas auxiliares como agregación de la información o propagación del tráfico con el objetivo de dividir dicho tráfico por toda la red. Podemos encontrar varias técnicas de diseminación como, por ejemplo, la basada en energía, donde un nodo incrementa su altura cuando su energía disminuye por debajo de cierto umbral, con lo que consigue que los demás nodos no le elijan para enviarle información. Además, dicho nodo propaga sus actualizaciones en la red tan lejos como sea necesario para mantener la consistencia de los gradientes. El objetivo primordial de estas técnicas es obtener una distribución balanceada del tráfico de la red y, por lo tanto, incrementar el tiempo de vida de la misma pero esto no es del todo plausible, ya que la propagación de sus actualizaciones añade una considerable sobrecarga a la red, lo que provoca problemas de latencia y congestión, anulando escalabilidad alguna y como punto primordial, un mayor consumo.

### 2.2.6 Information-driven sensor querying (IDSQ) y Constrained anisotropic diffusion routing (CADR)

*CADR* [CHU02] pretende generalizar la idea de Difusión Directa. La clave aquí es consultar a los sensores y encaminar la información en la red de forma que el número de paquetes entregándose maximice, mientras que la latencia y el ancho de banda se minimicen. *CADR* difunde consultas usando un criterio de selección para elegir los sensores que tengan más probabilidades de obtener los datos deseados. Esto lo consigue activando únicamente aquellos sensores que estén cerca de un evento determinado y ajustando dinámicamente las rutas de

información. Al parámetro involucrado en esta selección de los nodos a activar se le conoce como " $\gamma$ " ó *tradeoff parameter*. Mediante el ajuste empírico de este parámetro se consigue que la consulta alcance la zona del evento junto con los nodos que participan en su sensorización. Dicho ajuste es vital para obtener la ruta de información, como se observa en la figura 2.7.

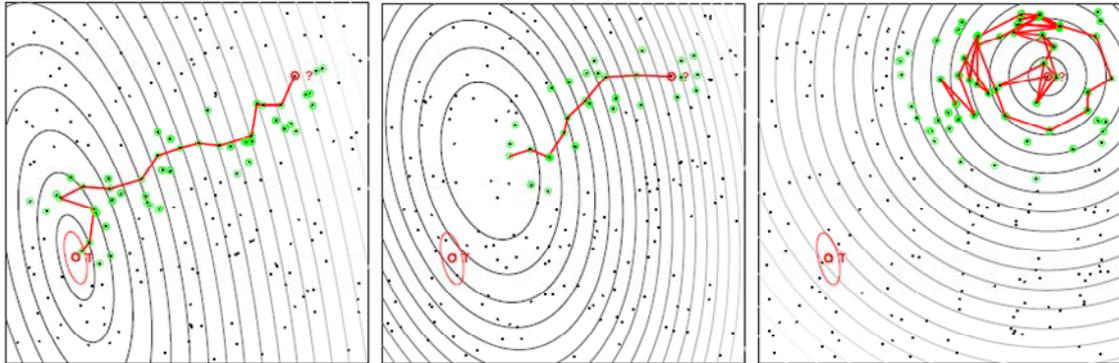


Figura 2.7. CADR para  $N = 200$  nodos. De izquierda a derecha:  $\gamma = 1$ ,  $\gamma = 0.2$ ,  $\gamma = 0.0$ . A efectos prácticos, la posición del evento 'T' (círculo rojo) y la de la consulta '?' son fijas e invariantes

La principal diferencia con Difusión Directa es la consideración del aumento de la productividad junto con el coste de comunicación. En *CADR*, cada nodo evalúa un índice información/coste y encamina los datos basándose en el gradiente información-coste local y los requerimientos del usuario final. En *IDSQ*, el nodo que consulta puede determinar qué nodo le puede proporcionar la información que requiere con el menor coste energético. Los resultados de las simulaciones muestran que estas aproximaciones son más eficientes en términos energéticos que las de Difusión Directa pero cuando la carga de consultas es alta, se produce una sobrecarga en la red de sensores (debido a su naturaleza basada en *queries*) con la consecuente pérdida de calidad de servicio.

### 2.2.7 COUGAR

El protocolo COUGAR [YAO02] se centra en los datos y considera la red como una gigantesca base de datos distribuida. La idea consiste en usar consultas declarativas con el fin de abstraer el procesamiento de las consultas de las funciones de la capa de red, como la selección de sensores relevantes o similares. Dada una consulta de usuario, un optimizador de consultas genera una planificación de la consulta para el procesamiento de la misma dentro de la red, consiguiendo de esta forma reducir drásticamente el consumo de recursos y, por lo tanto, aumentar el tiempo de vida de la red. Empleando agregación de datos en la red se puede reducir el consumo, pero, como en el caso de sus predecesores, basándose en consultas y difusión de las mismas, el ahorro energético global siempre será menor, lo que acarrea una desventaja notable.

### 2.2.8 Active Query forwarding in sensor networks (ACQUIRE)

*ACQUIRE* [SADAGOP03] es similar a *COUGAR*, *ACQUIRE* también ve la red como una gran base de datos donde consultas complejas se dividen en varias subconsultas, cada una de ellas sobre una variable de interés. La forma en la que trabaja *ACQUIRE* le permite resolver consultas complejas aún siendo éstas sobre varias variables a la vez.

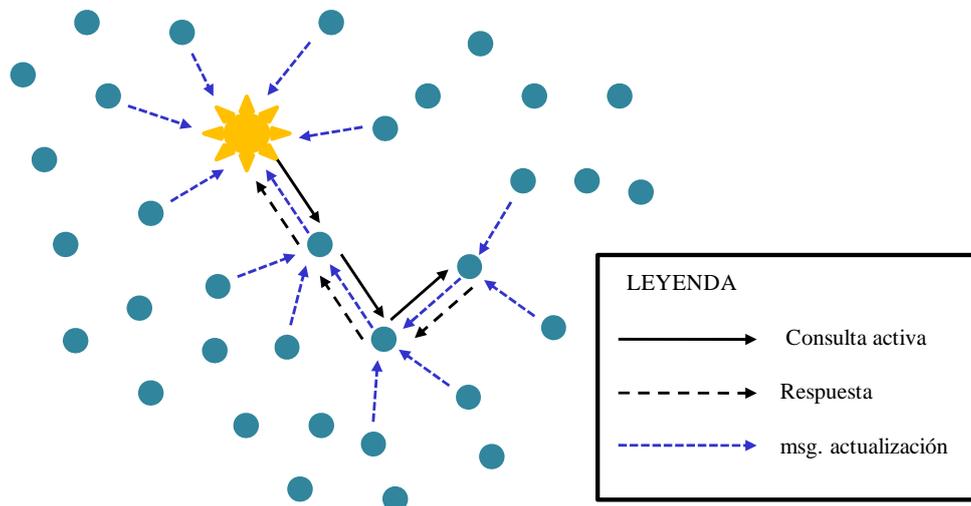


Figura 2.8. Operación protocolo *ACQUIRE*: En cada paso de la propagación de la consulta activa, el nodo portador de la consulta emplea el conocimiento adquirido mediante las actualizaciones desde todos sus nodos vecinos en un radio de  $d$  saltos, con el fin de resolver parcialmente la consulta

Las operaciones de *ACQUIRE* se pueden organizar como sigue:

1. El sumidero envía la consulta, la cual es retransmitida salto a salto por todos los nodos que la reciban.
2. Durante cada paso intermedio de este proceso, el nodo que está actualmente transmitiendo la *consulta activa* (*nodo activo*) utiliza actualizaciones recibidas por sus vecinos en un radio de  $d$  saltos para intentar responder parcialmente a la consulta.
3. Las actualizaciones son lanzadas de forma reactiva por el nodo activo una vez recibida la consulta activa solo si la información *pre-almacenada* actual que posee el nodo es obsoleta.
4. Una vez haya resuelto la consulta todo lo posible, el nodo activo elige un nuevo nodo al que retransmitirle la consulta. Esta elección se realiza de forma aleatoria o de forma inteligente, basada en conseguir la máxima resolución de la consulta por sucesivos nodos.
5. A medida que la consulta va profundizando en la red, se va haciendo más “pequeña”, dividiéndose en piezas menores que han ido resolviendo los nodos anteriores, hasta que llega a un nodo activo que puede resolverla completamente, esto es, resolver la última parte de la consulta.
6. En este punto, la consulta pasa a ser una *respuesta completa* y se envía de vuelta por el camino de menor coste o por el camino inverso (*reverse path*).

Un ejemplo de este esquema para  $d = 1$  en *ACQUIRE* se muestra en la *figura 2.8*.

La técnica de Difusión Directa no se emplea para resolver consultas complejas debido a las restricciones de energía que presenta, puesto que la Difusión Directa está basada en mecanismos de inundación para realizar la agregación de las mismas. Por este hecho, *ACQUIRE* es una buena solución si no fuera por tres desventajas relacionadas con la forma en la que la consulta se va inyectando en la red:

- Conforme  $d$  incrementa su valor, la consulta activa ha de viajar menos saltos, como media, pero esto conlleva un mayor coste debido a que las actualizaciones de todos sus vecinos provocan un mayor número de mensajes hacia el nodo activo, y por tanto más **sobrecarga** de mensajes para, generalmente, completar parcialmente la consulta.
- El número de vecinos y sus posteriores actualizaciones hacia el nodo activo es proporcional al número de sensores que haya a  $d$  saltos de distancia. Así, en el caso de redes con alta densidad de nodos, el algoritmo sufre mucha sobrecarga, y por tanto pierde toda su **escalabilidad**.
- Conforme el valor  $d$  aumenta, el comportamiento del protocolo se aproxima a un comportamiento equivalente a la inundación clásica, con los problemas que esto conlleva, siendo el caso límite cuando su valor se aproxima al diámetro de la red.
- Por el contrario, si  $d$  toma valores pequeños, el número de actualizaciones es adecuado, pero el camino generado durante la resolución de la consulta es mayor, lo que conlleva una mayor longitud en el camino de vuelta y una mayor **latencia**.

Para encontrar el valor del parámetro  $d$  óptimo, viendo las restricciones anteriores, los autores se basan en un modelo matemático que considera una red de sensores basada en topología *grid*, donde cada nodo tiene 4 vecinos inmediatos. Aunque se realiza el modelo, no exponen resultados validados por simulación, luego su posible eficacia queda en el aire dejando las desventajas anteriores en primer plano.

### 2.2.9 Energy Aware Routing

La mayoría de esquemas que pretenden maximizar la vida de los sensores van en la dirección de encontrar el camino de mínima energía para optimizar el consumo del nodo. Los autores de este protocolo proponen que no siempre usar el camino de menor energía conlleva el camino óptimo desde el punto de vista del tiempo de vida de la red y la conectividad a largo plazo [SHAH02]. El problema en potencia que suele ocurrir en los protocolos que siempre aplican el camino de menor coste energético para sus transmisiones es que los nodos involucrados en este camino terminan por agotar su batería mucho antes que el resto, lo que provoca la partición de la red. Por lo tanto, el objetivo de este protocolo es encontrar caminos sub-

óptimos para permitir a los nodos ahorrar energía, y a su vez distribuir el consumo más equitativamente.

Aunque este protocolo pueda parecerse a Difusión Directa, difiere de él en el sentido en el que se mantiene un conjunto de rutas en vez de mantener un único camino óptimo. La idea básica consiste en incrementar la longevidad de la red no siempre seleccionando el camino óptimo, sino alternándolo con elecciones de caminos sub-óptimos. De esta forma se asegura que los nodos pertenecientes al camino óptimo no se agoten prematuramente y la red vaya gradualmente perdiendo su energía de forma más uniforme, en lugar de dividirse por la desaparición de nodos. Estos caminos sub-óptimos poseen una probabilidad de ser escogidos dependiendo de su métrica energética, es decir, de lo que cueste pasar por él en términos de consumo. Eligiendo diferentes rutas dentro de este conjunto de caminos, la energía de cada uno de ellos no termina por agotarse, consiguiendo así alargar la vida promedio de la red. A su vez, la posibilidad de elegir varios caminos de entre un conjunto de caminos sub-óptimos, añade **tolerancia a fallos** a la red de sensores.

Cuando se compara con Difusión Directa, este protocolo logra un 44% de incremento en el tiempo de vida de la red y una mejora del 21.5% de ahorro en términos energéticos, como puede apreciarse en la *figura 2.9*.

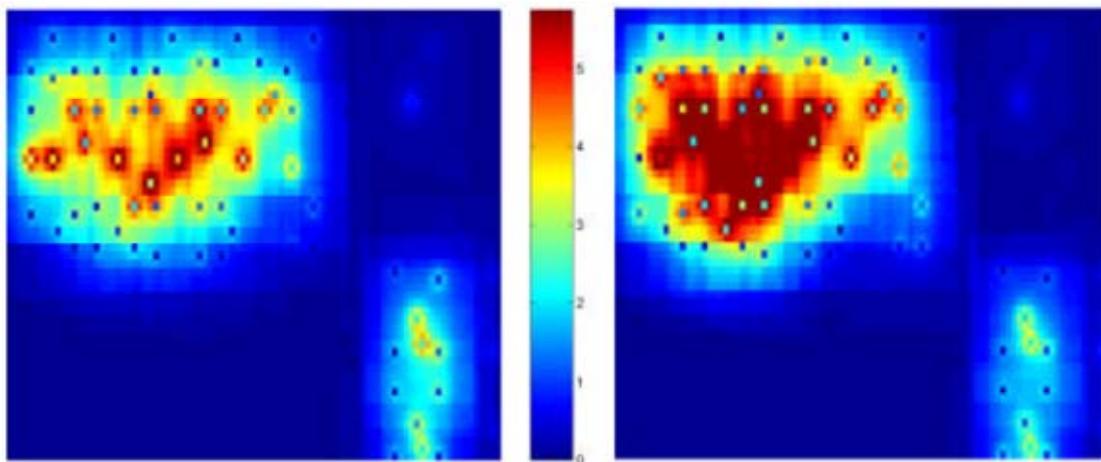


Figura 2.9. Energía consumida ( $\mu\text{J}$ ) para Energy Aware y Difusión Directa, de izquierda a derecha respectivamente

El algoritmo consigue homogeneizar la pérdida de energía en los nodos de la red, pero debido a su carácter proactivo, necesario para el mantenimiento de los conjuntos de caminos sub-óptimos y su posterior actualización, se puede incidir en una sobrecarga extra, añadiendo problemas de eficiencia energética.

### 2.2.10 Random Walks Routing Protocols

El objetivo de estos protocolos de enrutamiento es alcanzar una carga equitativa en un sentido estadístico haciendo uso de encaminamiento *multipath*. Esta técnica [SERVETT02] considera

únicamente grandes redes donde los nodos poseen altas restricciones de movilidad. La topología usada por los autores se basa en la construcción de pasos aleatorios (*random walks*) en una familia específica de grafos aleatorios, la cual ha sido elegida para abstraer el comportamiento de una red de micro-sensores. La topología se muestra en la figura 2.10.

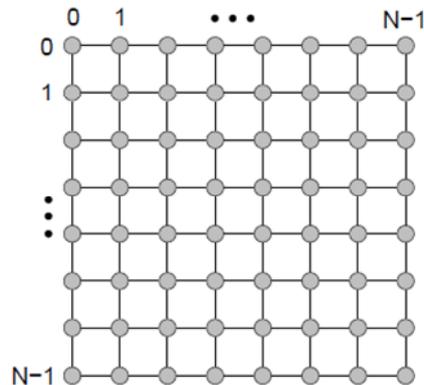


Figura 2.10. Malla cuadrada de tamaño  $N \times N$ . Los paquetes se inyectan en el nodo fuente situado en  $[0,0]$  y deben viajar salto a salto hasta el nodo destino  $[N-1,N-1]$ . Los nodos interiores  $[i, j]$  están conectados a 4 vecinos:  $[i-1, j]$ ,  $[i, j-1]$ ,  $[i+1, j]$  y  $[i, j+1]$ ; los dos primeros están más cerca de la fuente y los dos últimos lo están del destino

Se asume que los nodos pueden encenderse o apagarse arbitrariamente. Un nodo será elegido como nodo intermedio y como siguiente salto del enrutamiento mediante cierta función de probabilidad. Mediante la manipulación cuidadosa de dicha probabilidad se puede obtener el comedido balanceo de la red. Por ello, los autores dividen la topología de red anterior en dos regiones:

1. En la **etapa de expansión**, los paquetes se mueven a través de diagonales incrementando el número de nodos y, consecuentemente, la densidad de paquetes por nodo disminuye.
2. En la **etapa de compresión**, los paquetes se mueven a través de diagonales disminuyendo el número de nodos y, consecuentemente, la densidad de paquetes por nodo aumenta.

Estas dos etapas, junto con las probabilidades asociadas para determinar los caminos a seguir, se pueden ver más claramente en la *figura 2.11*:

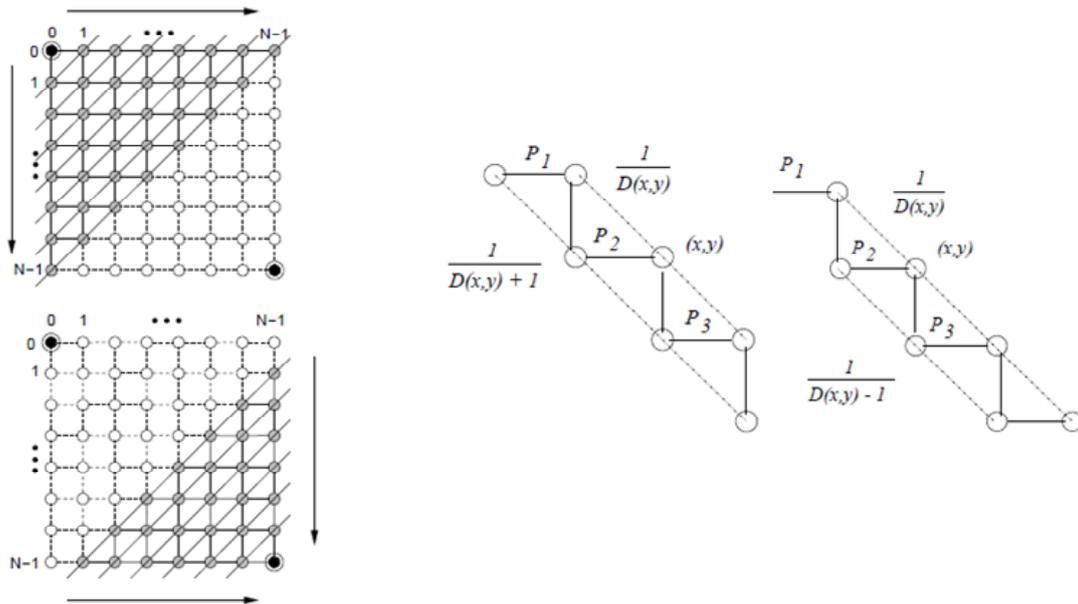


Figura 2.11. Etapas de expansión y compresión (izquierda) y probabilidades de retransmisión (derecha), donde se distinguen probabilidades para la etapa de expansión y compresión, respectivamente

De esta forma, el algoritmo es sencillo debido a que los nodos no han de mantener información de estado alguna. Además, se pueden elegir diferentes rutas en diferentes ocasiones incluso para el mismo par de nodos fuente-destino. Sin embargo, la desventaja primordial de este protocolo es que la topología de red dista mucho de ser práctica.

### 2.2.11 Conclusiones sobre redes planas

Como se ha visto a lo largo de los protocolos y estrategias algorítmicas anteriores, las redes planas tienden a la creación de algoritmos de enrutamiento basados en consultas (*queries*) y a abstraer la topología de red como si de una gran base de datos se tratara, haciendo que las retransmisiones y el problema del ahorro energético se transformen en una problemática centrada en cómo resolver y propagar dichas consultas. Algunas soluciones abordan el problema mediante inundación (*flooding*), descartándose directamente debido al excesivo coste y los problemas de latencia, congestión e inviabilidad de QoS que conllevan. Otros protocolos apuestan por simplificar dicha inundación convirtiéndola en una difusión controlada a sus vecinos ó dividiendo la consulta general en partes más pequeñas, cuya resolución en principio debe ser más sencilla; pero la verdadera solución para abordar con garantías el problema de **sobrecarga** y la poca **escalabilidad** que poseen todos estos esquemas es cambiar totalmente el punto de vista. El que todos los nodos que conforman la red sean vistos de igual manera no aprovecha las posibilidades de la totalidad de sensores, y es por ello que las mejoras energéticas que proporcionan estos protocolos son insuficientes. El dotar a los nodos de diferentes roles temporales para realizar tareas de procesamiento o control dentro del algoritmo, exprimirá al máximo las capacidades energéticas de la red y es la idea básica de

la siguiente rama de protocolos, los cuales se centran en **soluciones jerárquicas y particionales**.

## 2.3 Redes jerárquicas

El enrutamiento jerárquico ó basado en *clústeres*, originario de las redes interconectadas por cable, es una técnica bien conocida y con ventajas especiales relacionadas con la escalabilidad y eficiencia en sus comunicaciones. Como tal, el concepto de encaminamiento jerárquico también se ha aplicado para lograr eficiencia energética en *WSNs*. En una arquitectura jerárquica, los nodos con gran cantidad de energía pueden ser empleados para procesar y enviar la información mientras que los nodos de baja energía pueden ser usados únicamente para labores de sensorización en regiones cercanas al objetivo. Esto quiere decir que la creación de *clústeres* y la asignación de tareas especiales a sus *Clúster Heads (CHs)* pueden contribuir enormemente a la **escalabilidad** del sistema, su **tiempo de vida** y su eficiencia energética.

El encaminamiento jerárquico es una manera eficiente de disminuir el consumo de energía dentro del *clúster* y realizar tareas de **agregación** y **fusión** de información, con el fin de disminuir el número de mensajes transmitidos al sumidero.

A continuación, presentamos los protocolos más representativos de esta aproximación.

### 2.3.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

*LEACH* [HEINZEL00] es un protocolo basado en *clústeres*. *LEACH* selecciona aleatoriamente unos pocos sensores como *CHs* y turna este rol para distribuir la carga energética entre todos los nodos de la red. En *LEACH*, cada nodo *CH* comprime la información procedente de los nodos que pertenecen a su *clúster* (agregación) y envía el resultado al sumidero para reducir la cantidad de información que debe ser transmitida a la estación base. *LEACH* emplea *TDMC/CDMA MAC* para reducir las colisiones *inter-clúster* e *intra-clúster*. Este protocolo es adecuado cuando se necesita una constante monitorización por parte de la red de sensores. Un usuario no debe, en principio, necesitar la información inmediatamente. Es por ello que no se necesitan transmisiones periódicas de la información las cuales terminarían con la limitada energía de los sensores. Después de un intervalo aleatorio de tiempo, el rol de *CH* cambia a otro conjunto de nodos aleatorio con el fin de que el consumo de energía sea uniforme para toda la red.

Las operaciones de *LEACH* están organizadas en **dos fases**: la *fase de configuración* o *set-up phase* y la *fase de operación* o *steady-state phase*. En la **fase de configuración**, se eligen los *CHs* y se organizan los *clústeres*. En la **fase de operación**, se realizan las transferencias de información hacia el sumidero. La duración de la fase de operación es mayor que la fase de

configuración con el objetivo de minimizar la sobrecarga. Durante la fase de configuración, una determinada fracción de los nodos  $p$  se erige como *CHs*. Cada *CH* electo transmite un mensaje *broadcast* de notificación al resto de nodos de la red advirtiéndole de su *status* de *CH*. Todos los nodos restantes que no son *CHs*, denominados nodos *hoja* deciden el *clúster* al que pertenecerán. Esta decisión se basa en la potencia con la que ha llegado el mensaje de advertencia ya que a mayor potencia recibida, mayor cercanía de la fuente que emite dicha señal. Los nodos *hoja* informan al *CH* apropiado que les gustaría formar parte de su *clúster*. Los mensajes involucrados en el proceso, junto con su situación en el proceso global, se pueden ver en la figura 2.12.

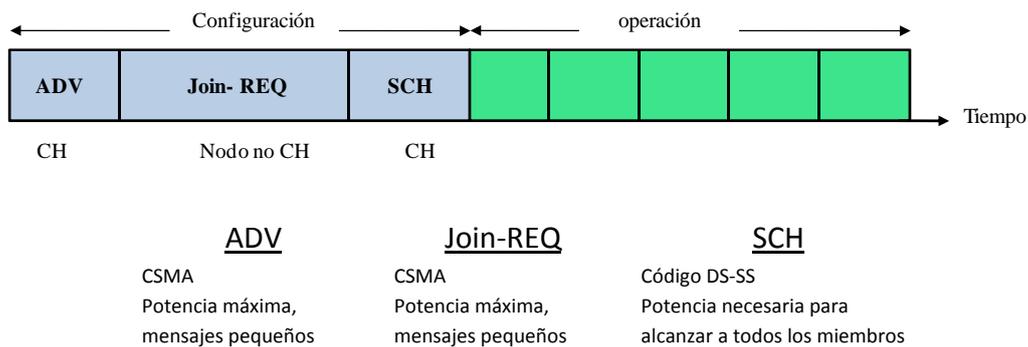


Figura 2.12. Mensajes involucrados en el proceso de asociación al clúster

A modo de ejemplo, en la figura 2.13 se aprecia cómo se forman y agrupan los *clústeres*:

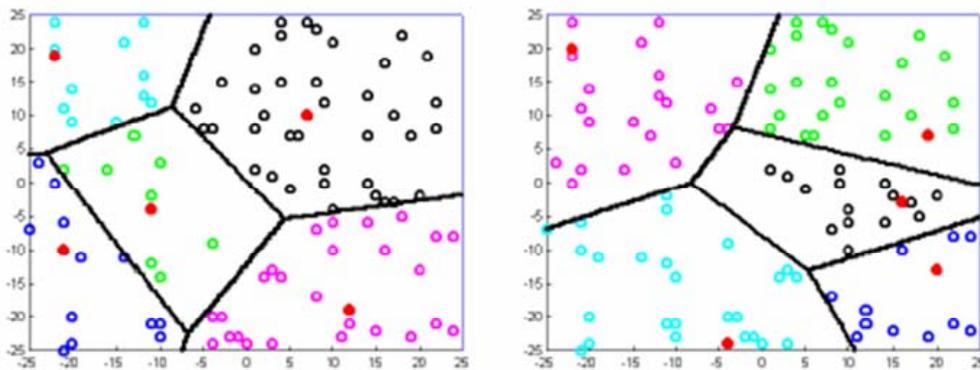


Figura 2.13. Distribución de los clústeres en la red. Se aprecia los CHs (rojo) y los nodos hoja junto con sus regiones de acción

Los autores, basados en su modelo de simulación, concluyeron que la población adecuada de CH es un 5% de los nodos que forman la red.

Después de recibir todos los mensajes de petición de los nodos que desean formar parte del *clúster* y basándose en el número de nodos que lo forman, el *CH* crea una planificación y asigna un *slot* de tiempo a cada nodo hoja del *clúster* para transmitir sus datos (figura 2.14). Esta planificación se transmite a los demás nodos hoja mediante mensajes de difusión.

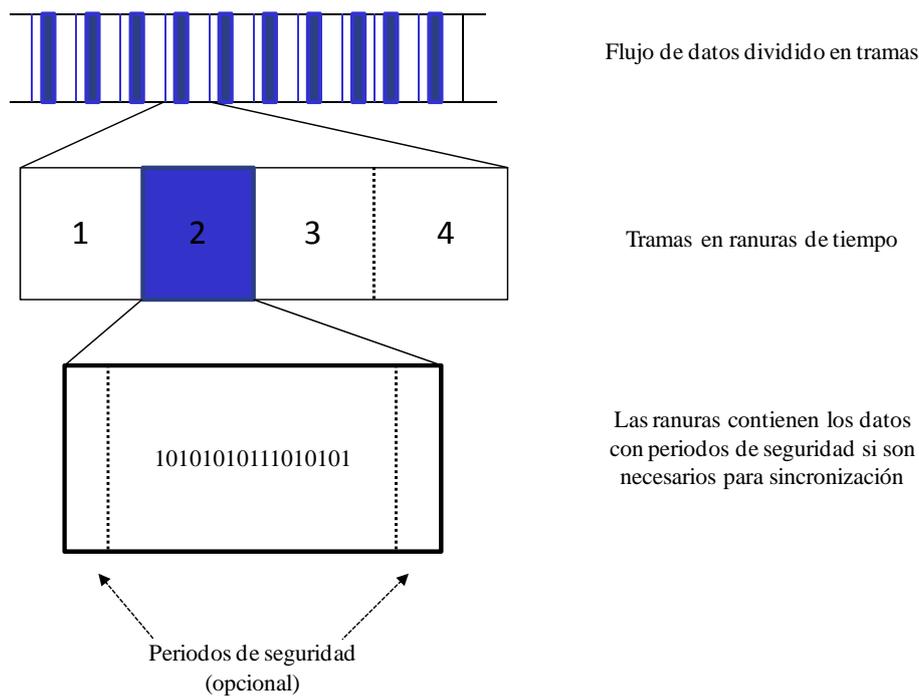


Figura 2.14. Estructura del *frame* TDMA donde se aprecia un flujo de datos dividido en *frames* y estos *frames* divididos en slots de tiempo

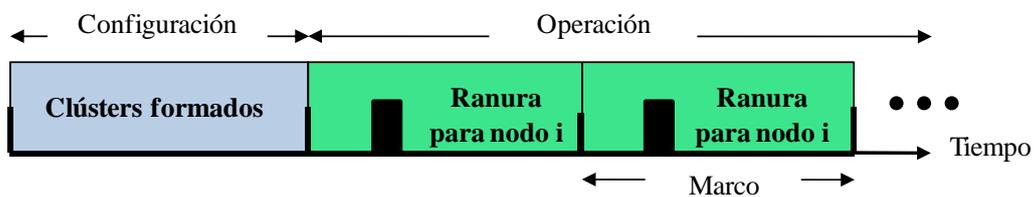


Figura 2.15. Asignación de ranuras dentro del marco a los nodos en la fase de operación

Durante la fase estable o de operación, los nodos pueden empezar a sensorizar e ir transmitiendo la información a sus respectivos *CHs* (figura 2.15). El nodo *CH*, después de recibir la información de todos los nodos hoja del *clúster*, la agrega y la envía al sumidero usando *single-hop* (transmisión directa al sumidero), opción muy criticable en muchos aspectos. Cada *clúster* se comunica usando diferentes códigos *CDMA* para reducir las interferencias procedentes de otros *clústeres*. Después de un cierto período de tiempo, la red vuelve a la fase de configuración de nuevo y se entra en una nueva ronda de selección de *CH*.

Este proceso se puede apreciar en la *figura 2.16*, correspondiente a la evolución temporal (*timeline*) de las operaciones en *LEACH*.

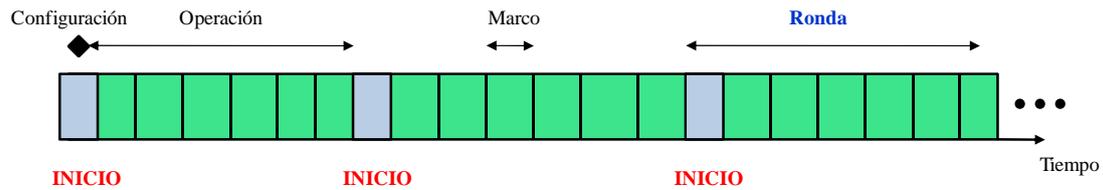


Figura 2.16. Evolución temporal operacional de LEACH donde se aprecia como una ronda se divide en *frames*, que conforman una pequeña parte la fase de configuración y una mayor duración la fase de operación

*LEACH* instaura una idea novedosa al turnar el rol de *CH* entre la totalidad de sensores para **distribuir la carga** energética que debe soportar cada uno. La forma en la que se divide el tiempo de operación del algoritmo en rondas y *frames* permite balancear el consumo de los sensores. Es por ello que *LEACH* consigue mejoras muy sustanciales frente a otros esquemas, como se aprecia en la *figura 2.17*:

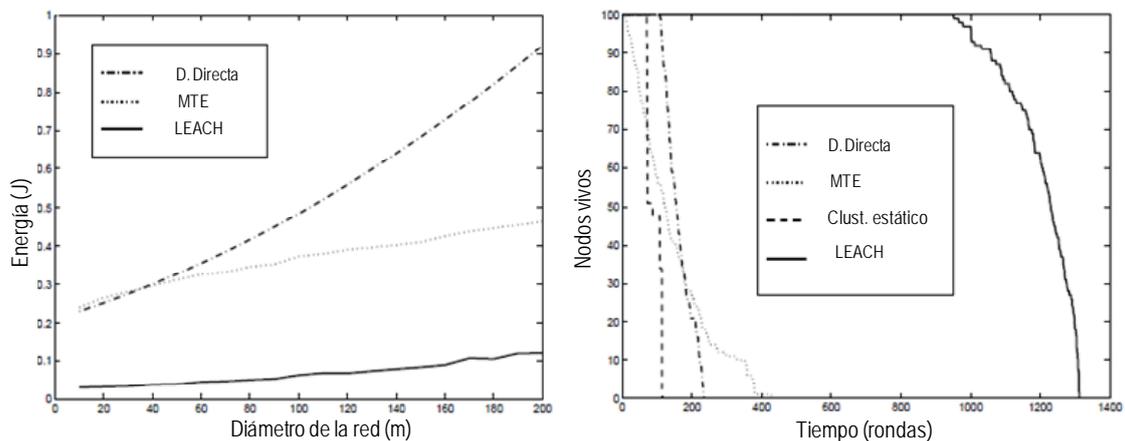


Figura 2.17. Energía total disipada (izquierda) y número de nodos que siguen vivos (derecha) en función del número de rondas para una red con  $N = 100$  nodos, distribuidos de forma aleatoria

Aunque *LEACH* es capaz de incrementar sustancialmente la vida de la red, existen ciertos **aspectos problemáticos** relacionados con las asunciones iniciales empleadas por el algoritmo:

- *LEACH* asume que todos los nodos pueden transmitir con suficiente potencia como para alcanzar el sumidero si fuera necesario y que cada nodo tiene potencia computacional como para soportar diferentes protocolos *MAC*. Por tanto, no es aplicable a redes desplegadas sobre grandes regiones, e introduce una complejidad mínima del nodo. El considerar aunque sea de lejos que el sensor deba transmitir directamente al sumidero con el gasto energético que ello conlleva, es muy cuestionable desde el punto de vista de la eficiencia energética.

- También se asume que los nodos siempre tienen datos que enviar y que nodos situados cerca poseen información correlacionada. Esto no es así en todos los casos aunque si es una suposición viable.
- No es trivial garantizar que los *CHs* predeterminados van a estar distribuidos uniformemente por la red. De hecho, existe la posibilidad de que los *CHs* electos estén concentrados únicamente en una zona de la red por lo que algunos nodos no tendrán ningún *CH* en sus proximidades y no podrán asociarse a ningún *clúster* existente, provocando una reducción sustancial en la productividad y el tiempo de vida de la red. La inclusión de mecanismos para evitar nodos huérfanos debe ser **primordial**, así como el distribuir de manera eficiente y equilibrada los *CHs* a lo largo de la red.
- La idea de *clustering* dinámico trae consigo una **carga** extra en el consumo debido a los cambios en los *CHs*, advertencias de unión etc., las cuales disminuirán la ganancia adquirida en el consumo energético. Esto es salvable en el caso en el que se incorporen mecanismos de control en la rotación de los roles para conservar la distribución eficiente del consumo.
- Debido al carácter jerárquico que posee el algoritmo y su organización en *clústeres*, la **escalabilidad** es media-alta, siempre y cuando se tenga en cuenta la importancia de mantener controlada la sobrecarga en las iteraciones de las rondas del protocolo.
- El propio control de la sobrecarga y su escalabilidad hace que las **latencias** estén en cierta medida acotadas, luego se obtiene la capacidad de proporcionar **QoS**.
- Finalmente, el protocolo asume que todos los nodos comienzan cada nueva ronda de selección con la misma cantidad de energía, asumiendo así que ser un *CH* consume aproximadamente la misma cantidad de energía para cada nodo. El protocolo debería extenderse para tener en cuenta capacidades energéticas no uniformes en los nodos e incorporar estrategias para que el *gap* energético no afecte a las operaciones del protocolo.

### 2.3.2 Low Energy Adaptive Clustering Hierarchy-Centralized (LEACH-C)

Aun habiendo ventajas usando el algoritmo distribuido de formación de *clústeres* de *LEACH*, éste no ofrece garantías sobre la **localización y/o cantidad de CHs**. Usar un algoritmo centralizado para la formación de los *clústeres* puede dar lugar a mejores resultados, dispersando los *CHs* por la red. Esta es la idea básica de *LEACH-C* [HEINZEL02], protocolo que emplea un algoritmo centralizado de *clustering* en conjunción con la misma fase constante de *LEACH*.

Durante la fase de configuración de *LEACH-C*, cada nodo envía información sobre su posición actual, pre-programada en el mismo u obtenida mediante técnicas de posicionamiento (posiblemente mediante posicionamiento global *GPS*), y sobre sus niveles de energía al

sumidero. Además de determinar una buena elección de los *clústeres*, el sumidero necesita asegurarse de que la carga está igualmente bien distribuida por la totalidad de la red. Con este fin, el sumidero computa la energía media de los nodos y, los nodos que estén por debajo de este valor, no podrán ser *CHs* en la ronda actual. Empleando los restantes nodos como posibles *CHs*, el sumidero encuentra los *clústeres* ejecutando un algoritmo [MURATA94] que resuelve el problema *NP-completo* de encontrar los *k-clústeres* óptimos. Este algoritmo intenta minimizar la cantidad de energía de los nodos hoja para transmitir sus datos al *CH*, minimizando la suma total de las distancias al cuadrado entre los nodos hoja y el *CH* más cercano.

Una vez que se han establecido los *CHs* y los nodos asociados a ellos, el sumidero realiza una difusión de un mensaje que contiene el identificador *ID* del *CH* para cada nodo de la red. Si el *ID* del *CH* de un nodo concuerda con su propio *ID*, el nodo obtiene el rol de *CH*; es entonces cuando determina su slot *TDMA* para la transmisión de datos y pasa al estado de *sleep* hasta que llega dicho momento. La fase estable de comunicación de *LEACH-C* es idéntica a la de *LEACH*.

Para la realización de las simulaciones y posterior comparación con otros protocolos, los autores emplean el simulador *NS* [NS09]. Se compara *LEACH* con *LEACH-C*, encaminamiento por rutas de mínima energía (energy aware routing ó *minimum-transmission-energy* - *MTE*) y *clustering* estático en términos de tiempo de vida de la red, disipación de la energía, cantidad de datos transmitidos y latencia. Los resultados obtenidos y su análisis se muestran en la figura 2.18.

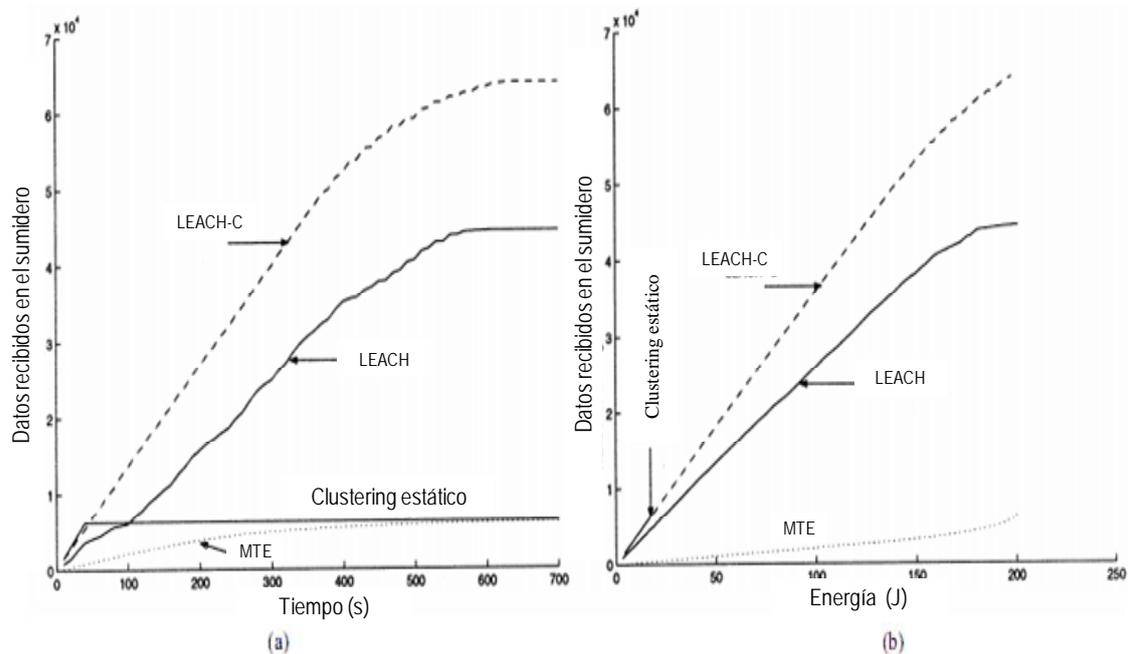


Figura 2.18. (a) Cantidad total de información recibida en el sumidero en función del tiempo y (b) cantidad total de información recibida en el sumidero en función de la energía. Cada nodo comienza con 2 J de energía

En lo que a cantidad de datos transmitidos, se observa cómo *LEACH* distribuye un orden de magnitud más datos por unidad de energía que *MTE*, *LEACH-C* consigue una entrega de datos por unidad de energía un 40% superior que *LEACH*. Las técnicas de *clustering* estático no consiguen un buen rendimiento cuando los nodos tienen restricciones de energía severas. En el caso de *MTE*, la razón se basa en que cada mensaje tiene que pasar por diversos saltos hasta llegar al sumidero. En los demás protocolos, cada mensaje se transmite únicamente en un salto hasta el *CH*, donde se realiza agregación y posteriormente el envío al sumidero, con el consecuente ahorro energético.

En el gráfico *b* se denota claramente que *LEACH* y *LEACH-C* alcanzan ambos la mayor eficiencia energética en entrega de información, en comparación con *clustering* estático o *MTE*, los cuales no realizan computación local, con el fin de reducir la cantidad de datos a enviar, en forma de agregación ni compresión.

Respecto a la comparación de *LEACH* con *LEACH-C*, se ve claramente que *LEACH* no es tan eficiente como *LEACH-C*. Esto es debido a que en *LEACH-C* la formación de clústeres se realiza basándose en el conocimiento global de la **posición** y **energía** de todos los nodos de la red, y por tanto optimiza los mismos de forma que requieran menos energía para realizar sus transmisiones.

En lo que a tiempo de vida se refiere, la *figura 2.19 (a)* muestra el número de nodos que quedan vivos en función del tiempo de simulación. Se observa como en *MTE* los nodos se mantienen vivos largos periodos, pero ello se debe a que se envía al sumidero una cantidad sustancialmente menor de datos, no porque el protocolo posea eficiencia energética. Si se representan el número total de nodos que quedan vivos en función de la información recibida por el sumidero (*b*), vemos que los nodos en *LEACH* pueden entregar 10 veces más efectivamente los datos que en *MTE*, para el mismo número de nodos agotados.

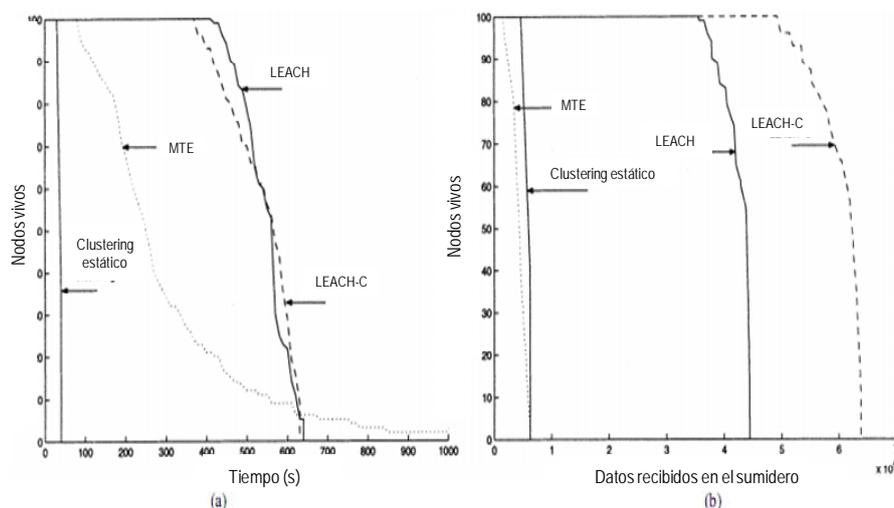


Figura 2.19. (a) Número de nodos vivos en función del tiempo y (b) número de nodos vivos en función de la cantidad de datos entregada al sumidero. Cada nodo comienza con 2 J de energía

Los métodos basados en *clustering* estático consiguen unos resultados pobres debido, fundamentalmente, a que los nodos que actúan de *CH* agotan rápidamente su energía, haciendo por tanto inaccesibles a todos los nodos que pertenecían a su *clúster*.

Los autores, pues, demuestran la potencia de LEACH-C en comparación a anteriores estrategias, debido principalmente a la idea de traspasar la sobrecarga de la formación de los clústeres al sumidero, la cual no tiene ni de lejos las mismas restricciones de energía que los nodos de la red, es un buen enfoque, centralizando esa fase del algoritmo y liberando a los sensores de dicha tarea. La sobrecarga causada por los mensajes asociados al proceso de formación de clústeres compensa sobradamente al optimizar la formación pseudoaleatoria de LEACH, puesto que el conocer la posición exacta de los nodos y su energía restante supone que se pueden tomar decisiones óptimas en el proceso.

Aun así, LEACH-C posee diversos inconvenientes. Partiendo de la base que las asunciones que se hacían en LEACH, se siguen haciendo en LEACH-C, se postulan nuevas asunciones:

- La estación base y los nodos dependen de la existencia de **mecanismos de localización (GPS)** para que el algoritmo de formación de clústeres de *LEACH-C* tenga éxito. Esto conlleva una nueva restricción ya que no siempre es posible dotar a los sensores de estos mecanismos, ya sea por cuestiones de **ahorro energético** o de producción, o bien complica extraordinariamente la fase de despliegue de la red.
- Los nodos deben **enviar en cada ronda** su información de posicionamiento y de energía al sumidero, creando una **sobrecarga** adicional en el protocolo debido a su centralización, y que puede traer nuevos problemas de **consumo** y **latencia** en las decisiones.
- Aun teniendo en cuenta que la centralización del protocolo en la fase de creación de *clústeres* es una estrategia que libera a los nodos de tomar decisiones *sub-óptimas* sobre sus *clústeres* y su agrupamiento, se echan en falta análisis empíricos que demuestren que este enfoque obtiene mejores resultados de consumo que algoritmos distribuidos. No queda patente si el beneficio de un algoritmo de estas características supera las desventajas en cuanto a **sobrecarga** en los mensajes de advertencia requeridos por el algoritmo para que el sumidero tome decisiones. Esta sobrecarga puede resultar muy superior si se considera que los nodos no pueden acceder directamente a la sumidero.

### 2.3.3 Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

La idea principal en *PEGASIS* [LINDSEY02] es formar una **cadena de sensores** donde cada nodo recibirá y transmitirá a un vecino cercano. La formación de dicha cadena puede ser una tarea realizada por los propios nodos usando un algoritmo voraz o, por el contrario, el sumidero puede computarla y luego distribuirla mediante *broadcast* a todos los sensores. La información recogida se transferirá de nodo en nodo por la cadena, se fusionará y eventualmente, un sensor designado previamente, se encargará de transmitirla hacia el sumidero (véase figura

2.20). Los nodos se turnan para transmitir al sumidero luego la energía media disipada por ronda en cada nodo se reduce.

Por lo tanto tenemos dos objetivos: primero, incrementar la vida de la red al máximo usando técnicas de colaboración y, segundo, permitir únicamente coordinación local entre nodos que próximos, por lo que el ancho de banda en términos de comunicación se reduce. Al contrario que en *LEACH*, *PEGASIS* no realiza formación de clúster alguna y solo usa un nodo en la cadena para transmitir al sumidero en vez de usar múltiples nodos.

$$c0 \longrightarrow c1 \longrightarrow c2 \longleftarrow c3 \longleftarrow c4$$


Sumidero

Figura 2.20. Encadenamiento en PEGASIS

Para localizar el vecino más próximo en *PEGASIS*, cada nodo emplea la potencia de la señal para estimar la distancia a cada uno de sus nodos vecinos y posteriormente ajusta la potencia de dicha señal, de modo que únicamente un nodo pueda escucharle. La cadena en *PEGASIS* está formada por aquellos nodos que están más cerca los unos de los otros y forman un camino hasta el sumidero. La información agregada será enviada hasta el sumidero por un nodo en la cadena y dichos nodos se turnarán en el envío. Como ya se dijo anteriormente, la formación de esta cadena se realiza de forma voraz debido a que los autores dan por buena esta aproximación, la cual se realiza antes de que la primera ronda de comunicación empiece.

En la construcción de la cadena se comienza por los nodos más alejados al sumidero, para asegurarse de que dichos nodos lejanos tengan vecinos próximos, ya que en el algoritmo voraz la distancia a los vecinos irá aumentando gradualmente a medida que se construye la cadena y los sensores que ya se encuentren en la cadena no podrán ser revisitados.

A primera vista, la construcción de estas cadenas tiene las siguientes **desventajas**:

- Construir una cadena minimizando la longitud total es similar al típico problema del “viajante de comercio”. Este problema pertenece al conjunto de problemas llamado problemas *NP-completos*, los cuales no poseen algoritmo alguno que aporte una solución eficiente. De hecho, se asume que el peor caso respecto al tiempo para un algoritmo de estas características depende exponencialmente del número de nodos. Por lo tanto, el planteamiento de la construcción de una cadena, aunque se suavice el problema mediante un algoritmo voraz, no es ni mucho menos óptimo, por lo que la solución no es de carácter apropiado.

- Debido a que la solución adoptada por el algoritmo depende exponencialmente del número de nodos, a altas cargas se produce un problema de **latencia** que se agrava con las longitudes de las cadenas ya que, ante cientos de nodos, éstas son de gran longitud.
- La **escalabilidad** de *PEGASIS* queda entonces anulada o muy disminuida precisamente por lo comentado en el punto anterior, ya que aun habiendo un número moderado de nodos, no se observa una clara tendencia en la escalabilidad.
- Puede ocurrir que algunos nodos tengan vecinos que se encuentren relativamente lejos dentro de la cadena. Estos nodos disiparán más energía en cada ronda en comparación con los demás nodos. Existe un mecanismo en *PEGASIS* que evita la elección de estos nodos como líderes de la cadena, instaurando umbrales de distancia, por debajo de los cuales un nodo no puede ser líder. Sin embargo, cuando un nodo muere, la cadena se reconstruye y el umbral se configura de nuevo para determinar qué nodos pueden o no ser líderes. Debido a que no existe mecanismo alguno que incida en la distribución del consumo de los nodos, esta reconstrucción de las cadenas añade una **sobrecarga** extra.

Las simulaciones arrojan que *PEGASIS* es capaz de incrementar la vida de la red el doble que el protocolo *LEACH*. Esta ganancia en rendimiento se debe a la eliminación de la sobrecarga por la formación de *clústeres* que impone *LEACH* y por la reducción de transmisiones de información y recepción a causa de la agregación. Aunque se evite la sobrecarga por parte del *clustering*, *PEGASIS* sigue necesitando ajustes dinámicos en la topología debido a que los nodos necesitan saber acerca del estado energético de sus vecinos con el fin de saber dónde dirigir los datos. Este ajuste en la topología de las cadenas puede introducir una **sobrecarga adicional** considerable, especialmente en redes altamente utilizadas como ya se dijo en párrafos anteriores. Además, *PEGASIS* asume que cada sensor es capaz de comunicarse con el sumidero directamente y que todos los nodos poseen la misma cantidad de energía, al igual que en *LEACH*, luego posiblemente se agoten más o menos al mismo tiempo.

Una extensión de *PEGASIS*, llamada *hierarchical-PEGASIS* [SAVVIDE01], se desarrolla con el objetivo de disminuir el retardo inducido por la longitud de la cadena de transmisión de los paquetes hacia el sumidero. Con este propósito, se estudian enfoques basados en transmisión simultánea de datos con el fin de evitar colisiones. Se propone que únicamente se permita transmitir a la vez a los nodos que estén físicamente separados dentro de las cadenas. Este protocolo basado en cadenas y nodos con capacidad *CDMA*, construye múltiples cadenas de nodos, conformando una especie de árbol jerárquico, donde cada nodo seleccionado en un nivel específico envía información a los nodos del nivel superior de la jerarquía. Un ejemplo de este tipo de jerarquía se muestra a continuación en la figura 2.21.

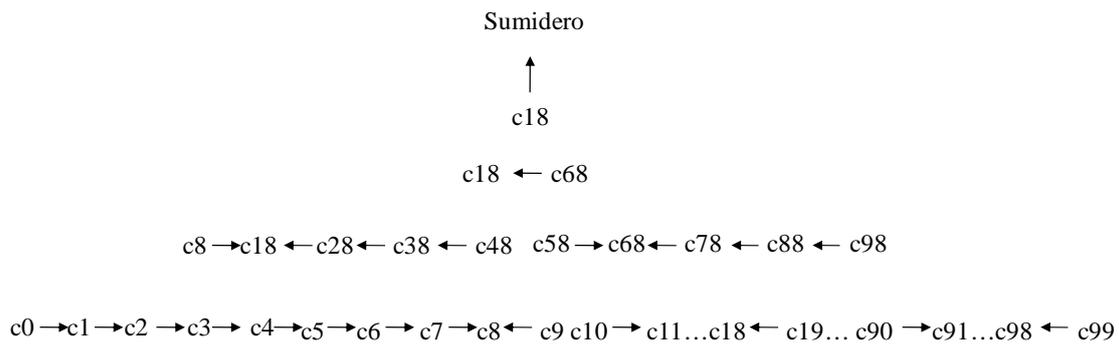


Figura 2.21. Esquema basado en cadenas de 3-niveles para una red de sensores con nodos sin CDMA

Este método se asegura la transmisión en paralelo, reduciendo el retardo. Este tipo de aproximación jerárquica ha demostrado tener hasta 60 veces más rendimiento que PEGASIS normal pero mantiene la mayoría de las desventajas de su predecesor, aunque añadiendo un punto de vista jerárquico con cadenas interesante que, si bien no es óptimo, puede dar lugar a otras aproximaciones jerárquicas en *clustering*, como se ve a continuación en *TEEN* y *APTEEN*.

#### 2.3.4 Threshold-sensitive Energy Efficient Protocols (*TEEN* y *APTEEN*)

Estos protocolos se desarrollaron para aplicaciones críticas donde se precisa de una respuesta rápida frente a repentinos cambios en los parámetros que están siendo sensorizados. En *TEEN* [MANJESH01], los nodos sensorizan el medio continuamente pero la transmisión de la información se hace de una forma menos frecuente. Cada nodo *CH* envía a los miembros de su clúster un umbral superior, que impide la transmisión de valores atributo que se está sensorizando por debajo del mismo, y un umbral inferior, que es el cambio mínimo en el valor del atributo sensorizado que produce la transmisión de un nuevo mensaje. De esta forma, se reduce el número de transmisiones al permitir a los nodos transmitir únicamente cuando el atributo sensorizado se encuentra en el rango de interés (por encima del umbral superior) y la diferencia con la anterior medida transmitida es significativa (mayor que el umbral inferior).

Un valor pequeño en el valor del umbral inferior da una visión más detallada de la red, a costa de un mayor coste en el consumo de energía. Por ello, el usuario puede introducir un equilibrio entre eficiencia energética y precisión en la información.

El modelo de topología creado por este protocolo tiene un aspecto jerárquico y centrado en los datos. Los *clústeres* se construyen de forma jerárquica a diferentes niveles, con *CHs* electos que sirven como enlaces de comunicación entre ellos mismos y el sumidero. La *figura 2.22* muestra la topología de red creada:

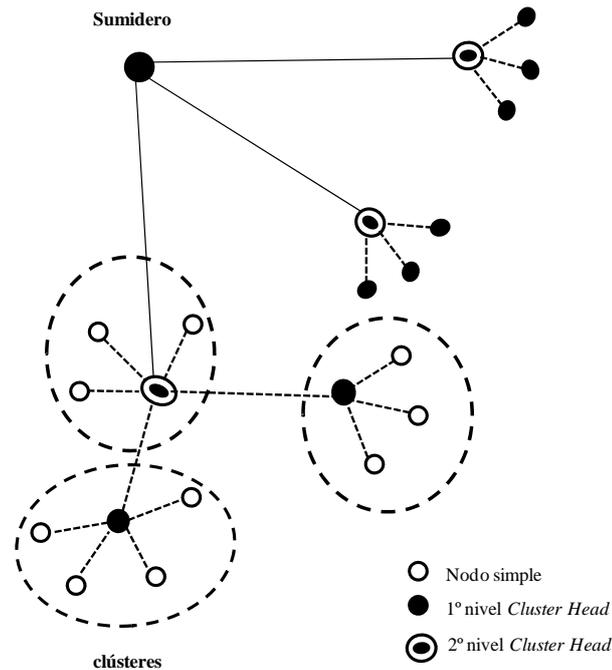


Figura 2.22. *Clustering* jerárquico en TEEN

Este modelo posee características interesantes que dejan entrever desventajas importantes, las cuales se detallan a continuación:

- Los nodos se agrupan formando *clústeres* como en *LEACH*, designando a un representante del *clúster* como único nodo que se comunica con el segundo nivel de *CHs*, donde la comunicación con el sumidero es directa y en *single-hop*. Esta topología en forma de árbol jerárquico presenta las mismas ventajas que *LEACH* pero *TEEN* no contempla mecanismos para la rotación alguna de los roles, sino que confía en otros mecanismos (umbrales) para reducir el consumo. Este enfoque no presenta una gran viabilidad en lo que a dispersión y distribución del consumo se refiere, por lo que además se incorporan problemas de **escalabilidad** cuando se pretende aumentar el número de nodos mínimamente.
- Los *CHs* presentan la mayor carga energética del modelo ya que tienen que soportar el cargo de representación de sus *clústeres* y además realizar envíos de gran potencia al sumidero, ya que desde el segundo nivel de jerarquía, la transmisión al sumidero es en *single-hop* y probablemente de larga distancia. Esta medida es muy discutible ya que se podría disminuir el consumo aplicando *encaminamiento multipath* hasta el sumidero, introduciendo más niveles jerárquicos en el modelo.

*APTEEN* [MANJESH02], por el contrario, es un protocolo híbrido sucesor de *TEEN* que cambia periódicamente los umbrales utilizados en *TEEN* de acuerdo a las necesidades del usuario final y el tipo de aplicación, respondiendo mejor a eventos críticos. Podemos observar los diagramas operacionales de *TEEN* y *APTEEN* en la figura 2.23. En *APTEEN*, los *CHs* transmiten mediante *broadcast* varios parámetros, entre los que destacan los propios umbrales o la

planificación *TDMA* que debe seguir cada nodo. Dicha planificación combina políticas tanto proactivas como reactivas. Este envío mediante *broadcast* introduce una **sobrecarga** que en términos energéticos aumenta el consumo, aunque se permite al usuario controlar los valores de los umbrales con el fin de acotar algo el consumo de energía.

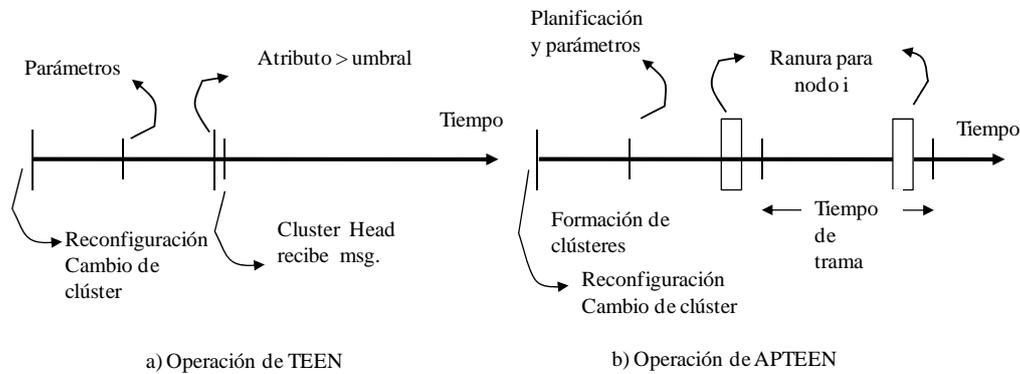


Figura 2.23. Línea operacional para (a) TEEN y (b) APTEEN

Análisis empíricos han demostrado que el rendimiento de *APTEEN* está entre *LEACH* y *TEEN* en términos de energía disipada y tiempo de vida de la red. *TEEN* muestra el mejor rendimiento debido a que decreta el número de transmisiones, todo ello a costa de la implementación de funciones basadas en umbrales las cuales poseen un coste y complejidad **altos**. Ambas aproximaciones poseen deficiencias en cuanto a eficiencia en el mantenimiento de su topología y la dispersión de sus mecanismos de actualización, que si bien incorporan ideas buenas, como la topología jerárquica en dos niveles, no terminan de encontrar una solución óptima ni interesante, aunque implantan estrategias interesantes a seguir para alcanzar una solución de compromiso aceptable.

### 2.3.5 Small Minimum Energy Communication Network (MECN)

El protocolo MECN [RODOPLU99] computa una **subred** de mínima energía para cierta red de sensores, ayudándose de **receptores GPS**. *MECN* identifica una región de acción para cada nodo. Dicha región consiste en un área circundante, a través de la cual transmitir mediante esos nodos es más eficiente energéticamente que Difusión Directa. La idea principal de *MECN* es encontrar dicha subred, que tendrá un número menor de nodos y requerirá menos energía para realizar transmisiones entre dos nodos. De esta forma, se encuentran rutas de mínimo coste energético sin tomar en consideración a todos los nodos de la red. Esto se realiza empleando búsqueda localizada para cada nodo en la región circundante. *MECN* se auto-reconfigura y con ello puede adaptarse automáticamente a fallos en los nodos o redistribución de los mismos.

En *MECN* se asume que cualquier nodo puede transmitir a cualquier otro nodo de la red, lo cual no es siempre posible, ya sea por obstáculos o porque la red no tiene por qué ser totalmente conexas. Por ello, se realiza una extensión a *MECN*, llamada *SMECN* [LI01] (*Small Minimum Energy Communication Energy*) donde se toman en cuenta posibles obstáculos entre

pares de nodos. Sin embargo, la red sigue asumiéndose que está totalmente conectada, como en el caso de *MECN*. La subred construida por *SMECN* es probablemente menor, en cuanto a número de aristas se refiere, lo que ayuda a que *SMECN* envíe mensajes a lo largo de caminos de mínimo consumo energético. En la figura 2.24 se muestra la diferencia de resultado en la construcción de la subred entre *MECN* y *SMECN*:

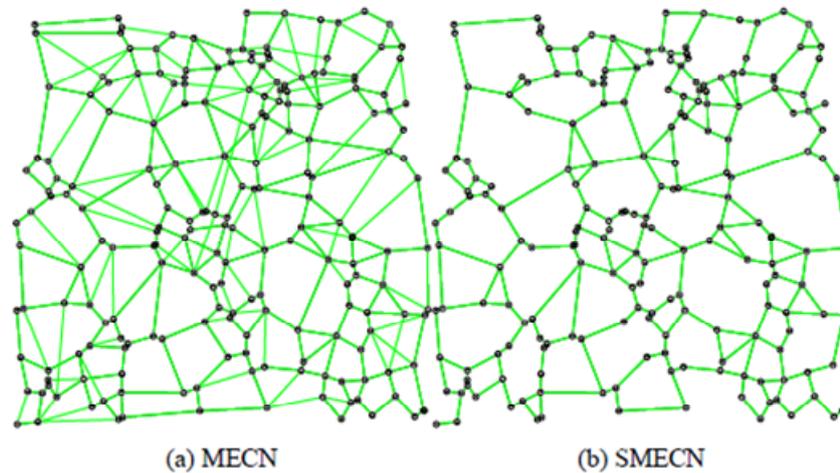


Figura 2.24. Subred inicial computada por (a) MECN y (b) SMECN

Sin embargo, el algoritmo propuesto es local en el sentido de que realmente no encuentra el camino de mínimo consumo energético sino que construye una subred en la que se garantiza que exista dicho camino. Además, la subred construida por *SMECN* hace más probable que la ruta empleada es una de las que requiere menor consumo. Todo ello muestra **desventajas** claras, unidas al hecho principal de que construir una subred con menor número de aristas impone una **sobrecarga** extra al algoritmo.

### 2.3.6 Self Organizing Protocol (SOP)

En SOP [SUBRAMA01], se diseña una arquitectura donde cada nodo posee sus propias capacidades y funcionalidades. Los tipos de nodos presentes en la arquitectura son los siguientes:

1. **Nodos sumidero**, los cuales poseen las características de cualquier computadora, con grandes capacidades de almacenamiento, alta potencia computacional, conexión con red cableada (*Internet*) y sin restricciones de energía alguna.
2. **Sensores especializados**, los cuales sensorizan el medio ya sea para monitorizar parámetros climáticos como temperaturas, presión, humedad, etc. ó para actuar como detectores de movimiento o fotografiar imágenes. En la arquitectura de *SOP*, cada sensor se identifica con una clase y puede comunicarse con los miembros de su clase o con los pertenecientes a otras clases. Estos nodos son perfectamente móviles.

3. **Router sensors**, los cuales actúan como enrutadores o como diseminadores de los datos. Estos nodos se auto-organizan para formar el esqueleto (troncal) de la red de sensores. En la arquitectura *SOP* se asume que estos nodos son estrictamente inmóviles ya que si fueran móviles, la troncal de la red necesitaría estar en continua reorganización, lo que puede ser muy costoso.
4. **Nodos agregadores**, los cuales aportan las funcionalidades de agregación de la información, comprimiéndola y agregándola antes de enviarla a otro nodo. Esta funcionalidad puede ser introducida en los propios nodos enrutadores o se pueden crear nodos especializados que también actúen de esta manera.

La diferenciación de roles en la arquitectura propuesta por *SOP* es una buena opción ya que aplica una idea parecida a la distinción de funciones rotativas de *LEACH*. El empleo de nodos que actúan como *routers*, diferenciándolos de otros sensores que realizarán labores de sensorización, conlleva las siguientes ventajas:

- El proceso de diseminación y enrutamiento de la información está totalmente separado del proceso de sensorización o descubrimiento de la información. En otras palabras, los nodos especializados se encargan de extraer los datos mientras que los *router sensors* son los encargados de distribuir dichos datos.
- Es bien conocido el hecho de que el consumo de energía usando  $N$  saltos cortos es aproximadamente  $N$  veces menor que el consumo energético para 1 salto largo. La presencia de estos *router sensors* hace que la mayoría de los saltos en la red sean cortos en vez de largos, ayudando a reducir el consumo de energía.
- La presencia de una gran cantidad de *router sensors* en la troncal de la red puede incrementar el tiempo de vida y la tolerancia a fallos de la red. El ratio *router sensors-sensores especializados* ( $R/S$ ) depende completamente de la aplicación, pero para nuestro caso seleccionamos un ratio de 1.

En la *figura 2.25* se aprecia los diferentes tipos de sensores presentes en la arquitectura.

El algoritmo que se sigue en el protocolo permite la auto-organización de un conjunto de sensores aleatorios dispuestos en un área concreta. Los *router sensors* se reorganizan ellos mismos en una red para implementar este algoritmo mientras que los nodos especializados únicamente mantienen un seguimiento de los *router sensors* que están vivos y más cercanos a ellos. El algoritmo consta de cuatro fases, las cuales se describen a continuación y en orden estricto:

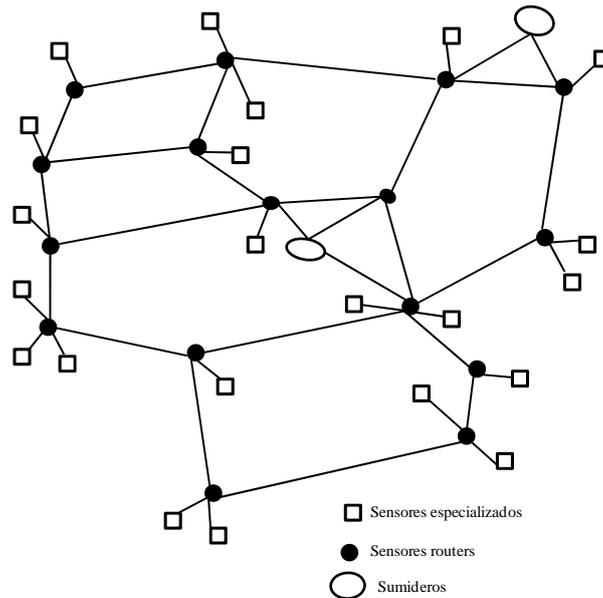


Figura 2.25. Sistema auto-organizativo con sensores routers, nodos especializados y nodos sumidero

1. **Fase de descubrimiento:** cada nodo descubre de forma independiente su conjunto de vecinos en la red y fija su máximo radio de transmisión. Se tiene en cuenta el número de vecinos descubiertos, ya que un excesivo número de ellos provoca problemas de latencia y sobrecarga, y el radio de transmisión, que se ajusta al conjunto de vecinos con el fin de ahorrar energía, ya que el consumo en transmisión es proporcional al cuadrado – al menos – de la distancia entre fuente y destino.
2. **Fase de organización:** durante esta fase la red se organiza y se realizan las siguientes operaciones:
  - Los nodos se asocian en grupos que a su vez vuelven a reunirse para formar grupos más grandes. De esta forma, se construye una jerarquía de grupos en la red y el propio protocolo se asegura de que dicha jerarquía esté balanceada.
  - Cada nodo se asigna una dirección basada en su posición en la jerarquía.
  - Se computa para cada nodo en la red una tabla de enrutamiento de  $\Theta(\log n)$ .
  - Se construye un **árbol** y un **grafo de difusión**, agrupando a todos los nodos de la red. El grafo de difusión posteriormente se convierte en un grafo acíclico dirigido orientado hacia el nodo sumidero en la red. La construcción de estos grafos permite añadir tolerancia a fallos a los nodos, haciendo que varios nodos sean accesibles mediante múltiples rutas. En *WSNs*, los grafos de *broadcast* consumen más energía que los árboles *broadcast* y por ello se convierte en un grafo acíclico dirigido, eliminando posibles bucles. Estas medidas pueden disminuir el consumo energético en gran medida, pero la propia construcción de grafos de *broadcast*, con el gasto que ello conlleva, para convertirlos posteriormente en grafos acíclicos, provoca una sobrecarga innecesaria, ya que existen otras estructuras de datos más sencillas de

construir, como un árbol de grupos, aproximación que los autores no exploran, dejando únicamente la idea de “grupos de sensores”.

3. **Fase de mantenimiento:** durante esta fase se realizan las siguientes labores:
  - En monitorización activa, cada nodo mantiene un seguimiento de su batería y envía un mensaje “*estoy vivo*” (*heartbeat*) a sus vecinos cada 30 segundos. En monitorización pasiva, cada nodo manda dicho mensaje únicamente por demanda. El empleo de *heartbits* en este tipo de protocolos donde la sobrecarga y el consumo son primordiales, parece una medida errónea. La elección del modo pasivo debe ser mayoritaria o incluso incorporar mecanismos de tolerancia a fallos no basados en envío constante de mensajes. Puede ser una buena idea para conseguir dicha tolerancia a fallos sin necesidad de añadir carga extra de mensajes.
  - Cada nodo actualiza constantemente su tabla de enrutamiento sobre el siguiente salto en el camino de menor energía y sobre el camino de menor retardo, informando a sus vecinos sobre su métrica actual. A primera vista, actualizar constantemente las tablas para mantener una información real en el tiempo, incrementa el consumo del propio nodo y aumenta la sobrecarga de mensajes debido a las advertencias a sus vecinos. Todas estas actualizaciones vienen dadas por el hecho de que el algoritmo implementa la idea de mandar los mensajes por los caminos de menor coste y menor latencia, estrategia que no siempre es óptima y que termina por drenar mayoritariamente la batería del conjunto de nodos pertenecientes a dicho camino, sin contemplar la rotación entre caminos viables.
  - Los nodos también informan a sus vecinos sobre sus tablas de enrutamiento y sus niveles de energía. Los problemas de sobrecarga son los mismos que en el caso anterior.
  
4. **Fase de auto-reorganización:** en esta fase, los nodos que detecten una partición de algún grupo ó isla, o el fallo de un nodo, cambiarán sus tablas de enrutamiento en función de la nueva topología de red. Si un nodo constata el fallo de todos sus vecinos, repetirá la fase de descubrimiento. Si un grupo falla, los subgrupos se reorganizarán para unirse con nuevos grupos, para conseguir el mayor balanceo de la red posible. Esta fase es muy cara en términos energéticos así que debe optimizarse y evitarse lo máximo posible.

Se ha constatado que la energía consumida por la difusión de mensajes es menor que la consumida en el protocolo *SPIN*. Sin embargo, el protocolo no se puede considerar *on-demand* y menos aún en la fase de organización de la jerarquía, la cual introduce una **sobrecarga extra** que puede ser muy costosa en términos energéticos. La idea básica de organización en jerarquía de los nodos, aplicando diferentes roles a los sensores y agrupándolos para gestionar el enrutamiento y la sensorización es una excelente estrategia para obtener un protocolo eficientemente energético. A pesar de ello, no es posible obviar los problemas de sobrecarga,

tanto de mensajes como de actualizaciones de las tablas en los que incurre el algoritmo, ya que pueden provocar problemas de **latencia**. Si bien este protocolo sienta las bases para futuros trabajos con determinados aspectos clave, no termina de alcanzar un buen esquema de operaciones en los nodos, aplicando métricas y estrategias para la transmisión y formación de la jerarquía que se consideran erróneas.

### 2.3.7 Sensor Aggregates Routing

El propósito de esta aproximación es monitorizar conjuntamente la actividad de un objetivo en un cierto medio [FANG03]. Un conjunto agregado de sensores comprende aquellos nodos en la red que satisfacen un predicado para una tarea de procesado en conjunto. Los nodos se dividen en *clústeres* de acuerdo a la potencia de su señal. Es entonces cuando se eligen los *CH*.

En la *figura 2.26* se observan los picos de señal y diagramas de *Voronoi*, correspondientes a la topología seleccionada por los autores.

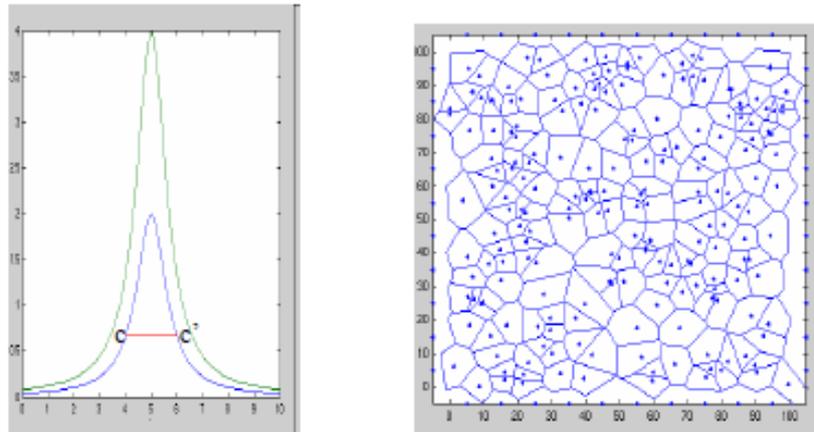


Figura 2.26. Dos fuentes de señal colocadas en un mismo punto (izquierda) y celdas de *Voronoi* para  $N = 200$  nodos aleatoriamente distribuidos (fuente [FANG03])

### 2.3.8 Virtual Grid Architecture Routing (VGA)

Se trata nuevamente de un protocolo que emplea agregación de la información y procesamiento dentro de la red para aumentar el tiempo de vida de la misma [ALKARAK04]. Debido al hecho de que los nodos son estáticos o de movilidad reducida (cierto en la mayoría de las aplicaciones de redes inalámbricas de sensores), una asunción razonable consiste en componer una topología fija. Se suelen emplear *clústeres* cuadrados para obtener una topología virtual rectilínea. Dentro de cada zona, se selecciona un *CH*. La agregación se realiza en dos niveles; primero localmente y luego globalmente. En el caso que nos ocupa, al conjunto de los *CHs* se le denomina *Local Aggregators (LAs)* y se utilizan para realizar la agregación local mientras que un subconjunto de éstos, llamados *Master Aggregators (MAs)*, la realizan a nivel global. A modo de ejemplo, en la *figura 2.27* se muestran dichos *LAs* y *MAs* en una aplicación militar.

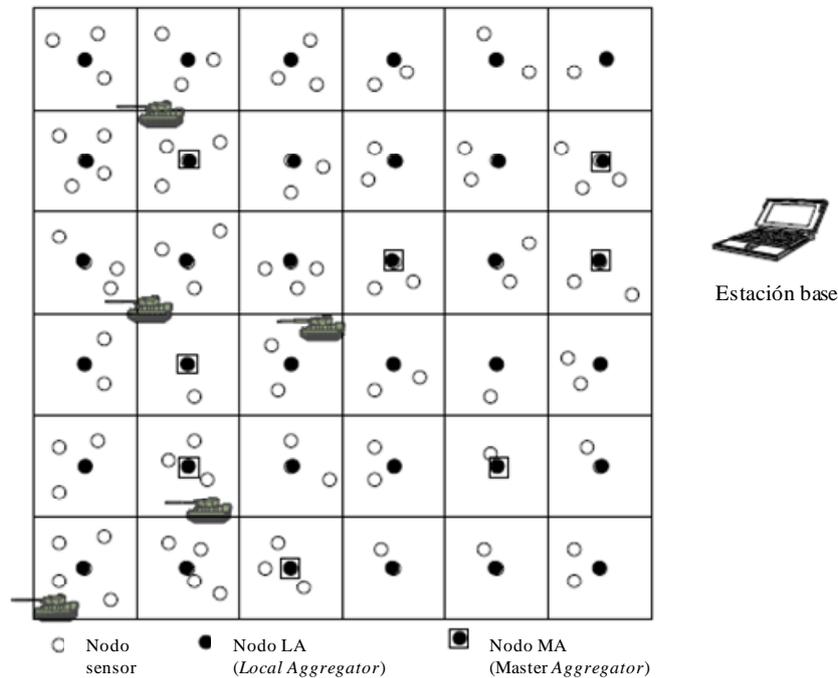


Figura 2.27. Distribución regular con la localización de LAs. Un subconjunto de éstos, los MAs, son seleccionados de manera subóptima (fuente [ALKARAK04])

Hay que destacar que la determinación subóptima del conjunto de nodos *MAs*, se realiza de esta manera por el mero hecho de que este problema corresponde al conjunto de problemas *NP-completos* y, por lo tanto, no existe algoritmo alguno que aporte la solución de manera eficiente.

### 2.3.9 Hierarchical Power-aware Routing (HPAR)

*HPAR* [QLI01] divide la red en grupos de sensores. Cada grupo geográficamente cercano se configura como un *clúster* zonal y cada uno de éstos se trata como una entidad. Para labores de encaminamiento, a cada zona se le permite decidir cómo va a encaminar el mensaje jerárquicamente a través de las demás zonas para conseguir minimizar el consumo de energía. Los mensajes son encaminados a través de la ruta que presenta la máxima reserva de energía de entre todos los caminos de mínima energía restante; a este camino se le denomina ruta *max-min*.

La motivación es que usar los nodos con alta energía residual puede ser más caro en comparación con el camino de menor consumo energético. Para conseguir esto, los autores desarrollan un algoritmo llamado *max-min zPmin*, el cual tiene como idea principal el balanceo entre la minimización del consumo total de energía y la maximización de la mínima energía

residual de la red. Por ello, el algoritmo trata de mejorar una ruta *max-min*, limitando su consumo de energía.

Mediante el empleo del algoritmo de *Dijkstra* y relajando restricciones, se puede conseguir que el algoritmo consuma como mucho *zPmin*, aunque a costa de incorporar **sobrecarga** en los mensajes del propio algoritmo y la consecuente **latencia**.

### 2.3.10 Two-Tier Data Dissemination (TTDD)

*TTDD* [YE02] proporciona una entrega eficiente y escalable a múltiples sumideros móviles. Cada fuente de información construye proactivamente una malla que utiliza para diseminar y recibir continuamente su información a los sumideros mediante consultas por *inundación*. En la figura 2.28 se puede ver la disposición de los nodos en la malla.

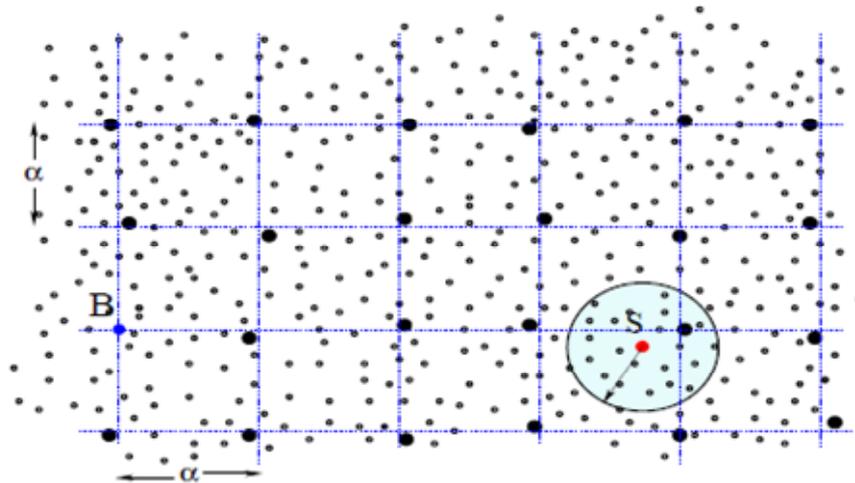


Figura 2.28. Disposición nodos en TTDD. Nodo fuente B. Sumidero S

Se asume que la posición de los demás nodos es conocida, cosa que no siempre es viable, por lo que la aplicabilidad del protocolo queda reducida. Para la retransmisión de la información entre la totalidad de los nodos, los cuales son de carácter homogéneo, se emplea *encaminamiento multi-salto*.

El encaminamiento en el protocolo se estructura como sigue:

- Cuando el sumidero necesita información, realiza una consulta mediante *inundación* en un área local del tamaño de una celda con el fin de descubrir nodos diseminadores cercanos. El sumidero especifica una distancia máxima de la consulta para que dicha consulta cese de distribuirse por los nodos que están a cierta distancia de ella, limitando en cierta manera la **sobrecarga** de la inundación, aunque **no** eliminándola.
- Una vez que la consulta alcanza un nodo diseminador, denominado **nodo inmediato diseminador** para el sumidero, se retransmite por la malla al nodo diseminador superior, el cual recibe mensajes del anterior.

- El diseminador superior continúa retransmitiendo de la misma forma la consulta hasta que finalmente alcanza la fuente. Durante todo este proceso cada uno de los nodos diseminadores ha ido almacenando la información de los nodos diseminadores previos por los que ha pasado la consulta. Esta información de estado se emplea para dirigir la consulta de vuelta al sumidero. En la *figura 2.29* se puede ver este proceso.

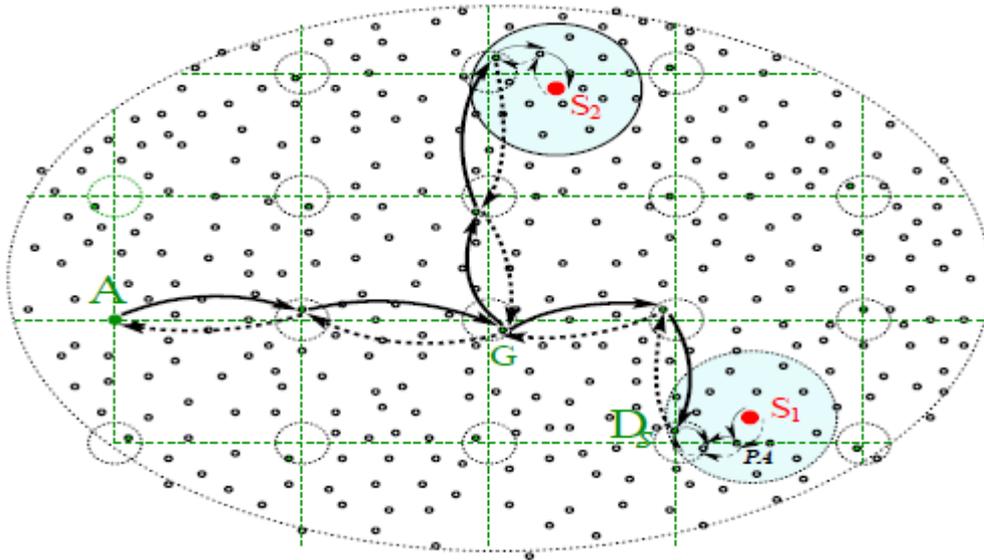


Figura 2.29. Consulta *Two-Tier* y posterior retransmisión entre la fuente A y los sumideros S1 y S2. El sumidero S1 comienza con el *flooding* de su consulta a su agente primario PA, a su nodo inmediato diseminador Ds. Ds la reenvía a diseminadores superiores hasta que llega a A. Los datos son devueltos a Ds por el mismo camino que atravesó la consulta.

Aunque *TTDD* es una aproximación relativamente eficiente, existen varias cuestiones a mejorar respecto a cómo el algoritmo obtiene la información local, imprescindible para configurar la estructura en malla:

- La longitud del camino de retransmisión es mayor que la longitud del camino más corto. Los autores de *TTDD* creen que la suboptimalidad en la longitud del camino merece la pena en comparación con la ganancia en la escalabilidad.
- La anteriormente citada escalabilidad queda totalmente anulada cuando se emplea *inundación* como técnica de retransmisión, ya que se elimina toda eficiencia alcanzada y se añaden problemas de congestión, con la correspondiente pérdida de paquetes y la desaparición de cualquier medida que beneficie a la *QoS*.
- Comparando los resultados entre *TTDD* y Difusión Directa, *TTDD* puede alcanzar mayores tiempos de vida en la red. Sin embargo, la **sobrecarga asociada con el mantenimiento** y recalcado de la malla a medida que la topología de red cambia, puede ser alta.

- Además, *TTDD* asume la incorporación de un sistema de posicionamiento muy exacto, lo cual no es energéticamente viable en *WSNs*, sobre todo comparado con otras estrategias que aún no usando *GPS* consiguen soluciones eficientes.

### 2.3.11 Energy-efficient Distributed Dynamic Diffusion routing algorithm (*e3D*)

*e3D* [RAICU04] pretende aumentar la vida de la red distribuyendo de la mejor manera posible la carga energética entre la totalidad de los sensores. Cualquier protocolo de comunicación que implique sincronización entre los nodos incurre en una **sobrecarga extra** debido a la inicialización en la comunicación. Con este protocolo se pretende determinar, comparándolo con otras estrategias conocidas, si los beneficios de un algoritmo más complejo eclipsan la sobrecarga inducida por la sincronización requerida por cada nodo para comunicarse. Cada nodo podría tener conocimiento total sobre la topología de la red y la energía de cada nodo y con ello tomaría las decisiones óptimas sobre qué hacer para realizar las transmisiones. Esta idea estaría en el buen camino si no fuera por la alta sobrecarga que conllevarían los mensajes de sincronización.

Para extraer conclusiones acerca de estos problemas, se compara *e3D* con tres tipos de algoritmos:

1. **Comunicación Directa**, donde se asume que cada nodo está en rango de comunicación con el sumidero. La mayor ventaja de este algoritmo es su simplicidad pero también su mayor desventaja es que el consumo de la comunicación vía radio aumenta en función del cuadrado de la distancia, por lo que los nodos lejanos consumirán más energía que los cercanos a la estación base. Obviamente, queda claro que transmitir a través de pequeños salto es mejor que hacerlo con un gran salto.
2. **Algoritmo basado en Difusión**, donde, como ya se vio en el caso de Difusión Directa, la principal preocupación es que no se realiza de manera totalmente eficiente la distribución del consumo energético por la red y por ello los nodos cercanos al sumidero agotan su energía antes que los lejanos, debido a que éstos son los últimos nodos involucrados en el proceso de difusión hasta el sumidero.
3. **Clústeres aleatorios**, similar a *LEACH*. La principal ventaja de este algoritmo es la distribución en la disipación de energía alcanzada mediante la elección aleatoria de los *CHs*, consiguiendo de esta forma una distribución aleatoria de los nodos que se quedan sin energía. Por el contrario, la desventaja primordial es su relativamente alta complejidad, con bastantes mensajes de sincronización, en comparación con *e3D*.

Un aspecto importante es obtener visualmente datos directos sobre el tiempo de vida global de la red. En la *figura 2.30*, se presentan gráficos donde se puede ver la energía restante para cada nodo de la red para los protocolos anteriormente señalados.

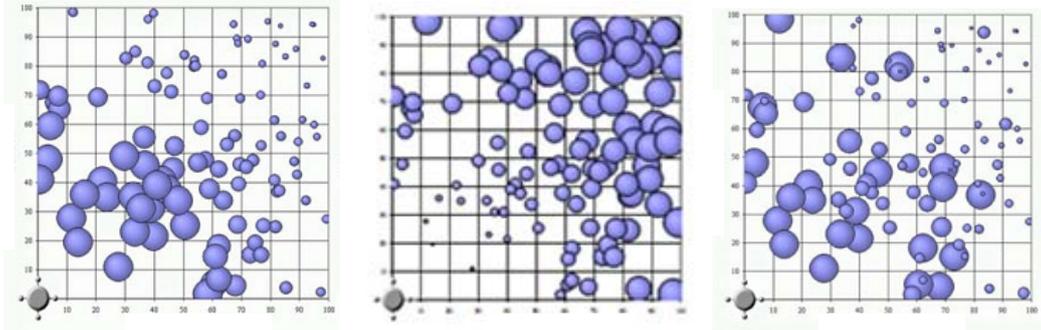


Figura 2.30. Tiempo de vida de los nodos para Comunicación Directa, Difusión Básica y *Clustering* Aleatorio respectivamente. El sumidero se localiza en la posición inferior izquierda (0,0) para la totalidad de las pruebas (fuente [RAICU04])

Las gráficas indican el estado de los nodos en el momento en el que un nodo se queda sin energía. El tamaño de las esferas es proporcional a la energía del nodo, luego esferas mayores indican mayor energía restante.

Cuando se comparan dichos resultados con los mostrados por *e3D* (figura 2.31), se observa como la disipación de energía se distribuye más por todos los nodos de la red, haciendo que el tiempo de vida de la misma aumente.

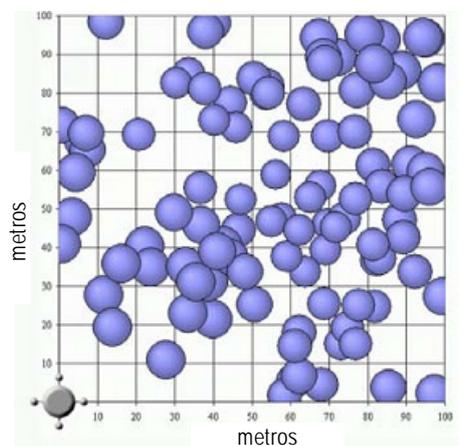


Figura 2.31. Tiempo de vida de los nodos aplicando Difusión *e3D*

*e3D* consigue estos resultados mediante la incorporación de las siguientes características:

- o Además de todo lo que el *algoritmo básico de difusión* realiza, cada nodo construye una lista de posibles vecinos y los ordena por preferencia. Cada vez que un nodo cambia de vecinos, el emisor requerirá un *ACK* para su primer mensaje, lo cual le asegurará de que el nodo receptor está vivo. Si se incurre en *timeout*, el emisor elegirá

- otro nodo vecino para transmitir y el proceso entero se repite de nuevo. Una vez que la comunicación se ha iniciado, no habrán más *ACKs* para ningún otro mensaje.
- Se producirán mensajes de excepción, los cuales provocan que el emisor no mande nada nunca más a dicho vecino, cuando el emisor detecte que el receptor tiene menos energía que él, concretamente menos del 50%. Este mecanismo permite minimizar la ingente cantidad de mensajes de control necesarios para intercambiar información entre dos nodos.
  - El coste de sincronización de *e3D* es de dos mensajes por cada par de nodos vecinos. El resto de decisiones se basan en información local, la cual se examina para determinar el mejor nodo siguiente al que se transmitirá. Cuando todos los nodos vecinos agoten su energía, los restantes enviarán directamente al sumidero. Este coste de sincronización según los autores es mínimo y dependiente de la aplicación, pero el tener un algoritmo basado en mensajes de sincronización y control con el fin de seleccionar el siguiente salto es pagar un precio muy caro para obtener mejoras que otras estrategias obtienen sin la **sobrecarga** de la sincronía.

Los autores no dejan constancia de los detalles del algoritmo, únicamente comparan sus resultados con lo que ellos consideran buenas estrategias, pero no con ningún protocolo conocido hasta la fecha ni con un enfoque específico. Debido a esto y a que el protocolo posee la sobrecarga citada anteriormente, consideramos que *e3D* no supone ventaja alguna sobre los algoritmos analizados hasta este apartado.

En su trabajo, los autores también comparan *e3D* con lo que ellos consideran “*clustering ideal*”, arrojando los resultados que podemos observar en la figura 2.32.

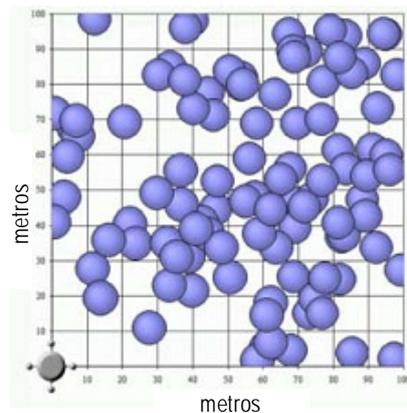


Figura 2.32. Tiempo de vida de los nodos para *clustering ideal*

Se deja constancia de que el enfoque *clusterizado* es una buena opción para un buen rendimiento energético, aún cuando ellos no lo emplean. Teniendo esto en mente, los autores proponen futuros trabajos basados en *clustering* jerárquico con agregación, puesto que *e3D* carece de ella, pudiendo generar mejores estrategias que ofrezcan más garantías y alta viabilidad.

### 2.3.12 Energy efficient clustering routing (EECR)

EECR [LILI06] desarrolla un enfoque jerárquico similar a los protocolos vistos hasta el momento. Comparándolo con *LEACH*, se ven mejoras significativas en el ahorro energético. El algoritmo propone que la formación de los *clústeres* y la elección de los *CHs* se realice conjuntamente por parte del sumidero y los nodos miembros. Se agrupan los sensores en *clústeres* y se les capacita para operar en dos modos de funcionamiento:

1. **Modo de sensorización**, donde los nodos sensorizan el medio y envían sus datos al *CH*.
2. **Modo de *CH***, donde dicho nodo recoge la información enviada por los miembros de su *clúster* y la fusiona, realizando labores de agregación y compresión, para posteriormente transmitir dichos datos a la estación base.

Como ya se puntualizó en algoritmos anteriores, el disponer de dos modos de trabajo en los nodos permite que el algoritmo sea más complejo y, disponiendo de los mecanismos apropiados, se consiga establecer una relación equilibrada entre modos y ahorro energético.

A su vez, el algoritmo opera en dos fases principales (figura 2.33):

1. **Fase de configuración**, donde se forman los *clústeres* y se eligen los *CH's*. El sumidero parte la totalidad de los nodos por la mitad, formando los dos primeros *sub-clústeres* y prosigue sucesivamente hasta conseguir el número deseado de *clústeres* (previamente calculado). Posteriormente, la estación base elige de nuevo a un *CH* de cada *clúster* que esté centrado en el mismo. Éste advierte a los demás de su nuevo rol y el resto de nodos eligen al *CH* al que seguirán, basándose en parámetros de potencia de la señal entrante. Cuando el *CH* recibe mensajes de unión a su *clúster*, éste asigna al nodo solicitante un slot de tiempo para transmitir datos.

Una vez que ha finalizado la primera ronda, el sumidero ya no es responsable de la elección de los *CH*. Esta tarea pasa ahora a los nodos. Éstos se convertirán en *CHs* o no, dependiendo de un peso asociado al nodo, cuyo cálculo se hace de manera local. Una vez un nodo se proclama como *CH*, éste seguirá el modo de funcionamiento *CH*.

Partiendo de la base de que los autores no indican detalladamente cómo el sumidero divide el conjunto de sensores, sabemos que dicha forma de trabajar asume que el sumidero está en cobertura con la totalidad de los nodos, lo cual no es siempre posible ni eficiente. Éste se involucra de manera activa en el proceso de formación de los *clústeres*, lo que da un carácter centralizado al protocolo que supone un sobrecoste en las comunicaciones. Aproximaciones distribuidas mejoran el rendimiento a largo plazo, por lo que se puede considerar este aspecto un *talón de Aquiles* del algoritmo.

2. **Fase de comunicación de información**, que consta de tres partes: *Adquisición de datos*, *fusión de la información* y *transmisión*. Inicialmente, cada nodo transmite su información al *CH* de su *clúster* durante su slot de tiempo. Para mejorar el ahorro de

energía, cada nodo dejará de transmitir durante los slots de tiempo que no le pertenecen.

Una vez que la información de todos los nodos ha sido recibida por el *CH*, éste realiza tareas de fusión y agregación para minimizar la cantidad de datos enviados a la estación base. Una vez que tanto la fase de recogida como la de fusión han finalizado, el *CH* transmite al sumidero los datos comprimidos.

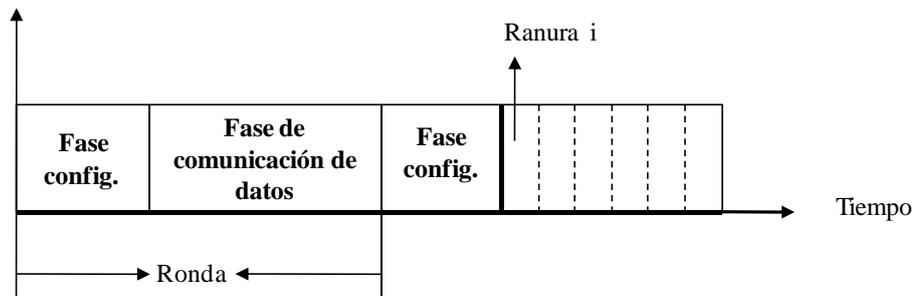


Figura 2.33. Composición de una ronda

Estas dos fases son similares a las desarrolladas por *LEACH*, lo que otorga al algoritmo de las mismas ventajas de agrupamiento (*clustering*) y rotación de roles.

Las simulaciones se han realizado para un número de nodos de  $N = 100$  distribuidos en un área de  $A = 100$  y  $d_1 = 90$  (distancia entre el *CH* y el sumidero), obteniendo un número de *clústeres* de 6. Los resultados de estas simulaciones muestran que este enfoque supera a *LEACH* tanto en número de nodos vivos al finalizar la simulación como en productividad de mensajes llegados al sumidero. *EECR* exhibe una mejora del 20% de reducción sobre la energía disipada por la red, como puede apreciarse en la figura 2.34.

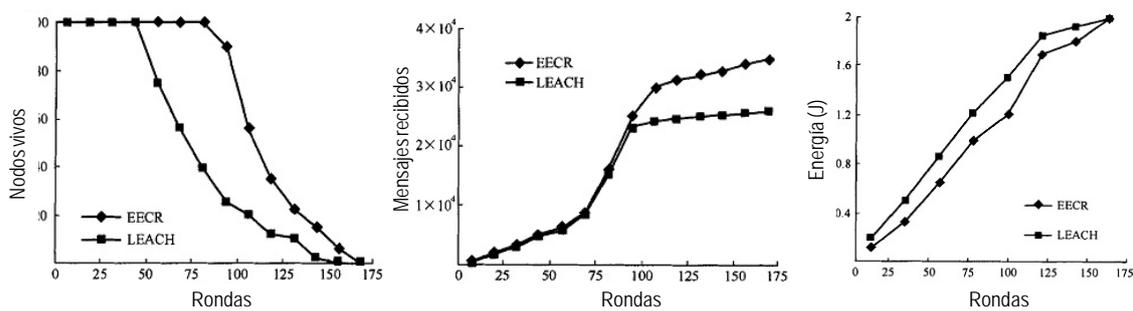


Figura 2.34. Número de nodos vivos, mensajes recibidos por el sumidero y disipación energética media

Vemos que ambos protocolos disipan similares cantidades de energía (*derecha*) pero *EECR* lo hace de una forma más equitativa, lo cual se observa en la primera gráfica (figura 34a), ya que *LEACH* conserva la totalidad de los nodos un número de rondas bastante inferior que *EECR*. La causa mayoritaria de esta ventaja se basa en la forma en la que se forman los *clústeres* y se

eligen los *CH's* en *EECR*. Éstos se construyen de forma centralizada desde el sumidero y, una vez ha finalizado la ronda actual, los *CH's* se auto-eligen mediante una variable ponderada, a diferencia de *LEACH* que realiza este cambio aleatoriamente. *EECR* optimiza más este cambio de *CHs*, haciendo que se distribuyan los roles de *CH* de una manera más eficaz y alargando la vida de la red consecuentemente.

Los autores hacen hincapié en la necesidad de futuras extensiones a sus ideas, centrándose en el rendimiento energético y en la manera en la que se rotan los roles de los sensores y cómo se forman los *clústeres*. La distribución de la energía siempre es mejor cuando se tienen mecanismos de formación de *clústeres* y jerarquización buenos, por lo que adquirir ideas sobre estos campos y fusionarlas con otras estrategias anteriores que hayan tenido buenas perspectivas, puede dar lugar a algoritmos y protocolos muy eficientes energéticamente hablando, que es la idea principal que persigue *EETA* y la presente tesis.

Consideramos que *EECR* es un buen enfoque en lo que a modos de funcionamiento y fases se refiere, al igual que destacamos similares conceptos en *SOP* o *LEACH*, pero consideramos necesaria la incorporación de mecanismos distribuidos para realizar de manera eficiente la elección de los *CHs* y la formación de los *clústeres*.

### 2.3.13 *IMRAFRA*

*IMRAFRA* [YIN08] es un algoritmo de enrutamiento que combina encaminamiento jerárquico con geográfico. Parecido a *LEACH* y con ideas similares, los sensores se organizan entre ellos en *clústeres* locales y un nodo actúa de *CH*. Las funciones de ese *CH* incluyen sus funciones tradicionales como adquisición de datos y *forwarding*, así como compresión de la información para minimizar el consumo.

Basado en la arquitectura de red jerárquica, el proceso de retransmisión de paquetes entre los nodos fuente en la región destino y el sumidero consiste en dos fases, *inter-clúster routing* e *intra-clúster routing*: se adopta un algoritmo voraz para el proceso de encaminamiento *inter-clúster* y un algoritmo de enrutamiento *multi-hop* basado en ángulo de retransmisión para el encaminamiento *intra-clúster*.

Los autores señalan las bondades del *intra-clúster routing* en los siguientes aspectos:

1. Puede aliviar los problemas de cuello de botella en el *CH*, ya que realizando agregación y compresión en los nodos intermedios se minimiza el futuro trabajo que haga el *CH*.
2. El fenómeno de largas distancias, en el cual las transmisiones inalámbricas de larga longitud son vulnerables a interferencias causadas por ruidos externos, se evita en mayor o menor medida debido a que las largas distancias se convierten en sucesivos saltos hasta el *CH*.

Sin embargo, estas técnicas presentan claras desventajas:

- Realizando compresión y agregación en los nodos intermedios del encaminamiento hacia el *CH*, se minimiza el trabajo que deba hacer el *CH* pero a costa de añadir **latencia** en el proceso de adquisición de los datos, y complicando las funciones a realizar por nodos distintos del *CH*, y por tanto aumentando su consumo energético, comprometiendo la durabilidad de la red, ya que se está realizando un trabajo en los sensores que se debe hacer en los *CHs*.
- La selección de los *CHs* se hace en función de su aptitud para cumplir con las misiones de agregación y compresión, mientras que la de los nodos intermedios se realiza por criterios estrictamente geográficos, pudiendo darse el caso de nodos con poca energía que, por su localización, deban realizar estas tareas. En estas circunstancias, el tiempo de vida del nodo no será óptimo.

Además, cuando *el número de nodos dentro del clúster es alto*, las cadenas formadas son largas y de nuevo surge el problema de **latencia** en las transmisiones hacia el *CH*, pudiendo incurrir en una alta **congestión** y perdiendo calidad de servicio. Además, el mantenimiento de dichas cadenas supone un gasto energético mayor si no se proponen mecanismos al *respecto*.

Una mejor opción sería no realizar encaminamiento alguno dentro del *clúster* sino que los sensores y el *CH* estén a un solo salto en la transmisión. En esta línea, los autores proponen dos posibles vías para mejorar el rendimiento, tanto en el encaminamiento *inter-clúster* como en el *intra-clúster*.

- *Encaminamiento inter-cluster*: El *CH* designado para transmitir hacia el sumidero lo hace comunicándose directamente o a través de otros *CHs*. En su aproximación, este enrutamiento se hace de una manera voraz. Cuando se pretende seleccionar un nuevo salto para la transmisión, el *CH* escoge la ruta hacia el vecino que posee el camino de menor coste hasta el sumidero, esto es, se tiene en cuenta el coste de sus vecinos al completo para seleccionar el siguiente *hop*. Emplear siempre el camino de menor coste para subir los datos hacia el *sink* conlleva un **mayor gasto de los nodos involucrados**, como se vio en *Energy Aware Routing*, por lo que este enfoque de *inter-clúster* no llega a ser adecuado.

En la *figura 2.35* se aprecia en azul el encaminamiento correspondiente a la fase *inter-clúster*, donde la información de los *clústeres* es encaminada hacia el sumidero.

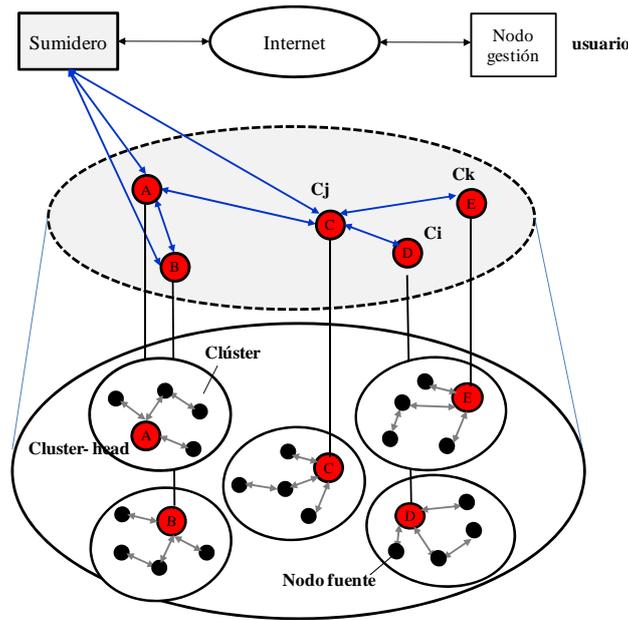


Figura 2.35. Encaminamiento inter-clúster en IMRAFRA

- *Encaminamiento intra-cluster* : Cuando los nodos transmiten la información al CH, se propone un nuevo método para el envío a través de la cadena. Basándose en la teoría de la desigualdad triangular del consumo de energía en las WSNs, el algoritmo trabaja pasando únicamente por aquellos nodos que minimizan la energía en un ángulo limitado de acción, realizando tareas de fusión en los propios nodos intermedios y no sólo en el CH. Como se puede ver en la *figura 2.36*, el nodo *D* calcula el ángulo entre la arista *DT* y *TC*, denominada *DTC*. Si *DTC* es mayor que un ángulo predeterminado de *forwarding*, *T* entonces pertenecerá al área limitada por el ángulo y el nodo *D* encaminará los datos hacia *T*. Si esto no se cumpliera, *D* debería buscar otro nodo intermedio en su lista de vecinos que cumpla las restricciones indicadas. Este cálculo implica información geográfica o que cada vecino difunda una tabla con sus potencias observadas, y estimar así la distancia, lo cual sería inexacto.

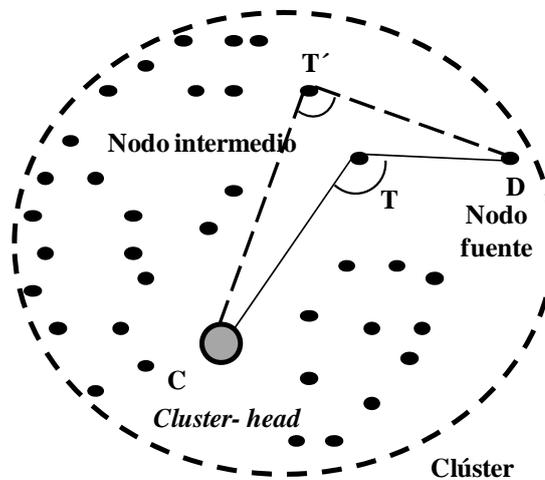


Figura 2.36. Limitación del ámbito de retransmisión empleando el ángulo de restricción



recibir un paquete del nodo  $B$ . Según el ajuste del radio de transmisión anterior, manteniendo la asunción en el radio de comunicación, esto sí sería cierto, pero debido a que es una mejor alternativa que esto no sea así y disponer de distintos radios de comunicación en los nodos, la comunicación simétrica no siempre es asumible.

4. La localización de cada nodo no es conocida ya que no se dispone de  $HW$  específico como  $GPS$  ni de mecanismos de triangulación. Esta asunción es totalmente válida ya que la incorporación de este  $HW$  conllevaría un mayor consumo y sin esta localización el protocolo puede desempeñar totalmente sus funciones.

$UDACH$  se divide en tres etapas, donde se construyen los *clústeres* y el árbol de  $CHs$  y posteriormente se envía la información. Más detalladamente:

1. *Construcción de los clústeres*: en esta etapa, cada nodo tiene una cierta probabilidad de designarse como  $CH$ , basándose en ciertos criterios:
  - a.  $P_{ini}$  es la probabilidad inicial del nodo para ser  $CH$ .
  - b.  $MARK$ , peso asociado a la energía remanente en el nodo.
  - c.  $W_c$  es el peso del nodo, el cual es inversamente proporcional a la distancia entre el  $CH$  y el nodo normal.
  - d.  $S_{can}$  es el conjunto de nodos que inicialmente se proclaman como  $CHs$  en un radio  $r$ .
2. *Construcción del árbol de  $CHs$* : con el fin de ahorrar consumo,  $UDACH$  organiza los  $CHs$  como un árbol, de acuerdo a la distancia al sumidero. Esta distancia sigue el siguiente parámetro:
  - a.  $W_d$  es el peso que determina la construcción del árbol, donde  $W_d = 1 / Distancia_{BS}$ . Sucesivos *broadcast* de este parámetro conforman un orden en los nodos, donde el que posea mayor valor será el padre de los restantes. A igualdad de valores, el nodo que posea el mayor  $ID$  será el elegido.

De esta forma se elabora un orden en el árbol, donde cada nodo sabe donde está localizado él mismo y cuál es su hijo directo. La manera en la que se establece el criterio padre-hijo es única y por lo tanto en el árbol de  $CHs$ , cada nodo tendrá únicamente un hijo. Esto difiere en gran medida en el protocolo propuesto en la presente tesis donde en el árbol de  $CHs$ , éstos pueden poseer más de un hijo dentro del árbol, por lo que la agregación y fusión de los datos tiene mayor sentido.

3. *Envío de datos*: los nodos recolectan la información necesaria y posteriormente se la envían a su  $CH$  periódicamente. El  $CH$  fusiona los datos recibidos y envía el resultado al sumidero

Tal y como se describe el proceso de comunicación de  $UDACH$ , se puede concluir que no es nada elaborado ni detallado para extraer el máximo partido a la energía de los

sensores. En la presente tesis se proponen mecanismos como *TDMA* con posibilidad de *polling intra-clúster* o *ACKs* para conseguir realizar el envío de los datos de la mejor manera posible. *UDACH* no da detalles el respecto sobre cómo dichos datos llegan paso a paso hasta el sumidero ni qué mejoras se incorporan respecto a anteriores aproximaciones en la fase de comunicación, por lo que pensamos que este aspecto necesita la debida atención.

Siguiendo esta línea de argumentación comparativa, se exponen los siguientes puntos:

- *UDACH* no presenta ningún mecanismo de tolerancia a fallos en los *CHs* ni en los sensores, aspecto vital cuando se habla de árbol *inter-clúster* ya que la caída de un *CH* es fatal tal y como se ha planteado en *UDACH* la construcción del árbol. En el protocolo propuesto se dispone de mecanismos de control de fallos en los nodos así como restitución de la funcionalidad del árbol ante cualquier error en los *CHs* por lo que su robustez es más amplia.
- En lo referente a coste temporal de las comunicaciones, tampoco se dan cotas máximas ni mínimas para las comunicaciones con el *sink* por lo que las posibles aplicaciones del protocolo para tareas de tiempo real quedan anuladas. El protocolo propuesto por el contrario trabaja con planificaciones y con ranuras *TDMA* por lo que en la medida de lo posible sí que ofrece dicho servicio, muy importante para controlar redes de sensores que trabajan con aspectos críticos de control.
- El protocolo propuesto ofrece la posibilidad de una comunicación bidireccional entre sumidero-nodos y comunicación *P2P*, por lo que aprovechando el flujo de comunicación entre los *CHs* y su *clúster*, se pueden dar órdenes provenientes del *sink* para cambiar aspectos o parámetros en los nodos. La comunicación *P2P* brinda la posibilidad de enviar directamente información entre dos nodos cualesquiera, ampliando el espectro de comunicaciones. *UDACH* carece de cualquiera de estos mecanismos por lo que a la hora de ofrecer servicios de comunicación entre nodos se queda bastante corto.

Para evaluar el rendimiento del protocolo, éste se compara con *LEACH* y *LEACH-C* en dos entornos: (1) *BS* localizada en (50, 100) y área de 500x500m con  $N = 100$  nodos distribuidos aleatoriamente y (2) *BS* localizada en (50, 175) y área de 1000x1000m con  $N = 100$  nodos distribuidos aleatoriamente. Cada nodo posee la misma cantidad de energía inicial, 2J. Una vez que un nodo muere, se extrae de la simulación, la cual tiene una duración de 200 segundos. Para simplificar los experimentos, se ha escogido un radio de transmisión fijo de 150m, que es aproximadamente la distancia media entre dos nodos en las simulaciones de *LEACH* y *LEACH-C*.

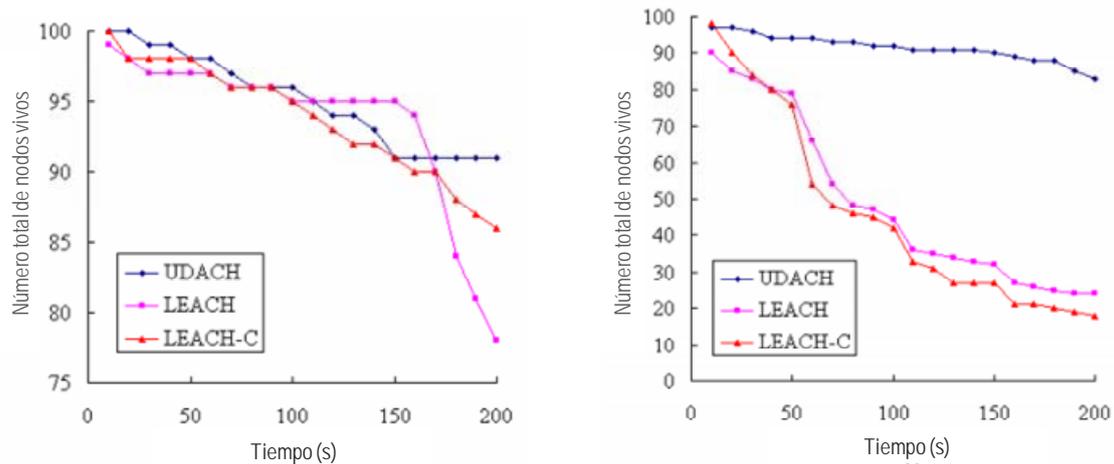


Figura 2.38. Número de nodos vivos para el área de (a) 500x500m y (b) 1000x1000m

Se ve claramente en la figura 2.38 como *UDACH* distribuye de mejor manera la energía de la red al conservar un mayor número de nodos vivos. El resultado es aún más claro conforme la red posee más extensión, ya que para el caso de  $1000 \times 1000m$ , aproximadamente el 87% de los nodos se mantienen vivos, frente al 25% en el mejor caso de *LEACH* o *LEACH-C*. El cambio es radical conforme la red aumenta de tamaño y esto es consecuencia de que en *LEACH* y en *LEACH-C* la disipación de energía debido a transmisión es proporcional a la distancia entre nodos, luego una red más extensa, conservando el mismo número de nodos y por lo tanto incorporando mayores distancias entre ellos, significa un gasto desproporcionado, palpable en los resultados. Esta disipación de energía se puede ver claramente en la figura 2.39, donde *UDACH* muestra un 50% menos de disipación energética:

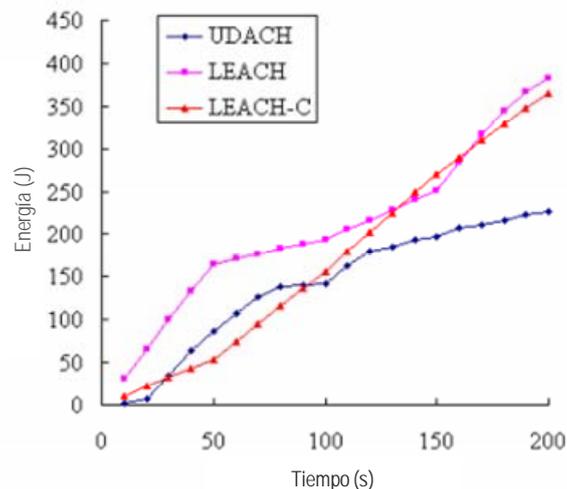


Figura 2.39. Energía total disipada para el área de 1000x1000m

Queda patente que la formación de un árbol de *CHs* para el enrutamiento de los paquetes en dirección al sumidero conlleva un menor gasto y una mejor disipación de la energía a lo largo de toda la red. El modo en que los *clústeres* se construyen, apoyándose en el conocimiento de su energía restante, da ventaja a *UDACH* frente a otras aproximaciones que no tienen esta

característica en cuenta. Pero como ya se vio anteriormente, la construcción del árbol en *UDACH* es demasiado simple, cuando organizándose de distinta forma se conseguirían mejores resultados en las comunicaciones *inter-clúster*.

La sobrecarga de mensajes requeridos para la formación del árbol, también consumen energía y sin mecanismos ni estrategias apropiadas para controlar esta **sobrecarga extra**, puede provocar mayor energía consumida, como se demuestra en la *figura 2.40*, donde únicamente para la fase de configuración, donde se forman los *clústeres* y el árbol, se aprecia como *UDACH* paga esta sobrecarga extra:

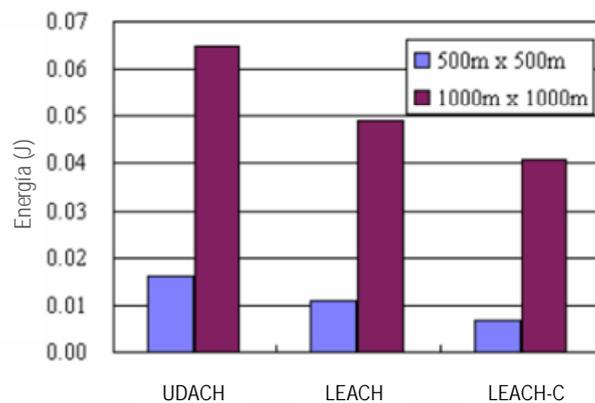


Figura 2.40. Consumo energético en la fase de configuración para diferentes tamaños de red

En comparación con *LEACH* y su versión centralizada, el gasto energético en la fase de *setup* es mayor, ya que principalmente ninguna de las dos versiones de *LEACH* realiza construcción de árbol alguna por lo que obviamente el gasto será mayor. Esto no quita que no haya que controlar este coste en el consumo pero sí tener en cuenta que las ventajas que derivan de este árbol de *CHs* son mayores que el coste que se necesita para su elaboración, por lo que mayoritariamente compensa.

Las simulaciones muestran como *UDACH* consigue un menor consumo y un mayor tiempo de vida de la red, como consecuencia de la elección de los *CHs* basándose en criterios de energía remanente y la construcción de un árbol de *CHs*. Este protocolo muestra grandes incrementos de rendimiento incorporando estas nuevas estrategias que pueden ir en la dirección correcta siempre y cuando la sobrecarga de mecanismos como estos se controle. Como siempre se ha ido remarcando, consideramos la construcción de estructuras de datos simples y eficientes en el enrutamiento *inter-clúster* como una medida apropiada, apostando por ello y por mecanismos de elección basados en energía remanente, tal y como muestra *UDACH*. Queda en el aire el cómo instaurar estos mecanismos para equilibrar las mejoras con las posibles desventajas que provoquen.

### 2.3.15 Tree-Clustered Data Gathering Protocol (TCDGP)

Como ya se apreció con anteriores protocolos, el enfoque clusterizado con árbol jerárquico obtiene buenos resultados en lo que a eficiencia energética se refiere. Con esta misma idea surge *TCDGP* [HUANG07], pretendiendo obtener mejores resultados aplicando otro tipo de estructura de datos en árbol.

*TCDGP* posee las siguientes asunciones, al igual que protocolos anteriores con sus ventajas y desventajas:

1. Cada nodo o el propio sumidero tiene la capacidad de transmitir mensajes a otro nodo o sumidero **directamente**. Esta capacidad no conlleva distinción alguna en las bondades del protocolo ya que normalmente la distancia entre los nodos y el *sink* suele ser grande y transmisiones grandes conllevan un mayor gasto. Aunque se tenga la capacidad de transmitir a grandes distancias siempre es mejor adecuar la potencia de las transmisiones al rango de actuación requerido, ya sea dentro del clúster o de manera más refinada.
2. Cada nodo tiene control sobre la potencia de transmisión empleada para sus comunicaciones. Como ya se dijo en el punto anterior, esta asunción sí es beneficiosa y requerida para un consumo eficiente.
3. Cada nodo en la red posee la **misma energía** inicial. Suele ser normal que a la hora de disponer los sensores en la zona, todos posean la misma energía inicial. Lo que ya no es asumible es que todos disminuyan su batería de manera similar ni que, en un futuro a medio o largo plazo, nuevos nodos conjuntamente con los anteriores muestren los mismos niveles de energía.
4. Cada nodo posee información de su **localización**. Como ya se vio en protocolos anteriores, no siempre es viable el proveer de *GPS* o algún otro mecanismo de localización a los nodos ya que cuanto más circuitería, más consumo.
5. La red de sensores no debe tener mantenimiento humano, es decir, debe ser **autónoma**. Este es el objetivo de la red de sensores, mantener un control sobre un proceso externo sin intromisión humana y de manera remota.
6. Los sensores inalámbricos se despliegan de manera **aleatoria y densa** en el campo exterior.

De manera global podemos considerar el diagrama de flujo del protocolo (figura 2.41), como medida para distinguir en líneas generales el proceso del mismo.

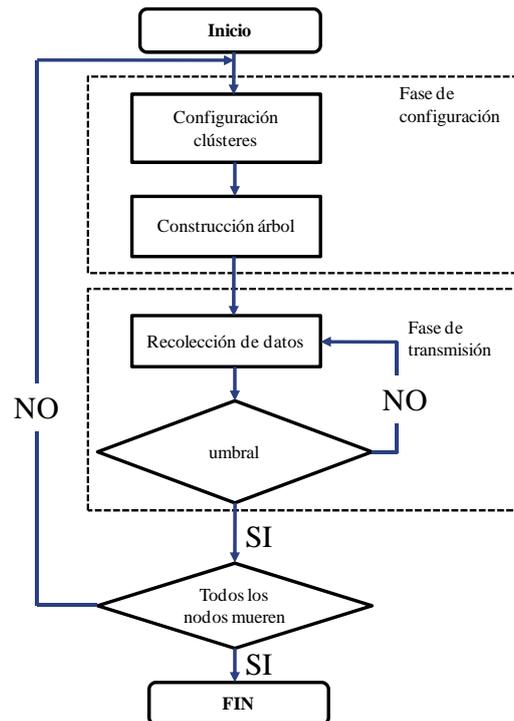


Figura 2.41. Diagrama de flujo de TCDGP

A continuación, se exponen de forma un poco más extensa las operaciones del protocolo, correspondientes al diagrama anterior:

- *Establecimiento de clústeres:* el sumidero construye los clústeres mediante el cómputo de las distancias entre los nodos, extraídas de tablas en los mismos que se actualizan periódicamente. A su vez, también obtiene la energía remanente de cada nodo, detallada en dicha tabla. De acuerdo a toda esta información, el sumidero selecciona el *CH* en base a distancias. Este proceso de elección se repite hasta que se alcanza un número de *CHs* determinado anteriormente y suficiente para cubrir las necesidades de la *WSN*.

Una vez se han seleccionado los *CHs*, el sumidero puede calcular la distancia entre dichos *CHs* y los nodos de acuerdo a coordenadas. Los nodos normales informarán a los *CHs* de su intención de unión a aquel que esté más cerca de ellos. Es entonces cuando los *CHs* identifican con una etiqueta aquellos nodos que desean pertenecer a su clúster. Tan pronto como esta fase termine y se conformen los clústeres, todos los *CHs* comenzarán a establecer el árbol entre ellos, correspondiente a la siguiente fase.

- *Construcción del árbol:* el sumidero recogerá la información de etiquetas que han ido elaborando los *CHs* y construye un *Árbol de Expansión Mínimo (Minimum Spanning Tree – MST)*, usado normalmente para resolver problemas de selección de rutas de mínimo peso en grafos no-dirigidos. Después de eliminar algunas conexiones en dicho árbol, el sub-grafo sigue siendo conexo, manteniendo la estructura de árbol anterior y la propiedad en la suma de sus pesos mínimos. Debido a que el cálculo de este *MST* de

manera voraz presenta un coste exponencial desde el punto de vista computacional, se emplea el conocido algoritmo de *Prim* para tal fin.

- *Agregación de datos*: una vez los mecanismos de enrutamiento se han establecido, todos los nodos transmiten los datos sensorizados a nodos en niveles superiores. Estos nodos fusionarán la información y la reenviarán a niveles superiores de nuevo. El proceso se repite hasta que se llega al nodo raíz. Cuando todos los nodos finalizan sus transmisiones, decimos que se termina una *ronda*, y el proceso se inicia de nuevo. En la figura 2.42 se observa este proceso.

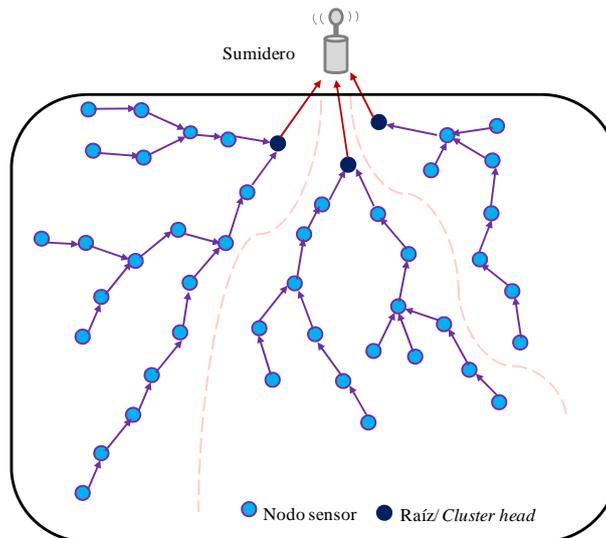


Figura 2.42. Agregación en TCDGP

Para el proceso de simulación se emplea el módulo *sLEACH* de *OMNet++*, dejando constancia de que el entorno de simulación es similar al de *LEACH* o *PEGASIS*. Los parámetros de las simulaciones se pueden observar en la tabla 2.1.

Número de nodos	100
Ubicación de los nodos	Aleatoria
Ubicación del sumidero	(50,300)
Área sensorizada	100m x 100m
Tamaño de paquete	2000 bits
Energía inicial	0,5 J
Coste fusion	5 nJ/bit/mensaje
Consumo radio	50 nJ/bit
Consumo etapa ampl.	100 pJ/bit/m <sup>2</sup>
Número de <i>cluster heads</i>	$C = 1 \sim 3$
Umbral	$25 < t, a = 2,3$

Tabla 2.1. Parámetros de los experimentos

A continuación, se compara *TCDGP* con Difusión Directa, *LEACH* y *PEGASIS*, y a su vez se prueban diferentes valores del parámetro  $c$ , correspondiente al número de *clústeres* (ver figura 2.43).

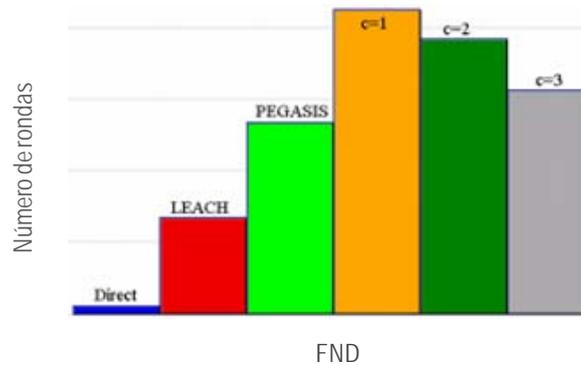


Figura 2.43. Comparación en número de rondas con *Direct Diffusion*, *LEACH* y *PEGASIS* mediante FND (*First Node Death*)

Se observa que la mejora en comparación con anteriores protocolos es sustancial y, dentro de *TCDGP*, el rendimiento con  $c = 1$  *clúster* es notable. De nuevo, tener constancia de la energía remanente a la hora de elegir los *CHs* es fundamental para repartir el consumo por la red y prolongar la vida de la misma.

Los autores no grafican comparaciones en cuanto a energía disipada para ver de manera más clara las ventajas del protocolo, quizás ocultando los malos resultados que la **sobrecarga de la construcción del MST y mensajes de información** entre el sumidero y los nodos les han causado. Las ideas, al igual que en *UDACH*, son buenas pero hay que ir con cuidado en cómo se incorporan y mediante qué mecanismos, como ya dijimos anteriormente ya que si no pueden acarrear desventajas directas en el consumo, que es lo primordial.

### 2.3.16 Energy-Efficient Distributed Unequal Clustering (EEDUC)

El algoritmo *EEDUC* [LEE09] se propone con el fin de proporcionar una nueva manera de crear *clústeres* distribuidos. En *EEDUC*, cada nodo configura un tiempo de espera. Dicho tiempo de espera se establece en función de la energía residual y del número de nodos vecinos, ayudando de esta forma al protocolo a distribuir los *CHs*.

Los autores presentan *EEDUC* como solución al problema del *hotspot* (punto caliente) en *WSNs*. Dicho problema, bien conocido en el mundo de las *WSNs*, radica en que cuando el radio de transmisión es fijo en los nodos, la cantidad de tráfico retransmitido por los sensores se incrementa notablemente a medida que se reduce la distancia al sumidero. Por ello, los sensores más cercanos al sumidero tienden a agotar antes su energía. Es de vital importancia controlar este aspecto en la medida de lo posible, puesto que este efecto hace que la conectividad de la red se pierda, aunque la mayoría de los nodos sigan en funcionamiento,

cuando los nodos cercanos al sumidero pierdan su energía. Como consecuencia de ello, el tiempo de vida de la red depende directamente del tiempo de vida de dichos nodos.

En *EEDUC*, a diferencia de en *LEACH*, los *CHs* se escogen por **competición local y sin iteración**. Anteriormente, en *EEUC* [CFLI05] (*An Energy-Efficient Unequal Clustering*), se sugirió, como solución al problema, que los *clústeres* más cercanos al *sink* fueran más pequeños. De esta forma, se esperaba que consumieran menos energía en el procesamiento de los datos *intra-clúster* y pudieran preservar más energía para la tarea del encaminamiento *inter-clúster*. Sin embargo, existe la posibilidad de que un *CH* escoja como nodo siguiente del enrutamiento a un sensor de un *clúster* adyacente, sin considerar el equilibrado del tráfico. En *EEDUC*, se centran en el equilibrio del tráfico y el *clustering* distribuido. Solucionan el problema del *hotspot* considerando el número de vecinos de los nodos y su energía residual.

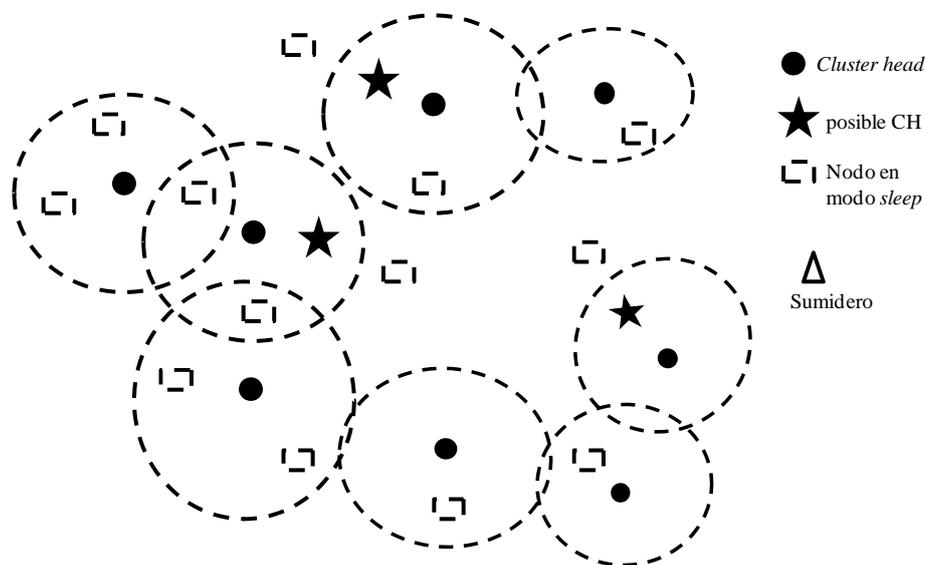


Figura 2.44. Visión general del mecanismo de comunicación en *EEUC*

Cuando la red de sensores se inicializa, el sumidero sincroniza con todos los nodos con un mensaje de advertencia, realizando un *broadcast* de este mensaje a todos los nodos. Una vez los nodos reciben dicho mensaje, calculan su distancia al sumidero sirviéndose de la calidad de señal con la que el mensaje les llegó. Posteriormente, cada nodo realiza un *broadcast* de advertencia con el fin de contar cuántos vecinos tiene en el rango de *1-hop*. Luego, el sumidero asigna un valor aleatorio a cada nodo y decide la distribución de unidades de tiempo para *clustering*. Cada sensor decidirá un tiempo de espera que cuadre con dichas unidades de tiempo.

Como ya se dijo anteriormente, con el fin de que los *CHs* más cercanos al sumidero no agoten su energía prematuramente, *EEUC* propone un mecanismo de *clustering* desigual, haciendo que los *clústeres* cercanos al sumidero sean más pequeños. Pero *EEUC* no consigue cubrir la red entera. Estas áreas no cubiertas son las que causan los desequilibrios de la red. En *EEDUC*, se considera el número de vecinos del nodo *i* para cubrir la red en su totalidad.

Después de la fase de inicialización, cada sensor decreta su tiempo de espera en sincronía con un temporizador interno. Cuando dicho tiempo de espera llega a 0, el nodo se erige como CH e informa de su status a sus vecinos mediante el *broadcast* de un mensaje de HELLO. El **radio de competición** del mensaje de HELLO se dictamina en función de la *distancia al sumidero*, el *número de vecinos* y la *energía residual del CH* mediante la expresión (2.1):

$$R_{comp} = \left[ \begin{array}{l} 1 - w_1(1 - TS_i / TS_{Máx}) - \\ w_2(1 - E_i / E_{Máx}) - w_3 N_i / N_{Máx} \end{array} \right] R_{Máx} \quad (2.1)$$

donde  $R_{comp}$  es el radio de competición obtenido,  $TS_i$  denota la distancia entre el nodo  $i$  y el sumidero,  $TS_{MAX}$  es la distancia máxima entre el sensor y el sumidero,  $E_i$  es la energía residual del nodo  $i$ ,  $E_{MAX}$  es la energía máxima del sensor,  $R_{MAX}$  es el tamaño máximo del *clúster* y  $w_1$ ,  $w_2$  y  $w_3$  son coeficientes constantes entre 0 y 1.

Cuando se realiza el *clustering* de nuevo en la siguiente ronda, cada nodo cambia y decide su propio tiempo de espera. Empleando este tiempo de espera, cada CH se distribuirá correctamente por la zona de sensorización.

En la figura 2.45 se describe de manera gráfica el proceso completo.

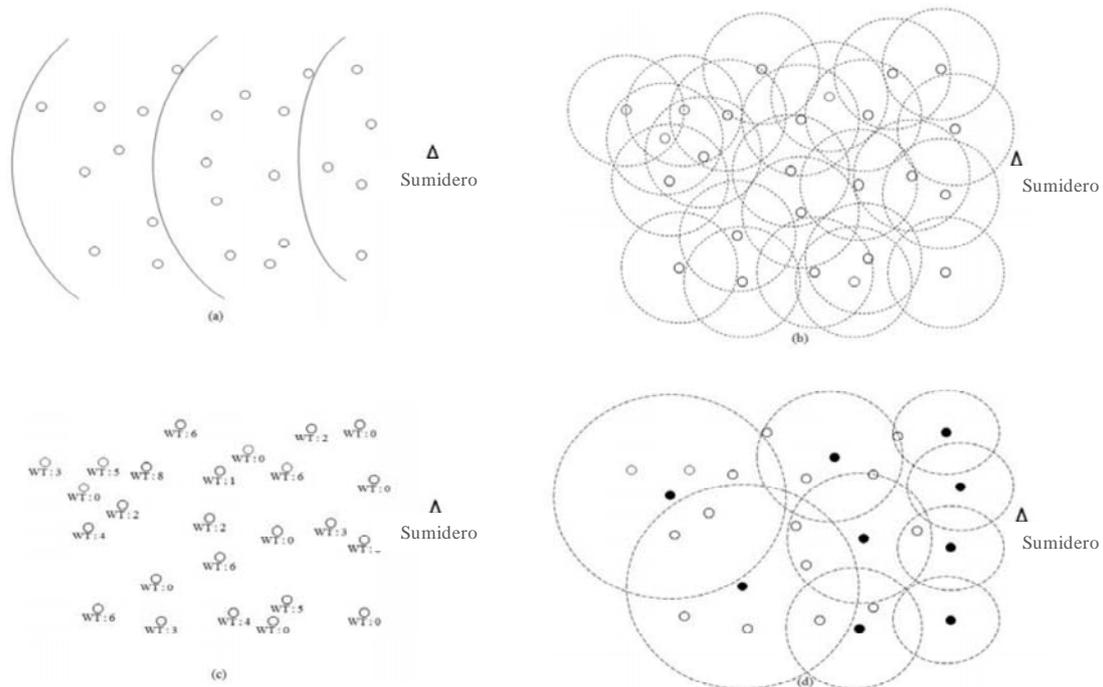


Figura 2.45. Fase de *clusterización* en EEDUC: (a) *broadcast* del mensaje de advertencia por parte del sumidero, (b) *broadcast* del mensaje de advertencia por parte del nodo, (c) decremento del tiempo de espera de los nodos y (d) *clustering* completo

Antes de entregar la información al sumidero, los *CHs* agregan los datos de los miembros de su *clúster* y posteriormente la envían al sumidero mediante comunicación *multi-hop*. En estrategias anteriores como *LEACH*, la información viaja hasta el sumidero en *1-hop*, pero ya quedó demostrado que esto incrementa el consumo de los *CHs* de forma sustancial. Esta comunicación *multi-hop* entre *CHs* tiene como base un algoritmo *multi-hop*, el cual sencillamente hace que los *CHs* escojan el siguiente salto de transmisión aleatoriamente entre sus vecinos *CHs*. De una forma más eficaz se podría haber adoptado algún mecanismo, como un árbol de *CHs*, para hacer esta tarea más eficiente en vez de aleatoria.

Las simulaciones se han llevado a cabo con el objetivo de comparar *EEDUC* con *LEACH* y *EEUC*, en función del consumo de los *CHs* individualmente y de la energía global restante de la red. En primer lugar en la *figura 2.46* se puede observar como *EEDUC* mejora notablemente a *LEACH* en cuanto a consumo de los *CHs* en 15 rondas. La mejora queda casi anulada cuando se compara con *EEUC*, ya que donde realmente se esperan resultados mejores es en cuanto a energía global consumida por la red.

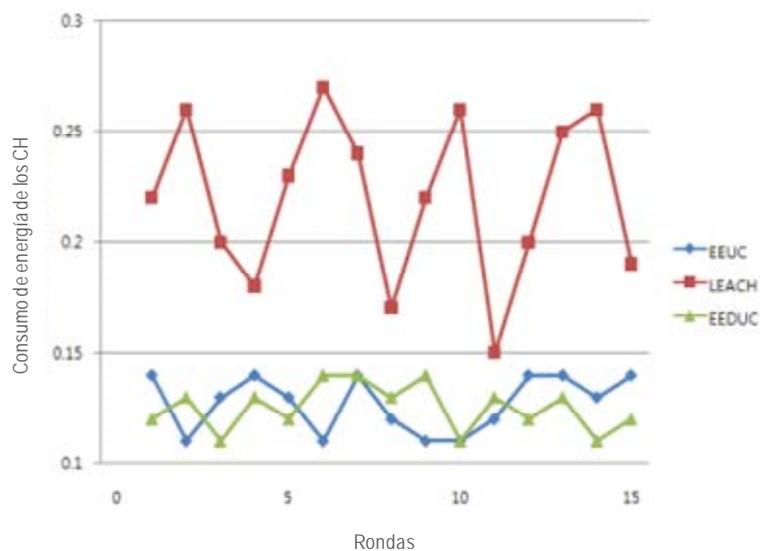


Figura 2.46. Energía consumida por los *CHs*

En la *figura 2.47* se muestra la energía restante total de la red, donde se ve claramente cómo *EEDUC* balancea el consumo de energía entre los *CHs* mejor. En *EEUC*, los *CHs* no pueden cubrir perfectamente la red por lo que la **sobrecarga** en las comunicaciones con el sumidero sale a la luz.

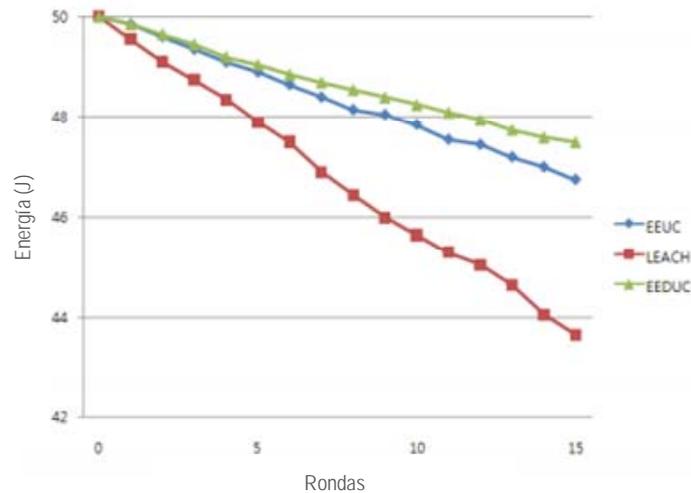


Figura 2.47. Energía total restante en la red

Aunque *EEDUC* no incorpore varias mejoras en cuanto a *clustering* se refiere, se centra en el problema de los *hotspots*, dejando patente esfuerzos por solucionar un gran problema de las *WSNs*. Su mejora respecto a *EEUC* es palpable pero no muy amplia, aunque en comparación con *LEACH* obtiene una mejora significativa.

Se viene remarcando hasta el momento que el conocer de manera global la energía de los nodos para saber qué opciones tomar o cuales no es de vital importancia. El poder saber estos valores energéticos no siempre es viable sin provocar **sobrecarga** en las comunicaciones y la consecuente **latencia** en las mismas. *EEDUC* se basa en este conocimiento para realizar una competición entre los posibles *CHs*, con el fin de optimizar en lo posible el consumo. Queda patente en las simulaciones que la mejora con respecto a estrategias anteriores como *LEACH* es sustancial pero también se echa en falta alguna comparación con estrategias de mayor calibre y no solo con *LEACH*. De ese modo se podría saber en qué forma y cuánto mejora *EEDUC* a estrategias de última generación. La línea abierta que deja *EEDUC* para el equilibrio en los *clústeres* se podría aprovechar para futuros trabajos junto con un buen enfoque de *encaminamiento inter-clúster* (árbol jerárquico), fundiendo estrategias que mantengan buenas expectativas energéticas.

### 2.3.17 Hierarchical Tree based Energy efficient and Congestion aware Routing Protocol (HTECRP)

Los autores del protocolo *HTECRP* [MOHAJER09] afirman que es eficiente desde el punto de vista energético, además de manejar la congestión y distribución en la red. En *HTECRP* se pretenden emplear las mejores rutas para el tráfico de alta prioridad y para prevenir la congestión, por lo que los autores sugieren la utilización del protocolo para labores de carga en tiempo real.

En los protocolos de enrutamiento jerárquico, el proceso de encaminamiento se divide en dos fases principales. En *HTECRP*, estas diferentes fases son tres:

1. *Establecimiento de clústeres*: En esta fase, los nodos de la red son particionados en diferentes *clústeres*. Dentro de la fase de *clustering*, se elige un *CH* para cada *clúster*. Al final de esta fase, la información de cada uno de los nodos debe ser entregada al *CH*.
2. *Creación de árbol*: En esta fase se usa la información enviada por los nodos en la fase de formación. En una estructura basada en árbol, se determina un camino desde cada nodo hasta el sumidero. El *CH* sabe la posición de todos los nodos que se encuentran en su *clúster* y es entonces cuando, en la primera parte de esta fase, el *CH* evalúa el coste del enlace entre todo par de nodos que se sitúen en su rango de comunicación mediante la siguiente fórmula (2.2):

$$\text{cost}_{ij} = \sum_{k=0}^3 CF_k = c_0 x(\text{dist}_{ij})^L + c_1 + c_2 + c_3 x f(e_{ij}) \quad (2.2)$$

donde:

- $CF_0$  (coste de la comunicación): este coste refleja el coste de la potencia de la transmisión *wireless*.
- $CF_1$  (stock de energía): este factor refleja el tiempo de vida de la batería principal.
- $CF_2$  (coste del estado de sensorización): constante añadida cuando el nodo se encuentra en estado de sensorización.
- $CF_3$  (ratio de error): los enlaces con alto ratio de error incrementarán el coste de la función, por ello serán evitados.

Los *CHs* usando la información de los nodos, el coste del enlace y el algoritmo de *Dijkstra* seleccionan la ruta de menor coste entre cada nodo del *clúster* y el sumidero. Empleando el algoritmo de *Dijkstra*, la ruta seleccionada entre cada nodo y el *sink* es óptima, por lo que el conjunto de todas las rutas tiene estructura de árbol, llamado árbol de enrutamiento. Si un nodo usa el camino de menor coste seleccionado para transmitir su tráfico, la red consumirá la menor cantidad de energía para dicho tráfico. Pero es importante hacer notar que, con respecto a los parámetros citados anteriormente correspondientes a la función del coste del enlace, **no siempre el camino de menor coste es el mejor camino**.

*HTECRP* considera dos tipos de tráfico: de *alta prioridad* y de *baja prioridad*. El tráfico, basándose en su prioridad, consigue más o menos servicios de red. *HTECRP*, después de construir el árbol de enrutamiento, lo mejora. Para cada nodo se determina la cantidad de tráfico de alta prioridad y la habilidad de encaminar los datos de otros nodos. Basándose en los parámetros anteriores, se determina la mayor cantidad de nodos hijo que puede tener un sensor para evitar la congestión. Un nodo hijo es aquel nodo que seleccionan los *clúster* formantes como siguiente salto de su enrutamiento hacia el *sink*. Después de determinar el máximo número de hijos posibles, el árbol de

enrutamiento cambia para equilibrarse en la medida de lo posible, no dejando que ningún nodo sobrepase ese máximo. Entonces, los *CHs* poseen la suficiente información sobre todas las rutas de los *clústeres* como para encontrar los hijos de cada nodo de manera sencilla.

Después de seleccionar la mejor ruta y determinar el número de nodos hijo para cada nodo, el *CH* crea una tabla de enrutamiento para cada nodo del *clúster*. Se añade una entrada especial en la tabla para la mejor ruta seleccionada. Para cada uno de los nodos vecinos que tengan la menor distancia al sumidero, se registra también una entrada en la tabla.

3. *Intercambio de datos*: al finalizar la fase de formación, todos los nodos poseen una tabla de enrutamiento. El objetivo principal de esta fase es determinar el siguiente salto del enrutamiento para cada paquete entrante. Dependiendo del tipo de tráfico, se considera uno u otro proceso de enrutamiento. La tabla de enrutamiento de los nodos debe ser actualizada constantemente para que el nodo no pueda desempeñar su labor eficientemente. Cuando la energía residual de un nodo baja de un umbral específico  $\alpha$ , informa a sus vecinos de su condición actual mediante un mensaje de *broadcast*. Los nodos que reciben el este mensaje, actualizan la entrada correspondiente en su tabla de enrutamiento. Asimismo, estas tablas también se modifican cuando los nodos vecinos informan sobre el estado de sus colas.

En palabras de los propios autores, el objetivo principal de *HTECRP* es evitar la congestión. *HTECRP* construye un árbol de enrutamiento con la mayoría de los hijos de sus nodos. Puesto que el número de hijos en un árbol de enrutamiento está limitado, el volumen de tráfico que reciben los nodos también lo está. Otro parámetro que influye directamente en el éxito en la gestión de la congestión es el conocimiento de los nodos sobre la longitud media de las colas de sus vecinos. En esta situación, cuando un nodo va a seleccionar el siguiente salto de su enrutamiento, considera dicha longitud de la cola para decidir qué nodo elegir, de forma que cuando la longitud de dicha cola es pequeña, la probabilidad de ser seleccionado como siguiente salto se incrementa.

Debido al objetivo único de autores en controlar la congestión del tráfico en el protocolo, en las simulaciones no se presentan resultados ni de distribución de energía ni de consumo, por lo que en la figura 2.48 se presenta únicamente una gráfica donde se aprecia el número de paquetes perdidos para el protocolo propuesto y otro protocolo basado en *QoS* citado por los autores.

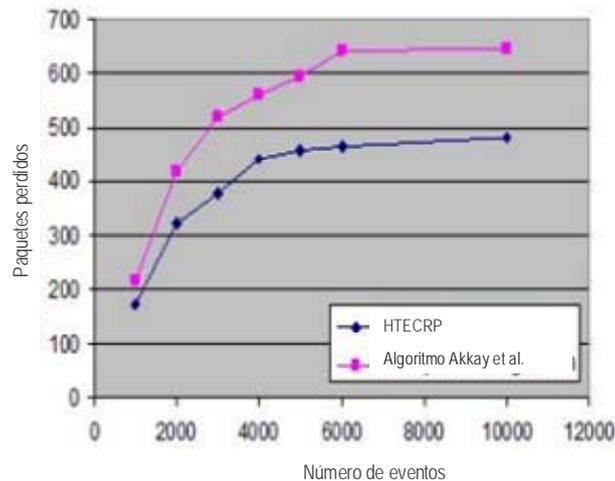


Figura 2.48. Número de paquetes perdidos en función del número de eventos en el protocolo

El protocolo propuesto controla bien, en comparación, la congestión por lo que pierde un menor número de paquetes.

Una de las razones primordiales que podemos dar para explicar el por qué los autores no indican resultados de simulaciones referentes al consumo ni a la distribución de la energía es porque el protocolo dispone de tablas de enrutamiento y mecanismos orientados principalmente en evitar la congestión, por lo que **deja en segundo plano la eficiencia energética**. El mantenimiento periódico de las tablas provoca una **sobrecarga** adicional, además de los mensajes constantes de *broadcast* para informar a los vecinos de la longitud de las colas, que si bien es una medida para evitar la congestión, provoca sobrecarga extra al protocolo. Además, se propone *HTECRP* como protocolo para tiempo real, pero viendo la sobrecarga que se impone a los mecanismos de control de la congestión, surge la duda de cuánto tiempo duraría su protocolo de tiempo real, ya que aunque sí se apliquen umbrales de energía y se construya un árbol de enrutamiento, dicho árbol se construye con fines de control de la congestión no con fines energéticos, por lo que el tiempo de vida de la red queda como *indeterminado*.

### 2.3.18 Hybrid Energy-efficient Tree-based Optimized Routing Protocol (HECTOR)

*HECTOR* [MITTON08] es un protocolo híbrido basado en árbol y energéticamente eficiente basado en dos conjuntos de coordenadas virtuales. Uno de estos conjuntos está basado en coordenadas raíz del árbol mientras que el otro se basa en distancias de saltos a lo largo de varios *hitos* (marcas). En el algoritmo, el nodo que actúa como portador del paquete lo retransmite al vecino que optimiza el ratio de coste en consumo en función del progreso en la distancia vía dichos hitos, ó entre los nodos que reducen las coordenadas de los hitos y no incrementan las coordenadas del árbol. Si no existe ningún nodo que cumpla alguna de esas dos características entonces el *forwarding* se realiza hacia el vecino que reduce la distancia en el árbol y optimiza el coste en consumo sobre el ratio de progreso en el árbol.

Debido a que la información de la posición de los nodos no siempre es factible ya que los receptores *GPS* son voluminosos, consumen bastante y son caros, ante la ausencia de estos dispositivos en los nodos es necesario incorporar algún tipo de nomenclatura basada en coordenadas virtuales para la localización geográfica. Estas coordenadas virtuales no tienen porque incorporar posicionamiento global.

Los autores se centran en el diseño de un protocolo escalable y basado en energía que garantice la entrega en los sensores sin necesidad de mecanismos conscientes del posicionamiento de los nodos. En *HECTOR*, el primer conjunto de coordenadas virtuales le permite encontrar de forma voraz un camino en la dirección de retransmisión del destino mientras que el segundo conjunto, le previene de alcanzar un punto muerto sin salida en el enrutamiento.

*HECTOR* posee las siguientes características:

- *Escalable*: excepto para el etiquetado, para realizar las tareas de decisión del enrutamiento, un nodo ha de ser consciente de la posición de sí mismo, de sus vecinos y de la posición del destino. Además, *HECTOR* no necesita mantener nada en memoria: ningún tipo de información del enrutamiento ha de ser almacenada en el nodo debido a que se añade la información de manera constante en los mensajes.
- *Libre de bucles*: debido a que el enrutamiento se realiza de forma voraz lo cual hace que siempre cualquier nodo  $s$  del camino retransmita a un nodo más cercano al destino (en el sistema de coordenadas) que el del propio emisor.
- *Entrega garantizada*: gracias a los conjuntos de coordenadas derivadas del árbol, *HECTOR* consigue entrega fiable. En el peor caso, *HECTOR* sigue el árbol únicamente, que le provee exactamente un camino entre cualquier par de nodos.
- *Eficiencia energética*: *HECTOR* selecciona el nodo que minimiza el coste a lo largo del progreso de avance hacia el destino.

Como ya se dijo anteriormente, para el proceso de enrutamiento, se siguen dos conjuntos de coordenadas  $(V, T)$ .  $V(u)$  es el conjunto de coordenadas del nodo  $u$  usadas para alcanzar un progreso en el **grafo geográfico**, limitando así el alargamiento de la longitud de la ruta, pero no garantizando la entrega por sí solas. Estas coordenadas  $V$  se emplean para computar la distancia de los saltos entre hitos.  $T(u)$  es el conjunto de etiquetas que permiten garantizar la entrega de los paquetes. En el caso de que la red esté conectada, las coordenadas  $T$  otorgan un camino entre cualquier par de nodos.

El proceso de etiquetado  $T$  se realiza a través de la construcción de un árbol. El árbol se construye de forma iterativa desde la raíz hasta las hojas. En cada paso, cada nodo recién etiquetado realiza un *query* a sus vecinos sin etiquetar y le da a cada nodo que responda una etiqueta. La distancia usada en el árbol por tanto se basa en el tamaño de la etiqueta y su

prefijo común, el cual puede dar la distancia del salto entre cualquier par de nodos de la red. Mediante la construcción de varios árboles, se puede obtener un vector de etiquetas, con una etiqueta por nodo para cada árbol.

El proceso de construcción de las coordenadas  $V$  se consigue mediante el envío de un mensaje *broadcast* de *beacon* en la red, incrementado en cada salto, distinguiendo a varios nodos como *hitos* ó *landmarks*. Desde ahí, un nodo arbitrario  $x$  conoce su vector de coordenadas virtuales  $V(x) = (h_1, \dots, h_k)$ , donde  $h_i$  es la distancia en saltos entre  $x$  y  $L_i$ .

En la *figura 2.49* se muestra un ejemplo de cómo se realiza el etiquetado  $T$  y la construcción de las coordenadas virtuales  $V$ :

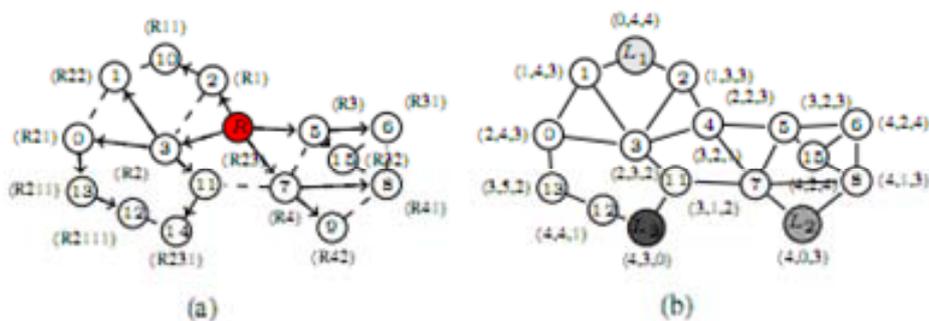


Figura 2.49. Etiquetado en HECTOR: (a) El nodo raíz es el nodo 4 y tiene la etiqueta R. El nodo 13 ha sido etiquetado con R211 porque es el primer hijo del nodo 0, el cual tiene la etiqueta R21. Las líneas punteadas indican enlaces físicos. (b) El nodo 4 tiene las coordenadas (2, 2, 3) porque está 2-saltos alejado de los hitos 1 y 2 y 3-saltos alejado del hito 3

En cuanto a la descripción del algoritmo, éste combina ventajas de ambos tipos de coordenadas. Recordemos que las coordenadas virtuales permiten minimizar la longitud del camino y que las etiquetas permiten evitar llegar a un punto muerto en el enrutamiento y garantizan la entrega. La idea básica es la siguiente. Un nodo emisor  $s$  que tiene un paquete para entregar al nodo destino  $d$ , ejecuta un algoritmo voraz de enrutamiento eficientemente energético. Para evitar caer atrapado en un mínimo local, el algoritmo de enrutamiento selecciona el siguiente salto teniendo en cuenta tanto de las coordenadas virtuales como las etiquetas. El proceso de enrutamiento es el siguiente:

7. Cuando un nodo  $u$  recibe el mensaje del nodo  $d$ , primero considera a sus vecinos en la dirección del enrutamiento, basándose en las coordenadas virtuales y en sus etiquetas.
8. Únicamente considera los nodos  $v$  para los cuales la distancia  $d_T$  frente a  $d$  es igual o menor que la distancia en el árbol entre  $u$  y  $d$  ( $d_T(v, d) \leq d_T(u, d)$ ).
9. Dicho tipo de nodos siempre existen, sin importar cómo estén conectados el nodo fuente y el destino, debido a la naturaleza convergente del etiquetado.
10. El algoritmo entonces comprueba si alguno de esos nodos también incorpora progreso respecto a las coordenadas de los hitos. Sea  $H = N_T(u) \cap \{N_V(u) \cup v \mid d_T(v, d) = d_T(u, d)\}$  el

conjunto de dichos nodos; si  $H \neq \emptyset$  entonces  $u$  selecciona su siguiente salto de entre los nodos en  $H$ , es decir, aquellos que reducen la distancia hacia el destino basándose en las coordenadas  $V$  y no incrementando la distancia de las etiquetas  $T$ .

En cuanto a las simulaciones, se compara *HECTOR* con las soluciones presentadas en la bibliografía referentes a algoritmos geográficos que no asumen información de la posición. Estos algoritmos son *VCost* [ELHAFSI07] y *LTP* [CHAVEZ07]. Con el objetivo de avanzar en sus aproximaciones, también comparan *HECTOR* con una variante de él mismo, *HECTOR'*, la cual incorpora un método estático de elección del siguiente salto en el enrutamiento. Los nodos se distribuyen en un cuadrado de  $1000 \times 1000$  siguiendo una distribución bidimensional de *Poisson* con diferentes intensidades  $\lambda$ . Los nodos pueden adaptar su radio de comunicación entre  $0$  y  $R = 200$ . Se comparan los protocolos citados anteriormente para las mismas muestras de distribución de los nodos, mismos pares de nodos fuente y destino, ambos seleccionados aleatoriamente. Los hitos y la raíz del árbol también son seleccionados aleatoriamente. Se evalúa el consumo de energía debido a sobrecarga (*Energy Consumption Overhead – ECO*) para cada algoritmo, dando una idea detallada de su eficiencia. Antes de presentar los resultados, en la figura 2.50 se muestra la topología empleada para las simulaciones:

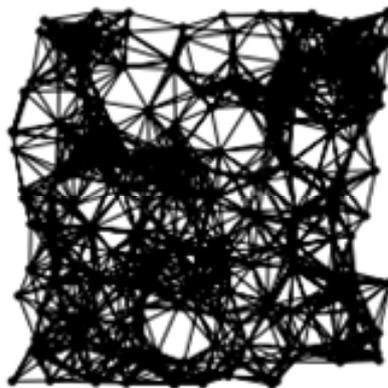


Figura 2.50. Topología homogénea

Debido a que *VCost* puede no encontrar camino y fallar, se evalúa para dicho *ECO* para cuando *VCost* tiene éxito y para cuando *VCost* falla, arrojando los resultados que podemos apreciar en la figura 2.51.

La sobrecarga de energía se grafica dependiendo de la media de los nodos y del número de hitos necesarios para construir las coordenadas  $V$ . Se observa que *HECTOR* alcanza la menor sobrecarga de entre los protocolos que garantizan la entrega, ya que *VCost* no lo garantiza. La media de nodos y el número de hitos son independientes y tienen un impacto limitado en el rendimiento de cada protocolo. Como era de esperar, *HECTOR* provee mayor sobrecarga que *VCost*. Esto es debido al proceso de enrutamiento en *HECTOR*, el cual intenta que se prograse en el árbol en cualquier paso del algoritmo. De la misma forma, la posición de la raíz el árbol es importante para minimizar el consumo de energía para unos nodos fuente y destino dados, ya que no es posible obtener un posicionamiento óptimo de la raíz para cada par fuente-destino.

**VCOST = ÉXITO**

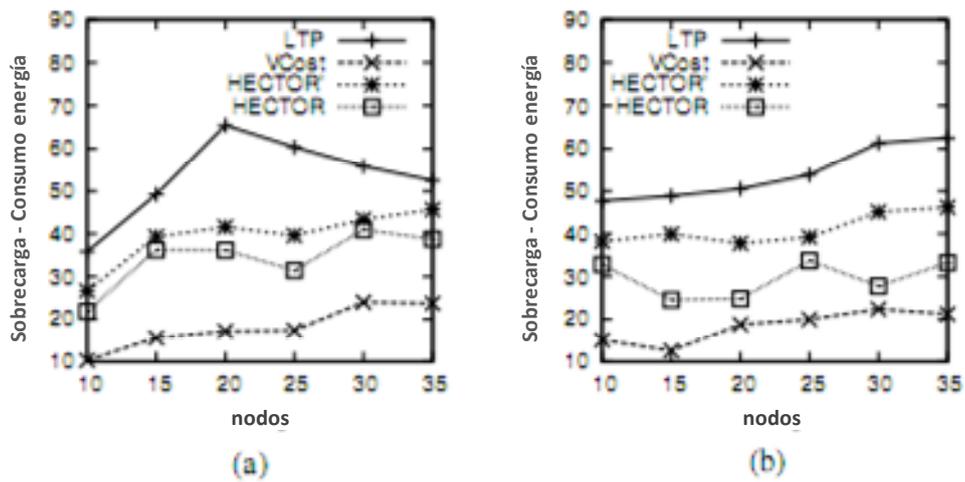


Figura 2.51. (a) 3 landmarks y (b) 5 landmarks

Cuando VCost falla, se ve claramente en la figura 2.52 como HECTOR es el algoritmo que da mejores resultados en lo que a energía se refiere, seguido de cerca por HECTOR'. Esto es debido a que, ya que HECTOR ya no sigue las coordenadas V, debido a que VCost también se basa en ellas y falla, sino que sigue las etiquetas T, al igual que LPT, hasta que encuentra algún nodo que sí puede hacerse cargo de dichas coordenadas V.

**VCOST = FALLO**

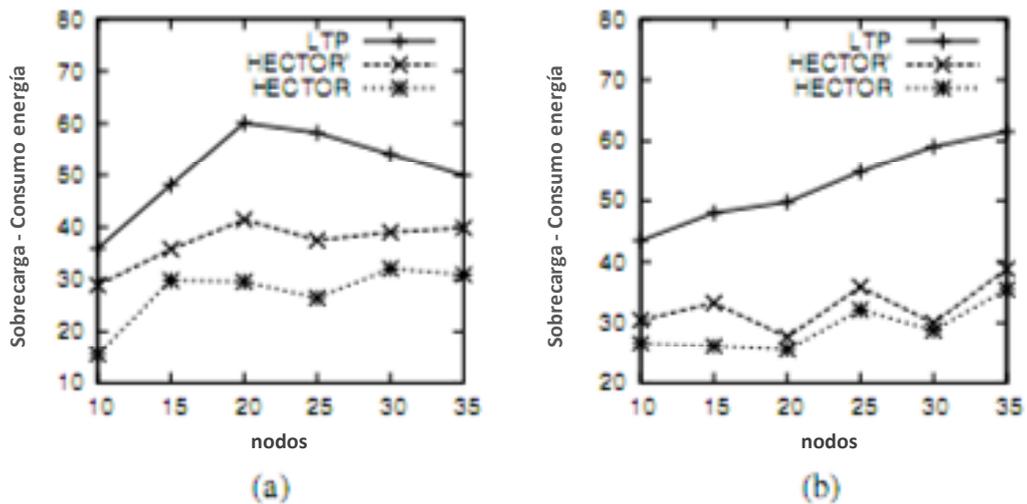


Figura 2.52. (a) 3 landmarks y (b) 5 landmarks

Los autores dictaminan que *HECTOR* es el primer protocolo de enrutamiento geográfico basado en coordenadas virtuales que es tanto eficientemente energético como fiable en cuanto a garantía de entrega. A su vez dejan la puerta abierta para futuros trabajos de mejora en lo que a construcción de las coordenadas virtuales se refiere. De hecho, el **punto débil de *HECTOR* es la propia construcción del árbol de enrutamiento**, ya que éste no se construye siguiendo unas pautas de energía residual en los nodos. Dicho árbol se forma basándose en distancias mientras que si se siguieran pautas energéticas se conseguirían mejores resultados.

En líneas generales, dejamos constancia de las **desventajas** que presenta *HECTOR*:

- Se deduce de la descripción de los autores que el protocolo debe albergarse en la rama de protocolos basados en localización. No sabemos a qué se refieren cuando hablan de híbrido, luego de ahora en adelante se le tratará como basado en localización a efectos de mecanismos comparativos.
- Debido a que se critica el uso de receptores *GPS*, asumimos que *HECTOR* no los incorpora, luego la forma en la que ubica a sus vecinos y les etiqueta con valores crecientes de la misma no queda claro, ya que no se citan ni mecanismos de comparación de la señal entrante ni triangulación.
- Aun quedando patente su eficiencia energética en comparación con los protocolos que se han elegido, se echa muy en falta, y más siendo el primer protocolo geográfico que aplica coordenadas virtuales, una comparación con opciones jerárquicas o clusterizadas, ya que si se asume que el protocolo es híbrido, se deben asumir dichas comparaciones.
- Ya que no se han hecho las comparaciones que demandamos en el punto anterior, podemos remarcar que una opción que aglutine jerarquización y construcción de un árbol con los mecanismos de etiquetado de *HECTOR* podría ser una buena línea de actuación.

### 2.3.19 Energy-Balancing Unequal Clustering Protocol (EB-UCP)

*EB-UCP* [YANG09] consigue un buen rendimiento en términos del tiempo de vida de los sensores mediante *clustering* desigual y equilibrando la carga energética entre todos los nodos. Se emplea un algoritmo de *clustering* desigual desde una perspectiva probabilística para conseguir dichos resultados. De esta forma, los *clústeres* más cercanos al sumidero tienen menor tamaño que los que están más alejados, por lo que los *CH* más cercanos pueden ahorrar más energía en su tarea de encaminamiento *inter-clúster*.

De la misma forma, la distribución de los sensores se realiza de acuerdo a un algoritmo de balanceo energético por capas, por lo que el consumo de energía en cada capa es prácticamente el mismo. Por último, un mecanismo de transmisión energéticamente eficiente es la base de la mejora respecto a *EEUC* y *LEACH*.

El concepto de capa en *EB-UCP* no es el mismo que en otros protocolos. Todos los nodos se asume que están dispensados en un área circular de radio  $R$ . El único *sink* existente está situado en el centro de dicha área, por lo que cada capa contiene un número particular de *clústeres*, tal y como se muestra en la *figura 2.53*:

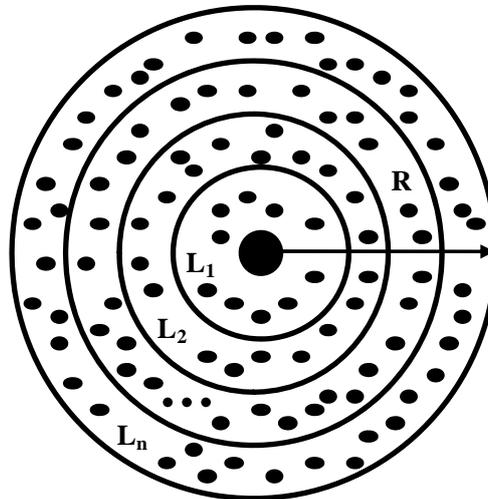


Figura 2.53. El área circular consistente en coronas

La idea clave del algoritmo de *clustering* es utilizar *clústeres* desiguales para solucionar el problema del *hotspot*, como ya se vio en el protocolo *EEDUC*. A las capas se les asignan diferentes probabilidades de acuerdo a la distancia con el sumidero. Las capas más cercanas a éste poseen mayor probabilidad que las lejanas. De esta forma, bajas probabilidades indican menor número de *CHs*, lo que implica que cada *CH* ocupará una mayor área que contendrá normalmente un mayor número de nodos *non-CH*. Por ello, es lícito pensar que los nodos más cercanos al *sink* recibirán mayores probabilidades para ser escogidos como *CH* en el contexto del equilibrio en la disipación energética.

La transmisión de datos en la red se realiza mediante iteraciones de *intra-clúster encaminamiento* e *inter-clúster encaminamiento*. La organización del *intra-clúster encaminamiento* es similar a *LEACH*, con sus ventajas e inconvenientes, por lo que se omitirá su explicación. En lo referente al *inter-clúster encaminamiento*, el algoritmo en pseudocódigo se muestra en la *figura 2.54*.

Con el objetivo de conseguir una disipación uniforme entre los *CHs*, el *CH i* selecciona un *CH* transmisor que posea la máxima energía residual. Durante este proceso, el *CH i* ha de intercambiar información acerca de las energías remanentes en cada uno de los *CH* candidatos (líneas 7-10 pseudocódigo *figura 2.54*) para que, en caso de que exista más de uno con la misma energía residual, seleccione uno de ellos de forma aleatoria. Una vez ha seleccionado su *CH* transmisor, envía la información a la capa superior a través de él y se repite el proceso hasta alcanzar el sumidero.

```

1.   FinalCH_msg (id,layer_id);
2.   Rec_msg (FinalCH_msg) from q;
3.   if (q.layer_id == i.layer_id+1)
4.       Add node q to i.Parent;
5.   else if (q.layer_id == i.layer_id-1)
6.       Add node q to i.Children;
7.   if (Rec_msg (InqResEnergy) from Parent)
8.       Send_msg (id,i.ResEnergy,layer_id);
9.   if (Rec_msg (AckInqResEnergy)from Children)
10.      Update (i.Children.ResEnergy);
11.  while (Data_arrival == TRUE){
12.      if (Rec_msg (Forwarddata)from Parent){
13.          u = MaxResEnergy (i.Children);
14.          Send_data (u, Forwardingdata);
15.      }else{
16.          u = MaxResEnergy (i.Children);
17.          Send_data (u, i.data);
18.      }
19.  }

```

Figura 2.54. Algoritmo de transmisión inter-clúster

Esta forma de enrutamiento hasta el sumidero puede otorgar mejoras respecto otras estrategias pero no posee mecanismos sistemáticos a la hora de dormir los sensores o sincronizarse entre ellos para los envíos, por lo que la mejora alcanza peores cotas que en el protocolo propuesto en la presente tesis.

A la hora de analizar los resultados de las simulaciones, los autores proponen analizar el rendimiento en el tiempo de vida y número de rondas respecto a un parámetro  $\alpha$ , que dictamina el coeficiente de agregación en el *CH*, en otra palabras, un coeficiente de  $\alpha = 0.1$  indica que el *CH* realiza un forward del 10% de la carga del *clúster*. Los resultados para distintas parametrizaciones los podemos observar en las gráficas de la figura 2.55.

Se ve claramente como la mejora es proporcional al valor del coeficiente de agregación puesto que cuanto menor es dicho parámetro y más se fusiona, mejores resultados se obtiene, aunque estos no son muy sustanciales frente a *EEUC*.

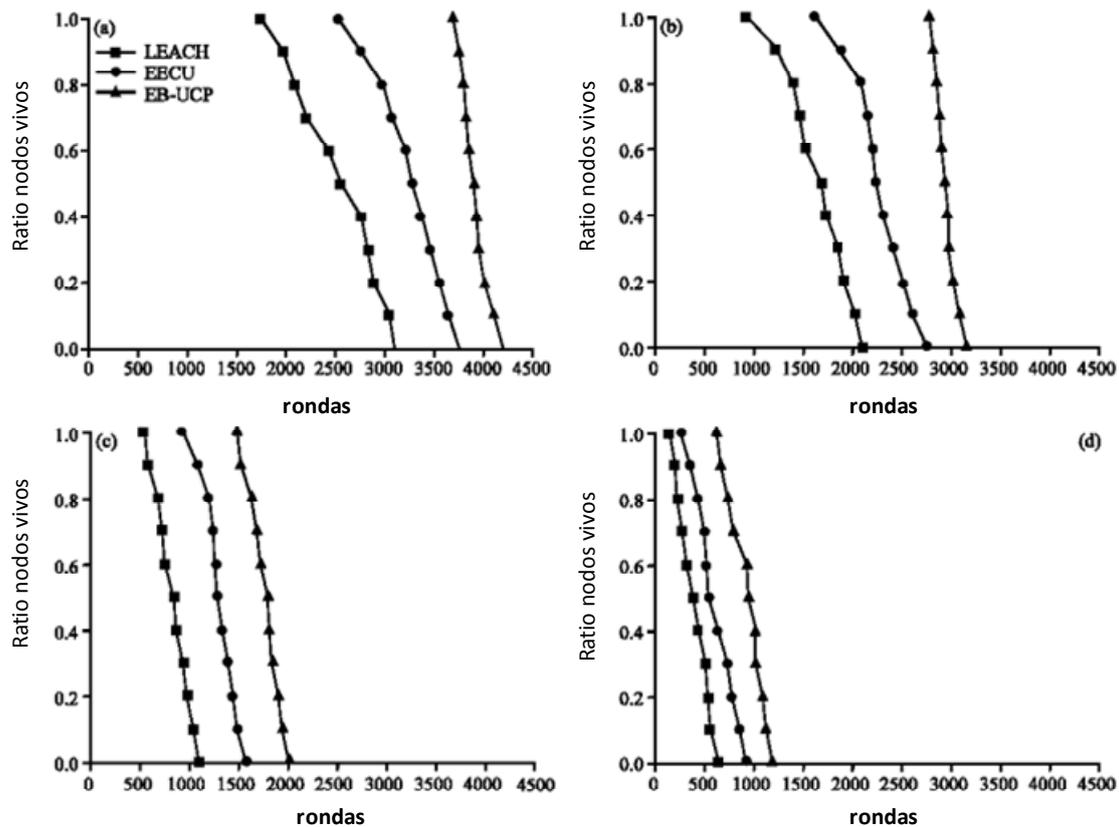


Figura 2.55. Comparación de rendimiento para (a)  $\alpha = 0.1$ , (b)  $\alpha = 0.3$ , (c)  $\alpha = 0.7$  y (d)  $\alpha = 1.0$

### 2.3.20 Grid-clustering Routing Protocol (GROUP)

*GROUP* [YU06] es un protocolo de enrutamiento en malla *clusterizada* principalmente orientado a proporcionar escalabilidad. El sumidero proactivamente, dinámicamente y aleatoriamente construye una malla de *clústeres* con el objetivo de enrutar los mensajes y los paquetes de datos. Los autores eligen una estructura en malla para los *CHs* basándose en dos consideraciones:

1. La estructura en malla puede garantizar una buena distribución de los *CHs*.
2. El enrutamiento entre *CHs* es sencillo cuando se trata de una estructura en malla y los *CHs* pueden seleccionar una ruta de las múltiples posibles para encaminar los datos hacia los sumideros basándose en energía residual y equilibrio de carga.

Únicamente una pequeña porción de los nodos participarán en la elección de los *CHs*. Un nodo es elegido como *CH* en un *clúster* y todos los *CHs* forman una malla virtual de *clústeres*. Las consultas (*queries*) de datos serán transmitidas desde los sumideros a todos los nodos vía sus *CHs* y la información que encaje en dichas *queries* será enrutada de vuelta a los sumideros

mediante dichos *CHs*. Las políticas de agregación serán aplicadas, dependiendo de la aplicación, en los *CHs*.

El proceso de construcción de la malla sigue los siguientes puntos:

- Todos los sumideros de la red elegirán un sumidero como *sink* primario (*Primary Sink – PS*), el cual inicia el proceso de construcción de la malla, basándose en su localización. El *PS* se encuentra más cerca al centro de la red que cualquier otro *sink*, con el fin de minimizar el tiempo de construcción de la malla. En la *figura 2.56*, el *sink S2* es el *PS*:

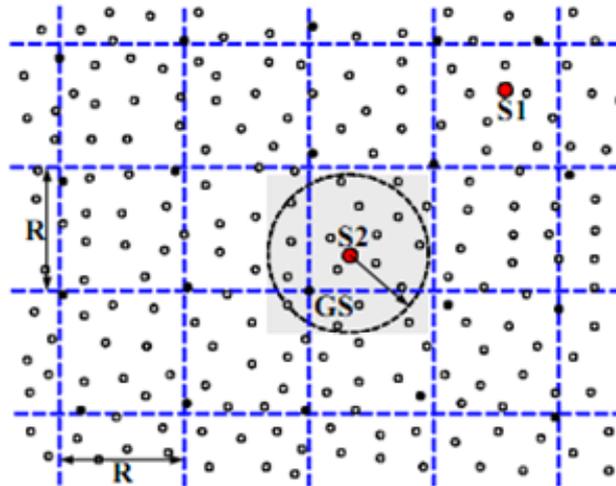


Figura 2.56. Ejemplo de clúster en malla

- Cada  $T$  segundos, el *PS* inicia el proceso de construcción de la malla mediante un *broadcast* en su radio de acción del mensaje *GS-election*. El objetivo de este mensaje es seleccionar un nodo como semilla de la malla (*Grid Seed – GS*) de entre todos los vecinos del *PS*. Todos los nodos que reciban dicho *broadcast*, decidirán si contestan o no.
- Cuando el *PS* envía el comando *GS-election*, pone en marcha un *timer*. Si el *PS* no recibe ninguna contestación por parte de sus nodos vecinos antes de que el temporizador expire, reenviará el mensaje.
- De entre todos los nodos que le contesten, el *PS* elige a uno de ellos como *GS* dependiendo de su energía residual. Ante igualdad de energía residual, el candidato que haya respondido más pronto, será el elegido.
- El *GS* se encuentra en uno de los puntos de unión de la malla de *clústeres* que se está construyendo actualmente. Para cada uno de estos puntos de cruce, *GROUP* selecciona un sensor, que esté más cerca de ese cruce que el resto de nodos, para actuar de *CH*.
- El nuevo *GS* realiza el *broadcast* de un paquete de *CH-election* a sus vecinos, para elegir a sus cuatro *CHs* inferiores. Los vecinos que reciban dicho paquete, calcularán la distancia a los cuatro puntos de cruce. Si la distancia a uno de esos cuatro puntos es inferior a  $0.5R$ , el nodo competirá con el resto de sus vecinos por ser *CH* enviando un *broadcast* de *CH-candidate*.

En lo referente al proceso de comunicación (*forwarding*) de datos, se organiza en tres puntos:

- Una vez que un sensor recibe el *query* por parte de su *CH*, comprueba la propia consulta y los datos recogidos.
- Si los datos recogidos cuadran con la *query*, envía los datos a su *CH* mediante comunicación de corto alcance.
- El paquete de datos será retransmitido recursivamente por el *CH* hacia arriba por sus *CHs* superiores hasta que alcance al sumidero que generó la consulta. La *figura 2.57* muestra este proceso:

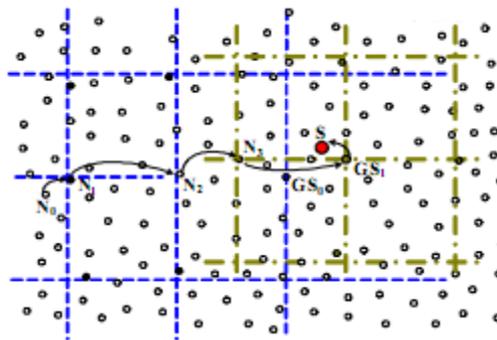


Figura 2.57. *Forwarding* de datos durante la construcción de la malla

Ambos procesos, junto con el proceso de recuperación de errores, en el cual si algún *CH* falla se busca nodos vecinos mediante paquetes de *CH-search* para que reemplacen al nodo caído, intercambian cantidad de mensajes en el proceso de construcción, los cuales hacen que se consuma mucha energía en los procesos. La forma en la que se organiza la topología de malla, con *CHs* en las esquina y actuando de *relays* hasta el sumidero obliga a que los nodos conozcan donde se encuentran dichas esquinas o cruces mediante el empleo de *GPS*, para localizar la posición de referencia del resto de *CHs*, o triangular sus posiciones. En ambos casos, se incorpora más **sobrecarga** al protocolo que puede disminuir en gran medida su eficiencia.

Los resultados de rendimiento conseguidos se muestran en las gráficas de la figura 2.58.

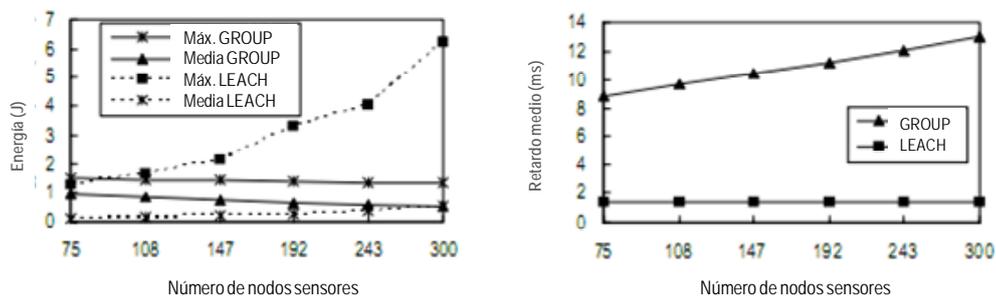


Figura 2.58. Consumo de energía y retardo medio entre GROUP y LEACH

Respecto al consumo de energía, analizando el máximo consumo y la media, se puede concluir de manera eficaz si un protocolo distribuye el consumo energético o no. Para el caso de *LEACH*, se observa claramente como su máximo consumo es alarmantemente alto en comparación con su media, lo que hace entrever, cosa que ya se sabía, que *LEACH* no consigue

una buena distribución del consumo. Para *GROUP*, se obtienen buenos resultados en cuanto a distribución, pero en comparación a *LEACH* para el caso medio, las mejoras son mínimas.

Si se observa ahora el *delay* punto-a-punto para ambos protocolos, se muestra que *GROUP* obtiene pésimos resultados, que los autores achacan a que *GROUP* trabaja con *multi-hop encaminamiento* y *LEACH* con *single-hop*, estrategia que ya se ha demostrado que es mejor en multitud de trabajos. Cuantos más nodos tenga la red, *GROUP* obtendrá peores resultados, lo que da mucho que pensar.

### 2.3.21 Base-station Controlled Dynamic Clustering Protocol (BCDCP)

*BCDCP* [FAPOJUW09] se propone como solución para el problema de gestión de recursos energéticos a la hora de construir los *clústeres* y distribuir las planificaciones para el enrutamiento. Éste asume desde un principio tres puntos clave para el desarrollo del protocolo:

1. El sumidero se asume que posee energía y capacidad de cómputo en abundancia. Esta asunción es válida siempre, ya que el sumidero está conectado directamente a red y con recursos ilimitados en comparación al resto de los nodos en la *WSN*.
2. El sumidero se asume que conoce la **posición y localización de todos los nodos de la red**. Como ya se ha indicado en otros protocolos anteriores, esto no siempre es posible ni viable, porque además de que el sumidero requeriría de *GPS*, cosa no muy problemática, lo problemático sería incorporar receptores *GPS* a los sensores, lo que no siempre es adecuado ni competente para los casos que estamos abordando.
3. Se asume que los nodos tiene total control sobre su potencia de transmisión, lo que se ha subrayado varias veces que es un buen mecanismo para controlar el consumo en las comunicaciones de los sensores dependiendo del rol que estén desempeñando.

Así mismo, el sumidero posee las siguientes responsabilidades:

- **Recopilar los niveles de energía de los nodos**, es decir, que los nodos deben enviar el estado de su batería *on-demand*, lo que puede provocar **sobrecarga** en las comunicaciones ya que éstos deben enviar al sumidero.
- Realiza las tareas de **clusterización de los nodos**, agrupándolos según convenga.
- Crea las **planificaciones TDMA** y las distribuye a los implicados.
- **Forma el árbol de recubrimiento mínimo de CHs** donde únicamente un *CH* se comunica con el sumidero.

Los pasos que sigue el sumidero en *BCDCP* se enumeran a continuación, mostrándose de una manera gráfica en la figura 2.59.

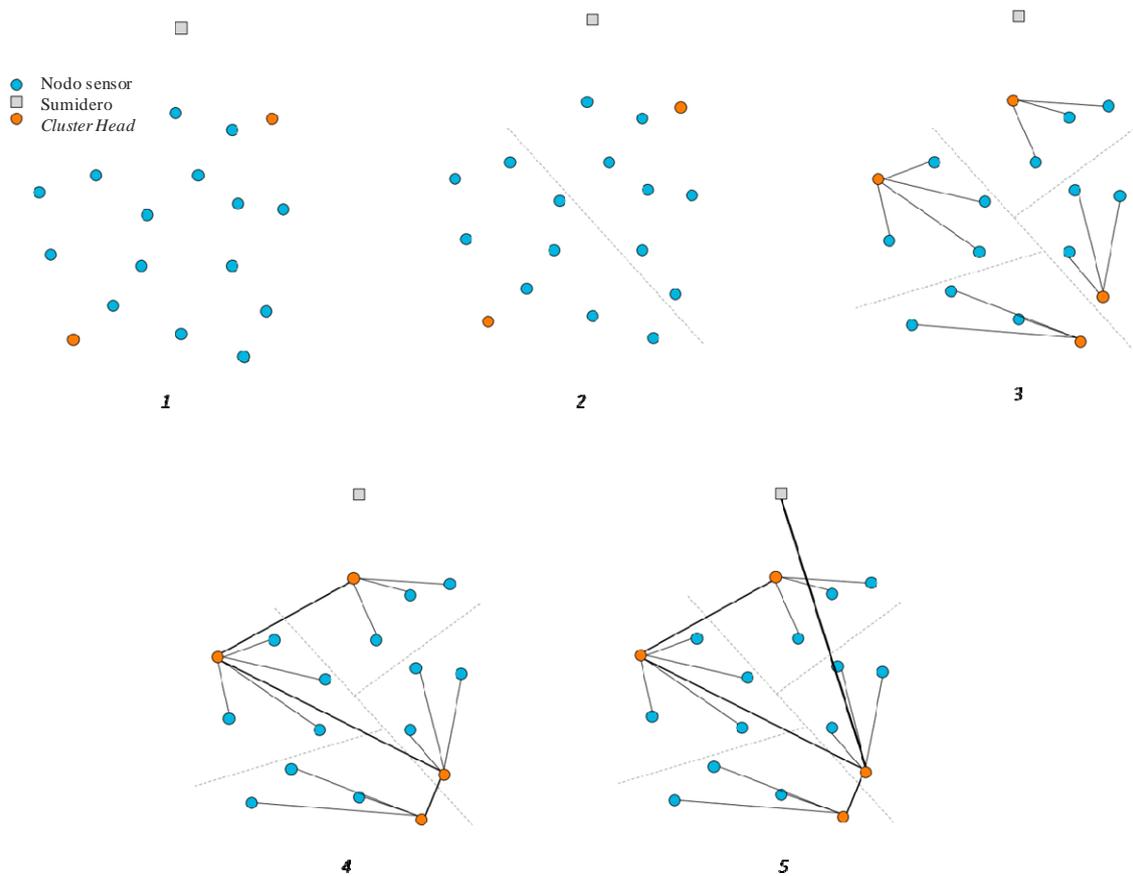


Figura 2.59. Proceso del algoritmo en el protocolo para las fases de Configuración y Comunicación

1. Se eligen los dos nodos más separados como *CHs* en potencia.
2. Se dividen los nodos basándose en parámetros de proximidad y se balancean los *clústeres*.
3. Se itera.
4. Se conectan los *CHs* mediante el *MST*.
5. Se selecciona un *CH* aleatoriamente para comunicarse con el sumidero y se crea y distribuye la planificación *TDMA*.
  - a. Con esta planificación *TDMA*, los nodos envían sus datos a los *CHs*.
  - b. Los *CHs* realizan las tareas de fusión y compresión y posteriormente la información se encamina al sumidero mediante el *CH* seleccionado.

Los resultados de las simulaciones para la disipación media en comparación con *LEACH*, *LEACH-C* y *PEGASIS* se muestran en la *figura 2.60*.

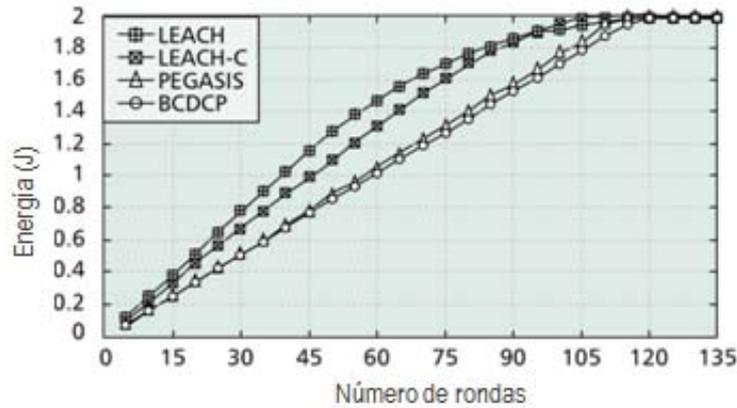


Figura 2.60. Disipación energética media en comparación con otras estrategias basadas en *clustering*

Se puede ver como *BCDCP* mejora a *LEACH* y a su variante pero la mejora en comparación a *PEGASIS* no es tanta. En el siguiente apartado, cuando se analice la mejora al propio *BCDCP*, *DMSTRP*, se detallarán los aspectos problemáticos de estos dos enfoques.

### 2.3.22 Dynamic Minimal Spanning Tree Routing Protocol (*DMSTRP*)

El protocolo *DMSTRP* [HUANG06] mejora a *BCDCP* mediante la construcción de ***MSTs*** (*Minimum Spanning Trees*) en vez de los *clubs* que conectan los nodos en los *clústeres*. La idea principal es usar estos *MSTs* para reemplazar dichos *clubs* en dos capas de la red: *intra-clúster* e *inter-clúster*. Debido a que los *clubs* no son más energéticamente eficientes que los árboles de expansión para conectar nodos en una red de gran extensión, *DMSTRP* es una solución elegante para redes amplias, según los autores.

La arquitectura de *DMSTRP* es similar a la de *BCDCP* pero con algunas diferencias. En *BCDCP*, los *CHs* se conectan mediante un árbol en vez de mediante *clubs*. *DMSTRP* mejora este aspecto conectando los nodos en los *clústeres* mediante *MSTs*, como se puede observar en la *figura 2.61*. En cada *clúster*, todos los nodos, incluyendo al *CH*, están conectados por un *MST*, dejando al *CH* la tarea de recolectar toda la información correspondiente al árbol entero. Los restantes *CHs* conectados mediante otro *MST* realizan el *forwarding* hacia el sumidero. Los aspectos de fusión se realizan a lo largo de dicho árbol.

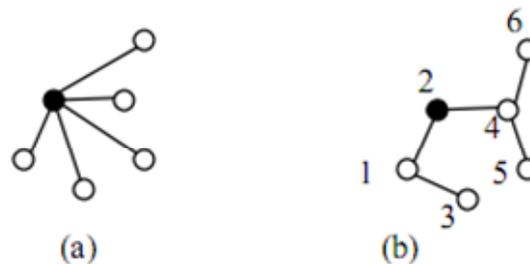


Figura 2.61. Cambio en la topología básica de (a) club en *LEACH* y *BCDCP* a (b) *MST* en *DMSTRP*

La distancia media de transmisión para cada nodo se puede ver reducida por el empleo de *MSTs* en vez de *clubs*. Sin embargo, los árboles **incrementan la disipación de energía** en recepción y fusión de la información, además de incorporar **más saltos** dentro del propio *clúster*. Por ello, únicamente cuando la distancia media de transmisión se ve reducida mayoritariamente por el *MST*, *DMSTRP* es más energéticamente eficiente que sus homólogos. Conforme la red de sensores se distribuya en una superficie más amplia, la distancia de transmisión que nos ahorremos con el *MST* será mayor. En los *clubs*, la información necesaria para enrutar los datos es simple y cada nodo únicamente necesita saber de su *CH* y enviarle los datos. En los árboles, cada nodo debe saber el siguiente nodo al que debe enviarle información, luego la encargada de notificar el *status* del *MST* será el sumidero.

Se debe abordar el **complejo** esquema de enrutamiento en árbol en *DMSTRP* de una forma adecuada y distribuyendo eficientemente los tiempos para evitar las colisiones y hacer del *MST* una tarea eficiente en términos de tiempo. Por ello, el sumidero genera la planificación y envía dicha información a los *CHs*. Son los *CHs* los que notifican a sus miembros sobre la información de enrutamiento que deben seguir. Es entonces cuando cada sensor encamina sus datos al *CH* mediante el uso de esta información sobre el *forwarding*.

Lo detallado hasta ahora sobre la topología y enrutamiento de *DMSTRP* posee **puntos débiles**, los cuales pasamos a argumentar a continuación:

- Como se comentó en párrafos anteriores, la construcción de un *MST* dentro del propio *clúster*, creando así diversos saltos desde los sensores hasta llegar al *CH*, **no** es eficiente en términos energéticos. Únicamente cuando se habla de redes de sensores de muy larga extensión, realizar un *encaminamiento multi-hop* dentro del *clúster* tiene sentido. En la inmensa mayoría de casos, las distancias dentro del *clúster* no son tan grandes como para rentabilizar el sobrecoste de la construcción de un *MST* en el *clúster*. Anteriores aproximaciones bibliográficas, incluso el propio EDETA, concluyen que la mejor estrategia es el enrutamiento en *1-hop* dentro del *clúster*, por motivos de disipación y *delays* en las transmisiones.
- Los propios autores recaen en la precaución que hay que tener con la construcción y el manejo de estructuras tan complejas en su formación como son los *MSTs*. Ya no solo el construirlos y gestionar cómo se comunican a través de ellos los nodos, sino también mantenerlos aún cuando ocurran fallos en los nodos. Es ahí donde entra el principal problema; la ausencia de tolerancia a fallos. Debido a que se construyen *MSTs* dentro del *clúster* y fuera de él, para comunicar a los *CHs* en el *inter-clúster encaminamiento*, ante la caída o el mal funcionamiento de algún nodo, se debe reconstruir el *MST* entero, ya que puede que ya no sea mínimo, y la reconstrucción no es barata en términos energéticos.
- Además, en ningún momento se habla de roles ni la rotación de ellos, luego elegir a los mismos nodos dentro del árbol de mínima expansión acabará con el agotamiento de la

energía de siempre los mismos nodos, lo que perjudica directamente al **tiempo de vida** de la red.

- La topología de *DMSTRP* se puede definir como topología de árbol-árbol en dos niveles, totalmente compleja. El sumidero es la que computa las planificaciones y las envía a los *CHs*, los cuales la difunden hacia abajo. Debido a la estructura en dos niveles tan compleja de *DMSTRP*, el sumidero gastará mucho tiempo de computación para calcularla. De la misma manera, la propia planificación, al ser enviada desde el sumidero, si se quiere incorporar mecanismos de *sleep* en los nodos, se debe coordinar el envío de esta información de arriba abajo, con el tiempo que tienen los nodos para dormirse, lo que lo hace poco eficiente si encima hablamos de árboles de gran extensión, como se demanda en *DMSTRP*.
- Cuando hablamos de *MSTs* hablamos de estructuras complejas y de construcción laboriosa luego si se quiere incorporar en el *background* del protocolo un algoritmo de gestión de *MST*, se va a añadir complejidad de cálculo a los nodos participantes, tanto en un nivel como en otro de los árboles. Esto implica **mayor memoria y mayor HW** de computación en los propios nodos, lo que contradice el principio fundamental de las *WSNs*, bajo consumo y bajo coste.

Aún viendo los problemas que la topología específica y los *MSTs* originan, pasamos a ver los resultados experimentales de *DMSTRP* frente a otras estrategias, en este caso *LEACH* y *BCDCP* (figura 2.62).

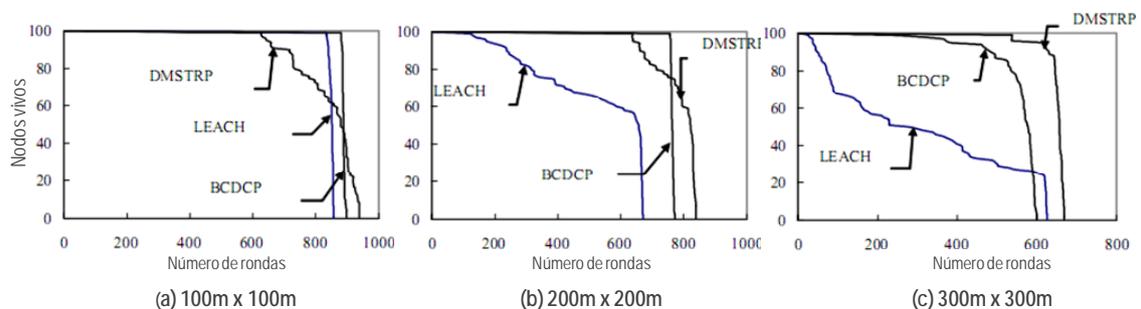


Figura 2.62. Tiempo de vida de la red para distintos tamaños en la red de sensores

Se ve claramente como *DMSTRP* no consigue tan buenos resultados cuando la red de sensores es de tamaño reducido, donde para una red de  $100 \times 100$  incluso se llega a reducir el número de nodos de la red **antes** que en *LEACH* y *BCDCP*. Esto es debido a que no sale rentable la construcción de los *MSTs* cuando la red tiene este tamaño y los nodos se agotan prematuramente.

Para el caso de  $200 \times 200$ , *DMSTRP* sigue siendo menos eficiente que *BCDCP*, ya que comienza a disminuir su número de nodo antes que éste, por los mismos motivos que en el caso anterior. *LEACH* consigue los peores resultados de los tres.

En el último caso de  $300 \times 300$ , *DMSTRP* comienza a conseguir mejores resultados que *LEACH* y *BCDCP*, aunque el número de rondas totales ha **disminuido** bastante. Esto quiere decir que aunque *DMSTRP* sea mejor que las estrategias con las que se ha comparado para una red de  $300 \times 300$ , para redes de distinto tamaño sea peor que otras estrategias que mantienen una eficiencia más constante y no únicamente para redes de un tamaño dado. La topología compleja que se instaura en *DMSTRP* como predeterminada no nos parece viable para una red en la que se mire al detalle el consumo, ya que no solo conlleva problemas de gestión de estas estructuras sino también que no se puede añadir mecanismos intrínsecos a dicha topología, como seguridad o tolerancia a fallos, ya que la propia arquitectura posee una complejidad demasiado elevada.

### 2.3.23 Power Efficiency Routing Protocol (PER)

Los autores proponen tres formas para disminuir el consumo en los nodos; la creación de un algoritmo de clusterizado para redes heterogéneas, un modelo de energía matemático para optimización de consumo (*EOR-tree*) y una solución para cubrir *agujeros* en la *WSN*.

*PER* [MA08] adopta una arquitectura jerárquica. La **capa de nivel alto** es de comunicación simétrica, donde los *headers* se comunican entre sí, mientras que la capa de nivel bajo es de comunicación asimétrica, entre *headers* y sensores. Se realizan las siguientes asunciones:

- No existe colisión en las comunicaciones ya que los dos niveles de comunicación emplean diferentes canales para ello pero sí existen colisiones *intra-clúster*.
- Los *headers* y los sensores no se mueven.
- Los nodos *CH* poseen suficiente batería, capacidad de computación y una antena superior.
- Los sensores normales son baratos, con batería limitada y un pequeño rango de transmisión.

Esto contradice el contexto en el que se está situando las *WSNs* en general, donde la red no se considera heterogénea sino que se quiere que todos los sensores involucrados en la red posean bajo coste y alta eficacia en sus labores. Si se parte de la base de que va a haber diferencia de tecnologías que permitan que unos sensores posean un *HW* más avanzado que otros, las comparaciones con enfoques anteriores no van a ser posibles.

*PER* considera la creación de *clústeres* con un miembro encargado de su gestión y comunicación con la capa superior. La diferencia de conceptos entre el *clúster* que se forma en *PER* y los *clústeres* en otros protocolos radica en que el *clúster* está organizado en anillos que diferencian los distintos radios de transmisión que el *CH* puede adoptar. En el principio de la creación del *clúster*, el *CH* puede escuchar únicamente a los sensores que se encuentran en el primer anillo. Posteriormente, va incrementando su rango de comunicación con el resto de anillos y recopilando la información de los sensores de dentro a fuera del *clúster*. Dentro de un anillo, los sensores seleccionan un tiempo de *delay* aleatorio para enviar los paquetes de datos. Los sensores que consigan enviar satisfactoriamente los datos, serán notificados por el

CH. Los que no hayan podido enviarlos, debido a problemas de **colisión**, deberán intentarlo de nuevo.

Esta estructura de *clúster* se muestra en la *figura 2.63*.

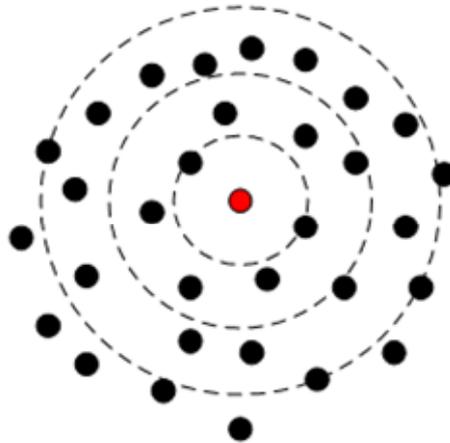


Figura 2.63. El CH y sus nodos sensores en el clúster. Las líneas punteadas denotan los anillos de comunicación

El *EOR-tree* es un árbol de retransmisión que se construye con el objetivo de subir los paquetes desde los sensores hasta el CH. Este árbol ha de satisfacer el principio de Mínimo Consumo Energético, que conlleva usar el mínimo número de saltos para retransmitir un paquete. Debido a que el árbol ha de cumplir dicho principio, provoca que se congestionen los nodos en las zonas donde existen agujeros en la red. Este problema se puede observar en la *figura 2.64*.

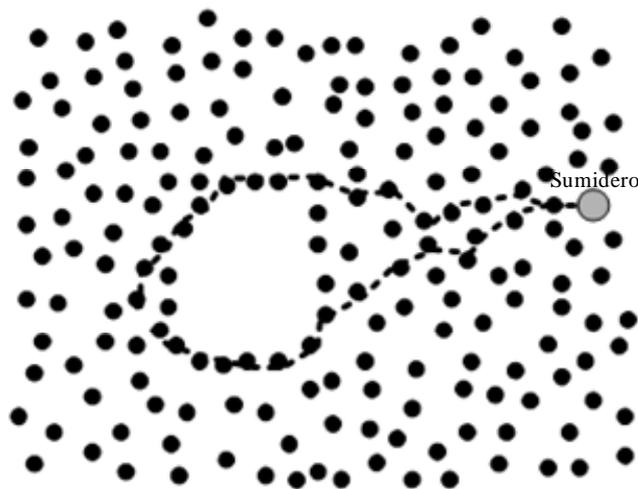


Figura 2.64. Congestión alrededor de un *agujero*

En *PER* se propone un método circular para solucionar dicho problema. El agujero se cubre por un círculo imaginario que denota que los nodos dentro de dicho círculo únicamente puedan actuar de retransmisores para nodos que también se encuentren dentro del círculo y no para nodos exteriores.

Un ejemplo de esta solución se muestra a continuación en la figura 2.65:

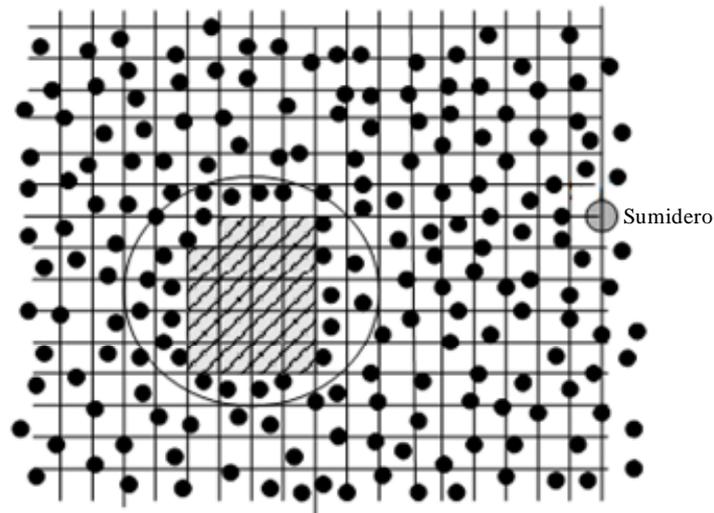


Figura 2.65. Método circular para solventar el problema de la congestión alrededor de los agujeros en la red

Para evaluar la eficiencia de *PER*, se ha comparado con *Difusión Directa* ya que ambos emplean *IEEE 802.11*. Debido a ello, no se pueden sacar conclusiones acerca de *PER* sobre los anteriores protocolos propuestos ya que se emplean bajo *IEEE 802.15.4*. En la gráfica de la figura 2.66 se muestran los resultados a modo de ejemplos empíricos.

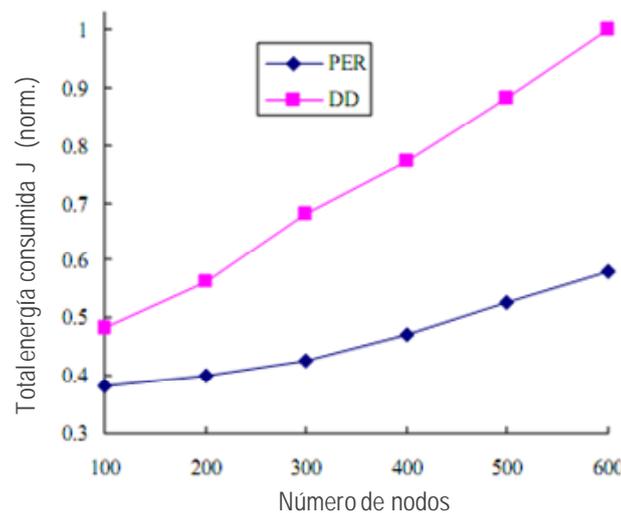


Figura 2.66. Normalización del consumo de energía total de PER y DD

Se ve claramente como *PER* mejora en gran medida a *Difusión Directa*.

### 2.3.24 Distributive Energy Efficient Adaptive Clustering (DEEAC)

DEEAC [SAJJANH07] es un protocolo adaptativo desde el punto de vista del ratio de entrega de datos y energía residual de los nodos de la red. Motivado por LEACH, como multitud de protocolos, DEEAC extiende su concepto de selección de *clúster* mediante modelo estocástico, controlándolo con un algoritmo para WSNs con variaciones espacio-temporales de los ratios de entrega para diferentes regiones. Las simulaciones indican que DEEAC distribuye un 50% mejor la energía que LEACH y por lo tanto mejora el tiempo de vida de la red.

Las regiones de la red que tienen gran ratio de generación de datos se denominan puntos calientes. El valor asociado a dicho punto caliente es un parámetro que indica el ratio de generación de datos de ese nodo en relación al resto de la red. DEEAC trata de optimizar el consumo de energía asegurándose de que los nodos que corresponden a esas regiones tengan una alta probabilidad para ser CHs. Dichos nodos que pertenecen a puntos calientes, de los cuales se espera que transmitan datos muy frecuentemente, lo hagan en distancias cortas, consiguiendo así el comedido equilibrio energético. DEEAC selecciona un nodo para ser CH dependiendo de su valor de punto caliente y de su energía residual.

Como ya se ha explicado, la idea principal del algoritmo es elegir a los nodos como CHs basándose en el criterio de valor de *hotness* (punto caliente) y su energía residual. Esto se consigue mediante el cambio del umbral  $T(n)$  elegido en LEACH (expresión 2.3) por el actual de DEEAC, calculado según (2.4):

$$T(n) = \begin{cases} \frac{P}{1 - P(r \bmod \frac{1}{P})}, & \text{Si } n \in G \\ 0, & \text{en otro caso} \end{cases} \quad (2.3)$$

$$T(n) = \begin{cases} k \times \frac{E_{res}}{E_{est\_net}} \times factor\_hotness & \end{cases} \quad (2.4)$$

Este cambio de criterio podría ser interesante ya que mediante el pequeño cambio en ciertos parámetros fundamentales del algoritmo, se pueden conseguir cambios eficientes en el comportamiento del algoritmo que deriven en buenos resultados globales, mejorando en gran medida la eficacia en el ahorro energético del protocolo.

DEEAC emplea una aproximación de la parametrización de los puntos calientes para adaptarse a las variaciones temporales del ratio de generación de datos. El factor de *hotness* para un nodo es su ratio de generación de datos en relación al de la red con una visión global, por lo que definen el ratio  $R$  como (2.5):

$$R = \frac{N_{used}}{N_{alloc}} \quad (2.5)$$

$N_{used}$  y  $N_{alloc}$  con valores en relación a la planificación TDMA en los nodos. Mediante la definición de  $H_{last\_5\_avg}$ , sumatorio de los últimos 5 valores de  $R$ , y  $H_{avg\_node}$ , sumatorio de todos los  $R$  calculados, así como la estimación de los valores de *hotness* de los CHs,  $H_{avg\_network}$ , se define una función que calcula el factor de *hotness* total (expresión 2.6), que es el expresado en el umbral mostrado anteriormente por DEEAC y que, tomando en cuenta todos estos valores variables en los nodos, obtiene un parámetro en función de los mismos, que aplica para la selección de los CHs.

$$hotness_{factor} = \left( \frac{H_{medio\_red}}{H_{medio\_red}} + \frac{H_{Últ.5medio}}{H_{medio\_red}} \right) \div 2 \quad (2.6)$$

Reiteramos que nos parece una buena idea el parametrizar de alguna forma valores variables en los nodos para tomarlos en cuenta a la hora de decidir la probabilidad con la que un nodo será CH, ya que dicha probabilidad no siempre debe ser la misma ni tener una distribución simple.

A la hora de realizar las comparaciones con LEACH, se ha realizado dos experimentos: uno relacionado con la cantidad de energía disipada por ambos protocolos y otro relacionado con cómo afecta el cálculo del umbral de *hotness* para la elección de CH, arrojando los resultados que podemos apreciar en la gráfica de la figura 2.67.

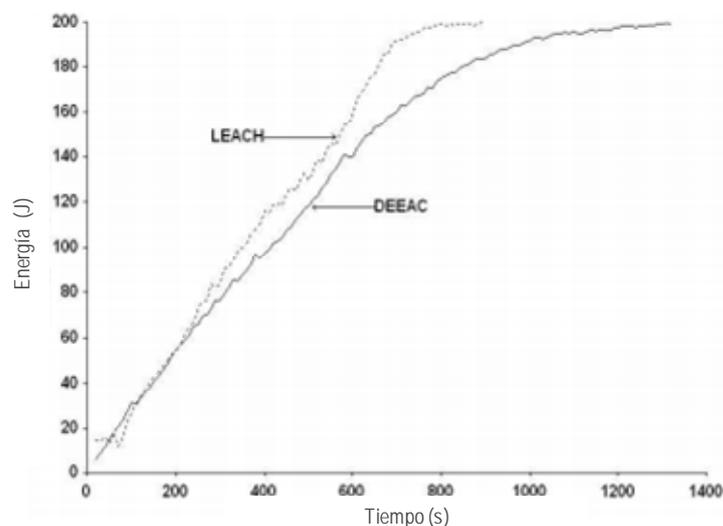


Figura 2.67. Energía total disipada en función del tiempo para LEACH y DEEAC

Se puede observar como DEEAC consigue una menor cantidad de energía disipada, alcanzando en consecuencia una mayor vida de la red.

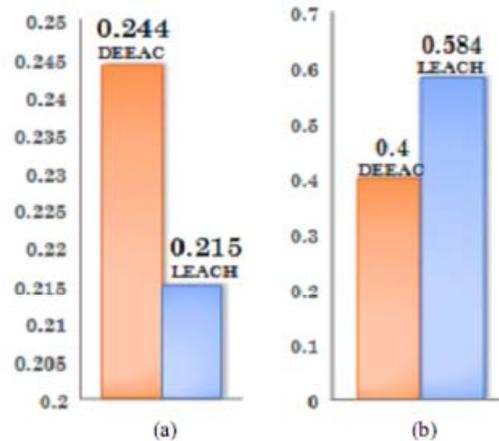


Figura 2.68. (a) Fracción de *CHs* en *puntos calientes* y (b) fracción de la energía disipada por los nodos que pertenecen a *puntos calientes*

Se ve claramente (figura 2.68) que en puntos calientes existe una mayor distribución de *CHs* en *DEEAC* y que en dichas regiones, con alto ratio de retransmisiones, la energía disipada es menor que en *LEACH*, lo que da mucha ventaja al respecto y confirma los buenos resultados en el cálculo de los umbrales de *DEEAC*. La gráfica de la figura 2.69 confirma lo expuesto hasta el momento.

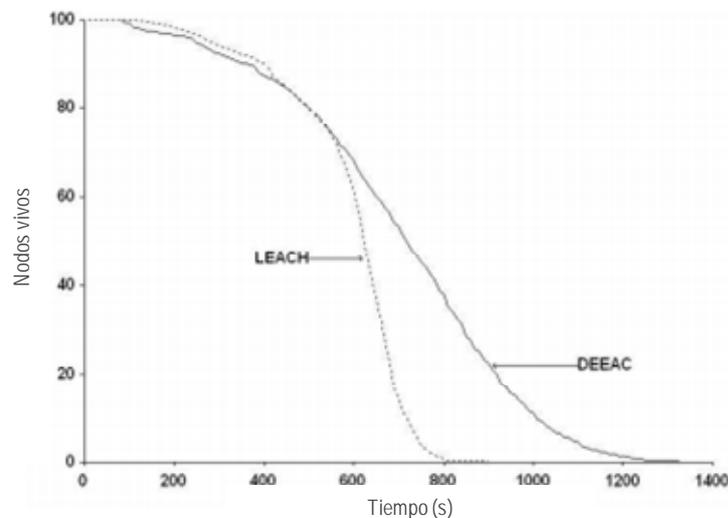


Figura 2.69. Número de nodos vivos en función del tiempo

Viendo las mejoras incorporadas por los protocolos jerárquicos en comparación con los planos, el esquema a seguir parece claro. Para conseguir buenos rendimientos, los protocolos jerárquicos emplean asociación y diferenciación de los roles en los nodos, construyendo normalmente una jerarquía en niveles pudiendo diferenciar dichos roles y consiguiendo así una separación de las capas y el trabajo que se realiza en cada una de ellas, para alcanzar un rendimiento más que aceptable. Diferentes estrategias han empleado el uso de *clústeres* con un representante para disminuir la sobrecarga incorporada por la adquisición de datos y su envío, haciendo de esta forma que un único sensor, con carácter rotativo, sea el encargado de llevar la información hasta el sumidero. *LEACH* fue pionero en este aspecto y queda marcado

como referencia eficiente para los protocolos que surgieron con posterioridad, como *TEEN*, *APTEEN* y *SOP*, los cuales siguen el esquema básico con unas u otras variantes. La idea de distribuir el consumo a lo largo de todos los nodos ya se había tenido en cuenta anteriormente pero estos protocolos instauran diversas estrategias que marcan posibles vías a seguir muy interesantes a nuestro modo de ver.

## 2.4 Redes basadas en localización

En este tipo de encaminamiento, los nodos son direccionados mediante métodos de **localización**. La distancia entre dos nodos vecinos se estima tomando como base la fuerza de la señal entrante. Las coordenadas relativas de los demás nodos se calculan mediante interpolación entre vecinos. Alternativamente, la posición de los nodos está disponible directamente mediante la comunicación con un satélite, usando *GPS (Global Positioning System)*, si se equipan los nodos con un receptor *GPS* de bajo coste. Para ahorrar energía, algunos protocolos basados en localización requieren que los nodos pasen al estado *sleep* si no existe actividad alguna. Mayores ahorros en el consumo se pueden obtener si se aumenta el número de nodos que pasan al estado *sleep* en la red. Sin embargo, no siempre es posible este aumento, se hará en la medida de lo posible.

Los protocolos más representativos de esta rama se presentan en los siguientes apartados del capítulo.

### 2.4.1 Geographic Adaptive Fidelity (GAF)

*GAF* [XU01] es un algoritmo basado en localización y diseñado principalmente para redes *ad-hoc*, pero también aplicable para redes de sensores. La red se divide primeramente en zonas fijas, formando una malla virtual. Dentro de cada zona, los nodos colaboran los unos con los otros jugando diferentes roles. Por ejemplo, los nodos elegirán un nodo para que esté despierto durante un cierto periodo de tiempo y entonces dichos nodos podrán pasar al estado *sleep*. El nodo seleccionado es responsable de monitorizar y enviar los datos al sumidero en nombre de los nodos de la zona. Esto se puede observar en la *figura 2.70*, donde se ve la malla virtual y los nodos seleccionados como representantes:

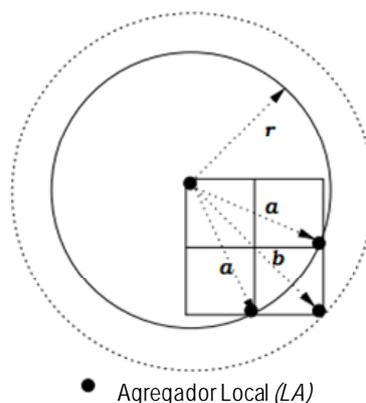


Figura 2.70. Malla virtual y representantes en *GAF*

De esta forma, *GAF* conserva energía apagando nodos no necesarios en la red, sin que esto afecte al nivel de enrutamiento.

Cada nodo usa posicionamiento *GPS* para asociarse con un punto de la malla virtual. Los nodos que se asocien con el mismo punto de la malla se consideran equivalentes en términos de coste del enrutamiento de los paquetes. Dicha equivalencia ayuda a mantener a los nodos localizados en una cierta zona de la malla en el estado de *sleep*, con el fin de ahorrar energía. Periódicamente, los nodos se despertarán para mantener la carga de la red equilibrada. Por ello, *GAF* incrementa el tiempo de vida de la red conforme el número de nodos aumenta.

En *GAF*, como se puede apreciar en el diagrama de la figura 2.71, los nodos pueden estar en tres estados; *sleeping*, *discovery* y *active*:

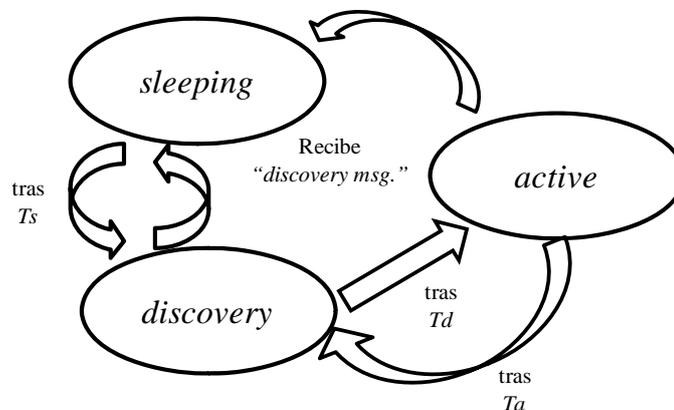


Figura 2.71. Transiciones entre estados en *GAF*

1. **Discovery:** inicialmente los nodos están en este estado. Cuando se está en estado *discovery*, el nodo enciende su radio e intercambia mensajes de descubrimiento para encontrar otros nodos en la misma malla. El mensaje de descubrimiento es una tupla formada por el *id* del nodo, el *id* de la malla, el tiempo activo estimado del nodo y el estado del nodo. El nodo emplea su posición y el tamaño de la malla para determinar el *id* de la misma.

Cuando un nodo entra en el estado *discovery*, configura un temporizador para  $T_d$  segundos. Cuando el temporizador salta, el nodo realiza un *broadcast* de su mensaje de descubrimiento y entra en el estado activo. El temporizador puede ser cancelado por otros mensajes de descubrimiento. Este temporizador reduce la probabilidad de colisión de los mensajes de descubrimiento.

2. **Active:** cuando un nodo entra en estado activo, fija un valor de *timeout*  $T_a$  para definir cuánto tiempo puede estar el nodo en este estado. Al cabo de  $T_a$  segundos, el nodo volverá el estado de *discovery*. Mientras esté en *active*, el nodo periódicamente realizará un *re-broadcast* de su mensaje de descubrimiento a intervalos de  $T_d$ .

Un nodo que esté en el estado de *discovery* o de *active*, puede cambiar su estado a *sleeping*, siempre y cuando pueda determinar algún otro nodo equivalente que pueda manejar el enrutamiento. Los nodos negocian entre ellos qué nodo manejará el enrutamiento.

3. ***Sleeping***: cuando se transita a *sleeping*, el nodo cancela todos los temporizadores pendientes y apaga la radio. El nodo se despertará después de un tiempo  $T_s$  dependiente de la aplicación subyacente y volverá al estado de *discovery*.

Estos diferentes estados describen el comportamiento energético del protocolo y disminuyen el consumo en la medida de lo posible. Cuanto más se maximice el tiempo en estados de *sleeping* sin afectar al enrutamiento, mejor consumo energético. Esta premisa es vital cuando se trabaja con algoritmos con diferentes estados y se pretende maximizar el tiempo en el que se está en algunos de bajo coste. EDETA actúa haciendo más hincapié en estas transiciones y definiendo más detalladamente su *DTE* y estados en función de las operaciones del algoritmo. Sin duda, *GAF* tiene buenas intenciones y sigue una buena línea de actuación para conseguir un consumo eficiente.

Además, la distribución de la carga también es un aspecto importante que se toca en el algoritmo. *GAF* emplea una estrategia de equilibrio de la carga con el fin de que todos los nodos estén despiertos y funcionando el mayor tiempo posible. La idea clave detrás de esto es que todos los nodos de la red son igualmente importantes y ninguno de ellos debe ser penalizado más que los demás. Una alternativa a esta estrategia es dejar que cada nodo agote su batería por completo por turnos, mientras otros nodos duermen. De esta forma no se consigue equilibrio alguno sino que se va terminando con la energía de los sensores de manera ordenada. Esto no es lo adecuado ya que lo óptimo es ir disminuyendo paulatinamente la energía de la red hasta que se agote por completo, no fijando el consumo en unos u otros nodos hasta su inanición.

La estrategia de *GAF* es la siguiente: después de que un nodo permanezca en el estado *active* por un periodo de  $T_a$ , cambia su estado a *discovery* para dar una oportunidad a otros nodos que se encuentren en su misma malla para pasar a *active*. Hay que recordar que los nodos están ordenados de acuerdo a su energía remanente. Cuando el nodo activo cambia su estado a *discovery*, es más probable que éste tenga menos energía restante que sus vecinos, ya que presumiblemente los nodos vecinos estuvieran en estado *sleeping* conservando energía mientras los otros están en estado activo. Consecuentemente, el nodo que estaba en estado activo, es menos probable que permanezca en *active* después de la fase de descubrimiento, con lo que el equilibrio es el adecuado.

Debido a la naturaleza del protocolo, los autores implementan *GAF* en *NS2* y lo comparan con *AODV* y distintas configuraciones de *GAF*. Las distintas configuraciones se basan en movimientos intermitentes a una velocidad fija cada 7 temporizaciones diferentes: 0, 30, 60, 120, 300, 600 y 900 segundos. Para cada tiempo de pausa, se generan 10 conjuntos de posiciones iniciales y *way-points* aleatorios.

Los resultados referentes al tiempo de vida de la red se muestran en la *figura 2.72*, donde se aprecia claramente como *AODV* a los 450 segundos se queda sin energía y las distintas configuraciones de *GAF* consiguen distintos niveles de mejora.

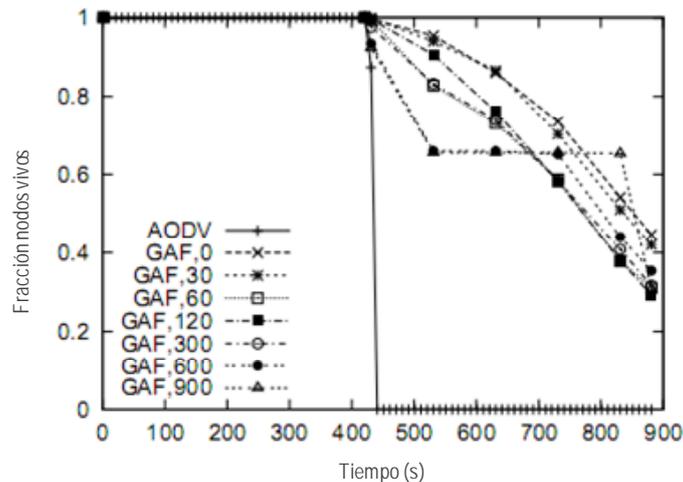


Figura 2.72. Comparación en cuanto al tiempo de vida de la red. *GAF* vs. *AODV* a velocidades bajas de los nodos y varios tiempos de pausa. Movimiento: 1m/s, tráfico: 20pkt/s

Después de 900 segundos, el 30-40% de los nodos siguen vivos dependiendo de la configuración asociada. Los escenarios con tiempo de pausa más pequeños consiguen mayores mejoras en el tiempo de vida que los que poseen tiempos de pausa más largos. Esto se debe a que la movilidad en los nodos mejora el equilibrio de la carga y tiene relación directa en el tiempo de vida de la red. Consideremos ahora el caso de 900 segundos (no hay movimiento en la simulación); en este caso, las celdas de la malla con un único nodo permanecerán constantemente en el estado *active* y por ello se agotarán al cabo de 450 segundos. Las mallas con varios nodos permanecerán más tiempo y entonces morirán todas al mismo tiempo, produciendo el efecto de escalera en 550 y 850 segundos.

Queda muy patente que *AODV* no es rival para *GAF*, pero no existe comparación alguna con protocolos exclusivos de redes de sensores. Recordemos que *GAF* es un protocolo destinado a redes *ad-hoc* por lo que la viabilidad de este algoritmo para *WSN* queda en entredicho. Existen mejores protocolos de otras ramas, por ejemplo la jerárquica, donde los resultados son más esperanzadores que los descritos anteriormente. Aun así, en *GAF* se deja constancia de que transitar a estados de bajo consumo y distribuir la carga es una medida primordial para conseguir un buen consumo en la red. Da igual el tipo de protocolo o la rama que deseemos usar, la idea es clara: los protocolos que deseen alcanzar un consumo energético eficiente, deben proponer mecanismos de distribución de la carga y, en la medida de lo posible, aprovechar al máximo los tiempos en estados de bajo consumo, ya que ambos conceptos están energéticamente relacionados.

### 2.4.2 Geographic and Energy Aware Routing (GEAR)

GEAR [YU01] emplea heurísticas de selección de vecinos basadas en **energía y geolocalización** para encaminar un paquete hacia la región de destino. Dentro de la región, se emplea una técnica de *forwarding* geográfico recursivo para diseminar el paquete. El uso de información geográfica mientras se diseminan las consultas a determinadas regiones debido a que normalmente las consultas (*queries*) incluyen atributos geográficos, radica en la idea de restringir el interés de los datos en Difusión Directa mediante la única consideración de una región, en vez de enviar el interés a toda la red. Haciendo esto, GEAR puede conseguir mayor ahorro energético que Difusión Directa. Aunque el diseño de equilibrado energético de GEAR es una motivación típica de las redes de sensores, el protocolo se aplica generalmente a redes *ad-hoc*.

Cada nodo en GEAR mantiene un *coste estimado* y un *coste aprendido* para alcanzar el destino a través de sus vecinos. El *coste estimado* es una combinación de energía residual y distancia hasta el destino. El *coste aprendido* es un refinamiento del *coste estimado* que tiene en cuenta el enrutamiento alrededor de agujeros en la red. Un agujero ocurre cuando un nodo no tiene ningún vecino cercano a la región de destino más que él mismo. Si no hay agujeros, el *coste estimado* es igual al *coste aprendido*. El *coste aprendido* se propaga un salto hacia atrás cada vez que un paquete llega al destino para que la configuración de la ruta del siguiente paquete se ajuste.

El proceso de *forwarding* de un paquete a todos los nodos de la región destino consiste en dos fases:

1. *Forwarding* del paquete hacia la región destino: GEAR emplea un heurístico de selección de vecinos basado en energía residual y geográfica para encaminar el paquete hacia la región de destino. Hay que considerar dos casos:
  - a. Cuando existe un nodo vecino más cercano al destino, GEAR selecciona el siguiente salto de entre dichos vecinos cercanos.
  - b. Cuando todos los vecinos están alejados, nos encontramos ante un agujero. GEAR selecciona el nodo como siguiente salto que minimice un coste asociado a dicho nodo.

En la figura 2.73 se puede observar cómo GEAR evita la existencia de estos *agujeros*:

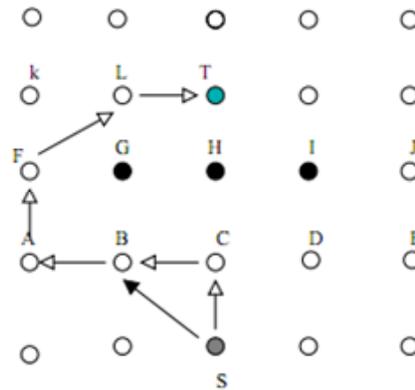


Figura 2.73. Aprendizaje de las rutas alrededor de los agujeros

- Diseminación del paquete en la región: Bajo la mayoría de las situaciones, se emplea el enrutamiento geográfico recursivo para diseminar el paquete en la región. En la figura 2.74 se aprecia dicho enrutamiento geográfico recursivo.

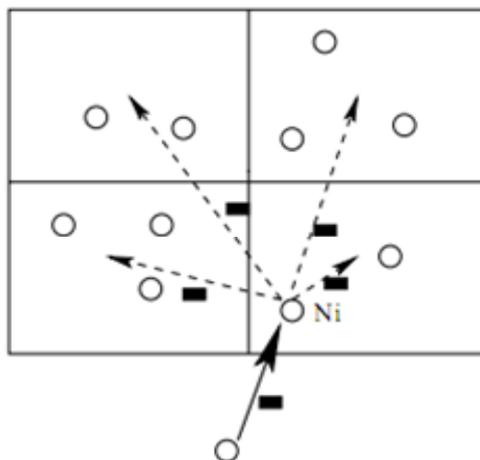


Figura 2.74. Forwarding geográfico recursivo

Sin embargo, bajo ciertas circunstancias, el enrutamiento geográfico recursivo no termina, encaminando inútilmente en bucle alrededor de una región vacía, antes de que la cuenta de saltos del paquete alcance cierto umbral. En este caso, se propone el uso de un *flooding* restrictivo.

GEAR realiza las siguientes asunciones:

- Cada paquete de consulta tiene su propia región destino, especificada de algún modo (de la descripción del algoritmo, se deduce que es una región rectangular supuestamente).
- Cada nodo sabe su propia posición y energía remanente, así como la posición y energía residual de sus vecinos a través de un "sencillo" sistema de mensajes HELLO. Obviamente, si el sensor posee receptor GPS puede saber su posición (con el consumo extra que un HW de este tipo conlleva), pero para conocer la de sus vecinos y además

la energía de éstos, efectivamente se necesita algún tipo de protocolo de distribución de dicha información. Mediante *HELLOs* se puede aproximar dicha posición triangulando de alguna manera, pero para conocer la energía de sus vecinos debe haber intercambio de mensajes. Este intercambio de información llegado a cierto punto, y ante la ausencia de mecanismos de control, crearía una **sobrecarga extra**. Remarcamos que "*sencillo*" no es un adjetivo apropiado.

6. Los enlaces son bidireccionales, luego si un nodo escucha a un vecino  $N_i$ , entonces su radio de transmisión puede alcanzar  $N_j$ . Esta asunción de casi siempre viable.

El empleo del *flooding* anteriormente citado, se puede ver con *buenos ojos* cuando la red no dispone de alta densidad de sensores. En redes con alta densidad de nodos, puede acabar con cualquier medida de eficiencia energética que tuviera el algoritmo, como ya se vio en protocolos anteriores que usaban este método de enrutamiento. Este es uno de los **puntos débiles** del algoritmo, ya que cuando hablamos de *WSNs* estamos refiriéndonos a redes con mucha cantidad de nodos, donde este tipo de medidas no son aceptables. Además, el protocolo se centra en el enrutamiento de paquetes, dejando de lado cualquier mecanismo de agrupación de sensores o de distribución de la carga, quitando la propia disposición de receptores *GPS* en los nodos, *HW* vital para que este protocolo funcione. Todos estos matices hacen que sea apropiado quizás para redes *ad-hoc* pero no para nuestro cometido, centrado en la eficiencia energética en redes de sensores inalámbricos.

## 2.5 Conclusiones

Queda claro que la adecuación de los protocolos basados en localización para obtener protocolos energéticamente eficientes es dudosa, por tanto, nos centraremos en la comparación de los dos tipos de protocolos de enrutamiento basados en las topologías de la red más viables que hemos visto: redes planas y jerárquicas.

Como continuación y a efectos de comparación y resumen de las características básicas de estos dos tipos, presentamos las siguientes tablas comparativas (véase tabla 2.2 y 2.3).

Encaminamiento jerárquico	Encaminamiento plano
Planificación basada en reserva	Planificación basada en contienda
Se evitan colisiones	Sobrecarga por colisiones
Ciclo de trabajo reducido debido a los periódicos modos de bajo consumo	Ciclo de trabajo variable
Agregación de datos realizada por <i>cluster head</i>	Los nodos en las ruta multisalto agregan los datos que reciben de los vecinos
Mecanismos de enrutamiento permiten implementaciones de complejidad razonable	Mecanismos de enrutamiento complejos de implementar
Requiere sincronización	Los enlaces se forman sobre la marcha
Sobrecarga de formación de clústeres	Las rutas se forman solo en regiones que tienen datos para transmitir. Sobrecarga cuando se necesitan
Bajas latencias en los múltiples saltos a través de <i>cluster heads</i>	Latencias mientras se despiertan nodos intermedios y se configuran las rutas
Consumo de energía uniforme	Consumo de energía en función de los patrones de tráfico
Reparto adecuado del canal	No se garantiza reparto óptimo del canal

Tabla 2.2. Encaminamiento jerárquico vs. plano

Destacamos la capacidad de evitar colisiones y la agregación de datos del encaminamiento jerárquico frente a las redes planas además de dos puntos muy importantes como son la uniforme disipación de energía, alargando consecuentemente la vida de la red, y la planificación por reserva del tiempo.

En la *tabla 2.3* podemos comparar más detalladamente las características de los diferentes protocolos que se han estudiado.

	Clasificación	Movilidad	Conocimiento de la posición	Basado en negociación	Agregación de datos	Localización	QoS	Complejidad	Escalabilidad	Multi-ruta	Basado en consultas
SPIN	Plano	Posible	No	Sí	Sí	No	No	Baja	Limitada	Sí	Sí
<i>Directed Diffusion</i>	Plano	Limitada	No	Sí	Sí	Sí	No	Baja	Limitada	Sí	Sí
Rumor Routing	Plano	Muy limitada	No	No	Sí	No	No	Baja	Buena	No	Sí
GBR	Plano	Limitada	No	No	Sí	No	No	Baja	Limitada	No	Sí
MCFA	Plano	No	No	No	No	No	No	Baja	Buena	No	No
CADR	Plano	No	No	No	Sí	No	No	Baja	Limitada	No	No
COUGAR	Plano	No	No	No	Sí	No	No	Baja	Limitada	No	Sí
ACQUIRE	Plano	Limitada	No	No	Sí	No	No	Baja	Limitada	No	Sí
EAR	Plano	Limitada	No	No	No	No	No	Baja	Limitada	No	Sí
LEACH	Jerárquico	Sumidero fijo	No	No	Sí	Sí	No	CHs	Buena	No	No
TEEN & APTEEN	Jerárquico	Sumidero fijo	No	No	Sí	Sí	No	CHs	Buena	No	No
PEGASIS	Jerárquico	Sumidero fijo	No	No	No	Sí	No	Baja	Buena	No	No
MECN & SMECN	Jerárquico	No	No	No	No	No	No	Baja	Baja	No	No
SOP	Jerárquico	No	No	No	No	No	No	Baja	Baja	No	No
HPAR	Jerárquico	No	No	No	No	No	No	Baja	Buena	No	No
VGA	Jerárquico	No	No	Sí	Sí	Sí	No	CHs	Buena	Sí	No
<i>Sensor aggregate</i>	Jerárquico	Limitada	No	No	Sí	No	No	Baja	Buena	No	Posible
TTDD	Jerárquico	Sí	Sí	No	No	No	No	Moderada	Baja	Posible	Posible

Tabla 2.3. Clasificación y comparación de protocolos de enrutamiento en WSN

Se observa en la tabla 2.3 como en la columna correspondiente a Consumo de Energía (*Power Usage*), las mejores opciones en cuanto a clasificación del protocolo a usar corresponden a la de un protocolo jerárquico, ya que en estos casos el ahorro energético es máximo. En *LEACH* y *TEEN-APTEEN* tenemos la posibilidad de agregación de la información y buena escalabilidad.

La diferencia entre la elección de protocolos planos frente a protocolos jerárquicos es clara en términos de ahorro energético. En el siguiente punto detallaremos los enfoques que se están siguiendo actualmente para obtener un protocolo que mejore el consumo energético y tenga una topología jerárquica, debido a que bajo nuestro punto de vista esta opción es la más adecuada para obtener un buen rendimiento.

## Capítulo 3

# Estrategias de optimización del consumo en WSN

### 3.1 Enfoque para optimización del consumo de energía

En gran parte de los trabajos revisados en el estudio bibliográfico previo plasmado en el capítulo 2, se ha dejado entrever la necesidad de trabajar en la dirección del enrutamiento jerárquico para obtener buenas prestaciones energéticas. La posibilidad de este tipo de protocolos de evitar colisiones distribuyendo en varias capas el trabajo, diferenciando roles entre los nodos y realizando agregación de la información hace que este enfoque cobre una gran importancia si se quiere desarrollar protocolos eficientes en estos aspectos.

Con el fin de diseñar buenos protocolos para redes de sensores, es importante determinar los parámetros más relevantes así como las métricas elaboradas para comparar diferentes enfoques. Puesto que hay diversas formas de evaluar estos parámetros, nos centraremos en los siguientes:

- Las redes de sensores pueden contener fácilmente cientos de nodos y éstos deben ser distribuidos en zonas remotas y de gran amplitud, permitiendo a los usuarios extraer información de maneras que de otra forma no serían posibles. Por ello, se requiere que los nodos sean capaces de comunicarse los unos con los otros incluso en la ausencia de una infraestructura de red.

- Dicha red de sensores también debe ser capaz de procesar la información captada por los nodos y enviarla **adecuadamente** al sumidero. y, como se ha visto anteriormente, cuando se habla de *WSNs* este objetivo es primordial. Los sensores deben ser capaces de maximizar dicha entrega de información al sumidero, minimizando de igual manera el consumo energético empleado para tal fin.
- Dentro de esta adecuación en el envío de los datos, estos datos son normalmente sensibles al **tiempo**, luego es importante acotar la recepción de los mismos en el tiempo, ajustando la **latencia** de la red lo máximo posible.
- Los protocolos propuestos deben tener mecanismos para el **ahorro de energía** en los nodos debido a que recargar la batería de los mismos es un gran inconveniente o imposible. Todos los aspectos de los nodos, desde el HW hasta las operaciones realizadas por éstos en el protocolo, en la medida de lo posible ayudar a **distribuir la carga energética**.
- El concepto de calidad en una red de sensores es diferente al concepto en redes *wireless*. Para las *WSNs*, el usuario final no requiere toda la información de la red debido a que la información de nodos vecinos está sensiblemente correlacionada, haciendo que exista redundancia. La calidad de la red se basa entonces en dos aspectos:
  1. Aunque se busca productividad y bajo consumo, los protocolos deben disponer de esquemas mínimos de **QoS**, que garanticen un retardo acorde con las especificaciones de la aplicación y una tasa de entrega aceptable, no poniendo en peligro otros aspectos. El desarrollar protocolos eficientemente energéticos no debe ser a costa de obtener deficiencias en los atributos principales de la red.
  2. La calidad de la red se basa principalmente en el conjunto de datos agregados, por lo que los protocolos deben designarse también pensando en optimizar este conjunto de datos y la carga que proporcionan.

En los protocolos de enrutamiento planos se asume que la red es generalmente homogénea, esto es, todos los nodos de la red comparten el mismo canal de acceso aleatorio *wireless*, el *status* y las capacidades de cada nodo son iguales. *SPIN* [KULIK99][KULIK02], *Direct Diffusion* [INTANAG03] y *Rumor Routing* [BRAGINS02] entre otros comparten este punto de vista. Se hace necesario disminuir la sobrecarga de estas redes y proporcionarles un buen escalado y, por ello, las investigaciones se centran en una opción jerárquica de desarrollo.

El enrutamiento en redes jerárquicas escala perfectamente con el tamaño de la red, conformando así un elegante paradigma de aproximación para la reducción del consumo

energético ya que se elimina el *flooding* de envío en la red, reduciéndose así el número de transmisiones gracias a su topología, como se puede ver en *LEACH* [HEINZEL00, DEOSAR09], *TEEN-APTEEN* [MANJESH01][MANJESH02] y *TTDD* [YE02].

### 3.2 Selección de las estrategias más adecuadas

Sabiendo que nuestra actual prioridad es la del ahorro energético, conservando valores de retardos, *QoS* y aportando tolerancia a fallos entre otras características, centramos nuestro trabajo de elaboración de un algoritmo energéticamente eficiente en los siguientes términos:

7. Conseguir un buen rendimiento energético es **primordial**. La tecnología seguirá evolucionando y se dispondrá de sensores más modernos y por lo tanto con mejores mecanismos de disipación. No menos importante es el rendimiento global, donde la vida de la totalidad de la red se vea aumentada considerablemente por medio de la distribución inteligente del gasto individual de cada nodo, en referencia a sus características operativas. Tanto en un primer momento pasado, como en los enfoques actuales, se presta mucha atención a la **forma en la que dicha carga energética se distribuye**.
8. Gracias a las ventajas del enrutamiento jerárquico, disponer de **diferentes roles** nos ayuda a **diferenciar posibles fases** en las operaciones del protocolo. En un principio, la existencia de dos modos de funcionamiento en los sensores puede entrever un mayor gasto de energía, pero debido a la existencia de un modo agresivo, como puede ser el modo *CH*, y la existencia de otro clásico como el modo de transmisión normal, hace que se pueda emplear uno de esos dos modos en una capa y otro modo en restantes capas. En la capa inferior estaría el modo normal de envío, donde los nodos retransmiten su información hasta un *CH*, lo que se denomina *intra-cluster routing*.

Restantes capas estarían formadas por todos los nodos con rol de *CH*, donde entre ellos se crearía una **red multipath** (o *singlepath*) hasta el sumidero, denominando a esta característica *inter-cluster routing*. Esta diferenciación acabaría por llevar a la extinción a las capas superiores, pero las **rotaciones sucesivas de este rol** de alta carga energética consiguen el balanceo de la red.

9. La existencia de **diferentes fases** en el protocolo permite, al igual que los diferentes roles en los nodos, disponer de mecanismos para distribuir mejor la carga energética a la que se ven sometidas los sensores. Un buen acercamiento a esta idea sería similar a *SOP* o *EECR*, donde se dispone de una fase de configuración para la formación de los *clústeres* y todo lo que conlleva, y una fase de comunicación donde se realizan las tareas típicas de transmisión al sumidero. El disponer de diferentes fases y modos de funcionamiento ayuda **diferenciar las distintas operaciones de los nodos** a lo largo del tiempo, donde podemos distribuir al detalle las actuaciones de los nodos y llevar el ahorro al máximo.

10. La división de los tiempos de transmisión también puede ser un buen mecanismo. El uso de **TDMA-scheduling** permite hacer saber a cualquier tipo de nodo los tiempos a los que se tiene que ceñir, disponiendo así de un uso detallado de su tiempo de vida.
11. Realizar tareas de **agregación-fusión y compresión** de la información antes de proceder al envío de los datos tiene una importancia vital ya que, sabiendo que las tareas de computación son más baratas que las de transmisión, se disminuye la cantidad de datos y por lo tanto el tiempo que se tarda en transmitirlos, lo que conlleva un menor tiempo en el que el módulo de radio está activo y por ello un **menor consumo**.

### 3.3 Principios para los mecanismos propuestos y su comparación con otras aproximaciones

En estos términos, se detallan los principios y características de los mecanismos que se propondrán en la presente tesis, novedosos y mejorados respecto de los enfoques anteriores y que han demostrado tener capacidad para ser una buena base en consumo energético regulado y eficiente. Se introduce el concepto de **Base estratégica**, para numerar los protocolos que anteriormente pudieron acercarse al enfoque que se describe, pero que por unos u otros motivos no lo incorporaron, se dejó como futura extensión o no se supo como encuadrar dicha mejora en el mecanismo del protocolo:

1. Eficiente consumo energético basado en la **distribución de la carga de forma global** entre el total del número de nodos de la red. *Base estratégica: LEACH, LEACH-C y e3D.*
  - a. Se ha demostrado [HEINZEL00] que el distribuir el consumo por medio de la **rotación de roles** pesados es una medida eficiente de ahorro energético.
2. Agrupación en **clústeres**: queda patente que el enfoque del **clúster** jerárquico obtiene mayores beneficios que una jerarquización sin clústeres. En **TTDD** por ejemplo, se emplea jerarquización pero sin la formación de clústeres, lo que obliga al empleo de cadenas y mecanismos de mantenimiento de dichas cadenas, provocando una sobrecarga y coste superiores a enfoques **clusterizados**. Por ello, agrupando un número determinado de nodos con rol normal en un **clúster**, gobernado por un único representante ó **CH**, proporciona una forma sencilla e inteligente de gobernar un número específico de sensores y **controlar sus labores, minimizando su consumo**. *Base estratégica: LEACH, SOP e IMRAFRA.*
3. Además de agrupación por **clúster** de los nodos normales, se realiza la construcción de un **árbol de CHs** en la capa superior, con el fin de ayudar al envío de la información hasta el sumidero mediante **inter-cluster routing**. *Base estratégica: hierarchical-PEGASIS e IMRAFRA.*

- a. Cabe destacar que ni *PEGASIS* ni *IMRAFRA* incorporan ninguna estrategia de datos en capas superiores para ayudar a la gestión del enrutamiento *inter-cluster*, pero intentan incorporar algún tipo de mecanismo a esta labor para facilitar la transmisión de la información de los *CHs* hasta el sumidero.
  - b. *Hierarchical-PEGASIS* construye cadenas enlazadas, con el problema que conlleva y ya se detalló en su apartado, quedando lejos de cualquier estructura de tipo árbol como la que planteamos en la presente tesis (arquitectura EDETA).
  - c. *IMRAFRA* intuye que proporcionar al *inter-cluster routing* de mecanismos óptimos mejora el rendimiento y por ello elabora un algoritmo voraz, que seleccionando pasos de coste óptimo, lleve la información hasta el sumidero con el menor consumo. Como también se detalla en su apartado correspondiente, esto no siempre lleva a soluciones óptimas por lo que no es buena solución, aunque si deja constancia de que algún tipo de estructura de datos puede ser apropiada.
4. Dos posibles roles: rol de *CH* ó rol de nodo normal. Cada tipo de rol, auto-asignado por una probabilidad, tiene asociado unas características propias tanto de modo de operación, especificado por su DTE, como por mecanismos propios de ahorro y mejora del rendimiento energético. Cada un número específico de rondas  $n$ , se efectuará una **rotación de estos roles**, para conservar el balanceo del consumo. *Base estratégica: LEACH, LEACH-C, SOP e IMRAFRA.*
5. Organización de la jerarquía en **dos capas** o *Tiers*: una primera capa, formada por los *clústeres* y donde al llegar la información al *CH* y tras la posterior agregación y compresión, se suben los datos por el árbol de *CHs* mediante *inter-cluster routing* hasta el sumidero. *Base estratégica: LEACH, LEACH-C e IMRAFRA.*
- a. Dentro del *clúster* **no** se realiza *intra-cluster routing*, puesto que el *clúster* se organiza en *single-hop*. Una vez que el sensor obtiene sus datos, los envía directamente a su *CH*, que pertenece a un *Tier* superior y está organizado en forma de **árbol**.
  - b. Tras la recepción de la información de todos sus hijos, el *CH* **agrega y comprime** los datos y los envía a su *CH* padre, realizando *inter-cluster routing*.
  - c. Tras sucesivas etapas de *inter-cluster routing*, la información llega al sumidero.
  - d. **Rotación de roles** después de un número  $n$  determinado de rondas, para distribuir la carga energética entre el mayor número posible de nodos, **y vuelta a empezar** desde la fase de formación de los *clústeres*.

6. Existencia de **dos fases de operación**: la *fase de inicialización*, donde se inicializan los *clústeres* junto con la elección de los *CHs* y la construcción del árbol en el nivel 2, y la *fase de comunicación*, en la cual se realizan las rondas *intra-cluster* e *inter-cluster* para el envío de los datos al *CH* y hasta el sumidero, respectivamente. *Base estratégica: LEACH, APTEEN y EECR.*
7. Uso de *TDMA-polling* para transmisión *intra-cluster*: de este modo, el *CH* **sondea** a sus hijos para la recepción de información, distribuyendo el canal en *TDMA* y evitando al máximo las colisiones, además de realizar un notable ahorro energético. *Base estratégica: LEACH y APTEEN.*
  - a. Si bien *LEACH* y *APTEEN* incorporan *TDMA*, el *TDMA-polling* **no** se ha empleado en ningún protocolo de los revisados, por lo que es una característica por la que **EDETA** apuesta y puede dar lugar a beneficios sustanciosos en el tiempo de vida de la red.
  - b. Las bondades que principalmente aporta el *TDMA-polling* a **EDETA** consisten en por un lado **flexibilizar** la transmisión sensor-*CH* mediante el sondeo por parte del *CH* cuando se requiera información en su planificación previamente enviada y en aportar **tolerancia** a posibles **fallos** en los nodos, ya que sondeando previamente a éstos, se consigue responder ante estas caídas de forma prematura.
8. Realización opcional de tareas de **agregación y compresión** antes del envío de la información por parte de los *CHs* por el árbol hasta el sumidero. De esta manera, se reduce la cantidad neta de datos a transmitir con el consecuente **ahorro** por su parte. *Base estratégica: LEACH, LEACH-C, COUGAR, PEGASIS, TEEN y APTEEN.*
9. Durante los períodos de espera por parte de los nodos, transición a **estados de bajo consumo** mediante el cambio de modo del módulo de radio a *sleep*, disminuyendo la disipación de energía y maximizando el ahorro. Estos cambios sucederán normalmente cuando el *CH* deba esperar a que finalice la ronda *intra-cluster* y que todos sus hijos hayan finalizado las transmisiones ó que el propio *CH* deba esperar a retransmitir dicha información a su *CH* padre en el árbol, debido a que tiene que atenerse a su planificación. *Base estratégica: ninguna.*
  - a. **EDETA** hace un análisis detallado de las posibles mejoras en el consumo cambiando el modo en el que la **radio** de los sensores trabaja. La principal baza consiste en **no malgastar energía esperando en modos de funcionamiento costosos**, energéticamente hablando, y pasar a modo de bajo consumo siempre que se pueda en el algoritmo. Debido a que otras estrategias y protocolos esperan en modo recepción, el **ahorro** energético en **EDETA** es **sustancial** en comparación.

Como se dijo anteriormente, algunos puntos característicos de **EDETA** mejoran anteriores alternativas elaboradas por otros protocolos. Ciertos matices añaden funcionalidad y mecanismos que hacen que se incremente el ahorro en el aspecto energético. Estos puntos son los siguientes:

1. A la hora de realizar el *inter-cluster routing*, **EDETA** organiza los *CHs* del *Tier 2* elaborando un árbol dinámico de *CHs*, extendiendo la idea que tímidamente introdujo *PEGASIS* ya que en este protocolo la especie de árbol formado por los *CHs* era de un *single-hop* hacia el sumidero, cuando esto no siempre es posible ni aconsejable [BJORNEM07].

**EDETA** introduce de forma totalmente nítida el concepto de árbol dinámico de *CHs*, organizando sucesivas jerarquías superiores con esta topología. El beneficio de la comunicación directa entre *tiers* dentro del árbol es palpable y, sabiendo de la rotación de éstos y en la misma manera del propio árbol, se asegura un consumo eficiente debido a la poca sobrecarga.

2. La transición a estados de bajo consumo cuando no se requiere actividad alguna por parte de los nodos, es algo vital para un buen rendimiento energético. Se ha demostrado en el propio modelo de energía de **EDETA** que mantenerse lo máximo posible en estos estados ayuda a aumentar el ahorro energético de forma considerable y de la misma forma prolonga la vida total de la red de sensores. Este concepto de ahorro enfocado a los tiempos de actividad, tímidamente nombrado y pasado por alto en *EECR*, hace que **EDETA** maximice el tiempo individual en el que el nodo se encuentra en su consumo mínimo. Esta medida se adopta localmente al nodo para conseguir un mayor ahorro ya que otras medidas, como la rotación aleatoria de roles pesados, tienen su punto de mira en el consumo equilibrado.
3. El empleo de *TDMA-polling* en vez de *TDMA* convencional consigue, además de evitar las colisiones propias de este medio en las transmisiones, que sea la capa superior la que inicie el período de transmisión mediante la trama de sondeo, haciendo que se concatene la planificación del propio nodo con el *polling* del *CH* y que se disponga tanto de **ahorro energético** gracias a la propia planificación, como de **fraccionamiento de los tiempos** gracias a *TDMA*. De esta forma la probabilidad de colisiones se reduce drásticamente.
4. Las tareas de agregación se realizan opcionalmente en todas las capas, desde el primer *CH* pasando por todos los pertenecientes al árbol global. Esto aumenta el beneficio otorgado por estos mecanismos de compresión en caso de que la aplicación o el usuario final lo requiera.

Se puede ver claramente como **EDETA** pretende, al igual que sus predecesores, maximizar el tiempo de vida de la red ya sea por medios convencionales y sencillos, como la rotación de roles, o con mecanismos más complejos, como el uso de tareas de planificación y *TDMA-*

*polling*, que además de realizar las acciones de repartición de tiempos ayudan a disminuir el consumo. También incorpora la propia topología jerárquica natural, añadiendo un componente de árbol dinámico y conformando un nuevo tipo de topología en varias capas que ofrece un buen rendimiento energético, tanto por el encaminamiento específico adoptado como por la posible agregación y compresión de los datos en más capas que sus antecesores. Además aporta nuevas características como *QoS* y tolerancia a fallos sin afectar significativamente a las prestaciones ni al consumo energético. De esta forma, adoptando medidas que protocolos anteriores incorporaron, en mejor o peor medida, y desarrollando mecanismos nuevos, **EDETA** se erige como un buen enfoque, con mejoras en el consumo individual de los nodos y tiempo de vida de la red sustanciales en comparación con cualquier estrategia anterior.

## Capítulo 4

# Protocolo de encaminamiento EDETA

### 4.1 Introducción

La arquitectura de este nuevo protocolo se basa en la combinación de técnicas para obtener la máxima eficiencia energética, escalabilidad, fiabilidad, seguridad y tiempos acotados (QoS) en redes de sensores inalámbricas. Es aplicable tanto a redes homogéneas como heterogéneas cada vez con mayor auge [BOUKER09], por su demostrada efectividad y eficiencia, además de presentar otras características innovadoras en redes de sensores como la comunicación desde sumidero hacia los nodos, comunicación peer-to-peer, reconfiguración en caliente, etc. Todo ello se consigue mediante operaciones y estructuras de datos perfectamente manejables por microcontroladores de baja potencia y bajo coste, lo cual es otra de las características básicas de las redes de sensores inalámbricas. El protocolo se plantea como un protocolo jerárquico, que estructura la red en árboles de clústeres, donde la raíz de cada árbol será un nodo sumidero, dado que a partir de cierto número de nodos en la red, es muy recomendable disponer de varios sumideros en funcionamiento no solo por tolerancia a fallos sino también por prestaciones. Los clústeres son grupos de nodos sensores, controlados por uno de ellos ejerciendo el rol de nodo líder (NL) de forma rotatoria (mediante mecanismos de auto-elección y auto-regulación de población adecuados). Estos nodos líderes gestionan el proceso de formación del clúster y gobiernan comunicación dentro del clúster mediante el protocolo intra-clúster propuesto al efecto, que está basado en la asignación de tiempos a cada nodo (*TDMA-polling*). Asimismo, realizan funciones de agregación y transmisión de datos hacia el sumidero de su árbol. Para ello emplean a nivel inter-clúster un algoritmo de enrutamiento

multi-salto (*multi-hop*) en árbol optimizado y basado también en *TDMA-polling*. Todos los procedimientos y mecanismos se realizan automáticamente en el área que se despliegan los nodos, sin intervención de fuentes externas. Para no sobrecargar a un nodo, el rol de NL se distribuye de forma rotativa entre todos los nodos.

La técnica de *TDMA-polling* implementada permite la acotación de tiempos tanto intra-clúster como inter-clúster permitiendo ofrecer características de tiempo real así como detección temprana de fallos en nodos (uno nodo que no transmite cuando debe p.e), flexibilidad y posibilidad de recuperación e implementación de mecanismos de tolerancia a fallos transparentes o con muy bajo impacto en las prestaciones de la red.

Por todo lo expuesto, cabe resaltar que el abanico de posibles aplicaciones para el presente protocolo es muy extenso, dado que soporta tareas de monitorización con frecuencias de muestreo diferentes, redes homogéneas y heterogéneas, comunicación bidireccional, aporta fiabilidad y acotación de tiempos para aplicaciones que lo requieran (alarmas).

En el capítulo 3 se presentó una solución para reducir el consumo en redes inalámbricas de sensores. Utilizando los resultados de dicho capítulo se propone una nueva arquitectura jerárquica basada en clústeres y árboles para WSN que permite a los nodos permanecer en modo de bajo consumo la mayor parte del tiempo posible, reduciendo notablemente el número de transiciones on-off-on y permitiendo aplicar técnicas muy eficaces de reducción de energía, ofreciendo además las características citadas de tiempo real (ya que los tiempos de comunicación están acotados tanto intra como inter clúster) y tolerancia a fallos (dado que la arquitectura jerárquica propuesta presenta mecanismos de tolerancia a fallos a todos los niveles).

Los experimentos realizados, basados en medidas reales, muestran que EDETA reduce en un factor de 8 la energía consumida comparado con los protocolos *power-aware* más populares, multiplicando, por tanto, la duración de la vida de la red, y ofreciendo además novedosas características en redes inalámbricas de sensores, a la par que necesarias para determinados nuevos campos de aplicación, como son mecanismos de tolerancia a fallos y tiempos acotados.

En este capítulo se detallan la arquitectura propuesta, los tipos de nodos que contempla y los protocolos inter e intra clúster.

## 4.2 Arquitectura propuesta

La arquitectura jerárquica propuesta organiza la WSN en clústeres y árboles. La arquitectura propuesta utiliza esta característica para implementar mecanismos de encaminamiento que incorporan estrategias de alta eficiencia energética. Se implementa una arquitectura de clústeres sencilla que permite operaciones de reconfiguración automáticas ante cambios de la estructura de la red sin una sobrecarga excesiva.

La figura 4.1 muestra el modelo de la arquitectura propuesta. El número de sumideros dependerá principalmente de la cantidad de nodos sensores de la WSN y su ubicación se

discute en el capítulo 5. La arquitectura propuesta asume que al menos se dispone de un nodo sumidero conectado a internet y alimentado.

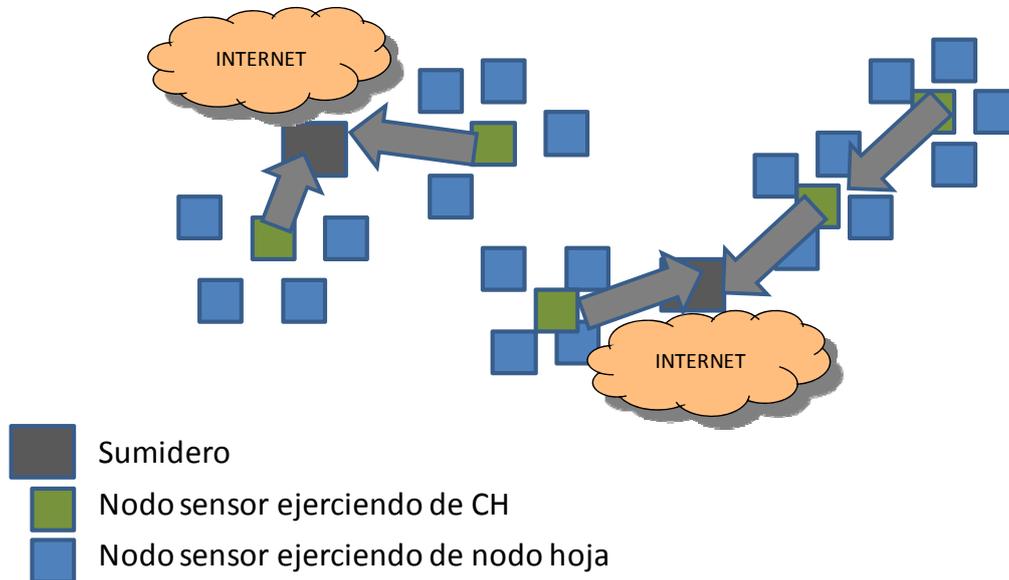


Figura 4.1. Arquitectura propuesta.

La arquitectura EDETA (*Energy-efficient aDaptive hiErarchical and robuST Architecture*) integra dos protocolos diferentes: un protocolo interno del clúster, gobernado por un nodo seleccionado al efecto, denominado protocolo *Intra-Cluster-Communication* y un protocolo entre clústeres denominado *Inter-Cluster-Routing*. Ambos protocolos se implementan sobre el nivel MAC.

A continuación se describen con mayor detalle los diferentes tipos de nodo.

#### 4.2.1 Tipos de nodo

##### 4.2.1.1 Nodo sumidero

Este nodo se encarga de recibir los datos de todos los nodos sensores de la red, realizar un proceso de los mismos y reenviarlos hacia el sistema de gestión de la información. Asimismo, puede realizar algunas funciones de gestión de la red inalámbrica de sensores, como el diagnóstico acerca del estado de los nodos sensores y aplicar los mecanismos de recuperación en caso de fallos.



Figura 4.2. Estructura nodo sumidero

Considerándolo desde el punto de vista de los datos, es una pasarela inteligente entre comunicaciones locales (red de sensores) y remotas (acceso al sistema de gestión de información), puesto que puede realizar diferentes tratamientos (filtrado, resumen, compresión y agregación) sobre los datos recibidos antes de reenviarlos.

Como se puede apreciar en la figura 4.2, se compone de un microcontrolador, un modulo de radio para su conexión con la red de sensores, un modulo de comunicaciones a gran distancia (puede ser una tecnología como Wifi o WiMax, una red LAN cableada o bien utilizar una red WAN como puede ser GSM/GPRS) para la transmisión de los datos hacia el sistema gestor de la información, y un sistema de alimentación perpetua (red eléctrica / paneles solares).

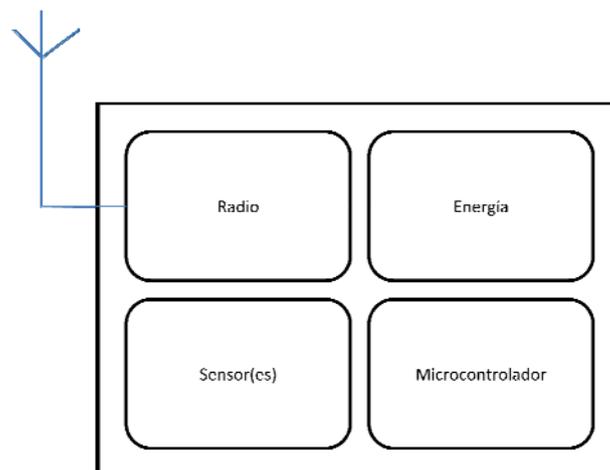


Figura 4.3. Estructura nodo sensor

#### 4.2.1.2 Nodo sensor

Cada nodo sensor, como muestra la figura 4.3, está basado en un microcontrolador que, por una parte, controla los diferentes transductores de que disponga, y por otra dispone de un módulo de comunicaciones y el de alimentación autónoma. En principio pueden ser todos

iguales (WSN homogéneas). Pueden desarrollar el papel de NL o de nodo final, en función de las necesidades de la red. Opcionalmente se pueden incluir nodos más sencillos, con poca capacidad energética, y por lo tanto más económicos como nodos “hoja” que no tengan la posibilidad de ejercer como nodo líder. El protocolo soporta este tercer tipo de nodos, aunque cabe decir que el protocolo optimiza el uso de energía para cualquier tipo de nodos, sea cual sea su fuente de energía y la capacidad de la misma, y es capaz de ejecutarse en nodos incluso de baja potencia de cómputo. Sin embargo, con esta ampliación se adecua también a aplicaciones en las que sería posible tener un subconjunto de nodos con alimentación perpetua (paneles solares + batería + supercondensadores) los cuales ejecutarían el papel de NL, y otro subconjunto de nodos hoja mucho más simples y económicos (con solo una pequeña batería) que solo sensoricen y envíen en su slot, pudiendo encontrarse en modo de bajo consumo la mayoría del tiempo, siendo por tanto la arquitectura propuesta perfectamente aplicable a WSN heterogéneas. Además, posibilitaría en tales escenarios la repoblación de nodos hoja posibilitando la elongación de la vida útil de la red de forma extraordinaria.

#### 4.2.2 Mensajes EDETA

La tabla 4.1 especifica los diferentes tipos de mensaje definidos en el protocolo EDETA, que en los siguientes apartados serán explicados.

Código	Mensaje	Significado
0x01	LEADER	Notificación rol NL
0x02	JOIN-REQ	Petición unión al clúster
0x03	JOIN-REPLY	Respuesta a la petición de unión al clúster (Accepted / Rejected)
0x04	NEED_LEADER	Adaptación de la población de NLs
0x05	ASSOC-REQ	Petición asociación nodo padre árbol
0x06	ASSOC-REPLY	Respuesta a la asociación a nodo padre (Accepted / Rejected)
0x07	DINFO	Información descendencia a nodo padre
0x08	SICI	Asignación planificación desde nodo padre
0x09	SN	Notificación nodo sustituto
0x0A	SINK-FAILURE	Notificación fallo de nodo sumidero
0x0B	POLL	Polling y Tx/Rx de datos/config. opc. down
0x0C	DATA	Datos
0x0D	ACK	Reconocimiento

Tabla 4.1. Mensajes Protocolo EDETA.

##### 4.2.2.1 Notificación rol NL (LEADER) 0x01

Son los mensajes que difunde un nodo líder durante la fase de inicialización y también al final de cada ronda, para notificar su rol y dando a conocer la posibilidad de unirse a él. Al transmitir el mensaje de notificación de rol se utilizará CSMA (escuchando antes de hacer la transmisión) para evitar colisiones con NL adyacentes o por si acaso la red ya estaba

establecida y en funcionamiento, debido a que dicho nodo se hubiera encendido en este instante posterior o se hubiera recuperado de algún tipo de fallo. Los NL empezarán a notificar su rol mediante el envío de este mensaje una vez hayan recibido algún mensaje LEADER de otro NL ó directamente de un nodo sumidero (si procede del sumidero, el campo nivel del mensaje será 0) y puedan por tanto indicar cuál será su nivel en el árbol (en el campo reservado para tal fin en este mensaje de código 0x01), que será el menor nivel observado en los mensajes LEADER escuchados + 1.

Con este mensaje también se ofrece información a nodos líderes que busquen asociación en el árbol para que elijan nodo padre con criterio.

Consta de los siguientes campos:

- Código: 0x01
- Nivel: Número de saltos que tiene su ruta en el árbol para llegar al Sumidero. El sumidero enviará en este campo el valor "0". Cada NL incrementará en un salto el nivel recibido de su padre.
- Nº de hijos: Número de nodos que tiene por debajo en el árbol.
- Energía: energía actual del nodo, para que los nodos que lo escuchen puedan actualizar el umbral de energía utilizado para la elección de rol, y los NL que busquen nodo padre en el árbol lo tengan en cuenta a la hora de decidir entre varias alternativas.

#### **4.2.2.2 Petición de unión al clúster (JOIN-REQ) 0x02**

Este tipo de mensaje es utilizado por los nodos hoja para solicitar formar parte de un clúster. Se envía al NL seleccionado utilizando CSMA. El criterio que usarán los nodos hoja para elegir NL será la calidad de señal recibida, con lo que solicitarán unirse con el que escuchen con mayor calidad de señal. Ello permite ajustar al mínimo la potencia de emisión necesaria para toda comunicación y por tanto permitirá un mayor ahorro energético. Como criterio secundario, ante varias ofertas con misma calidad de señal se seleccionará el NL más cercano al sumidero (que anuncie menor nivel) para además reducir retardos.

El mensaje está formado por los siguientes campos:

- Código: 0x02
- Energía restante: Indicación de batería restante del nodo que lo transmite, para que el NL cuando elija "sustitutos" no elija nodos por debajo de un umbral y así prolongar la vida de los de nodos líderes.
- Flag UO (única opción): Indica al NL que el nodo que solicita unirse al clúster no tiene en su área de cobertura más nodos líderes.

#### 4.2.2.3 Respuesta a la petición de unión al clúster (JOIN-REPLY) 0x03

Es el mensaje que envía el Nodo Líder mediante CSMA para notificar la aceptación o rechazo a la petición de un nodo normal para unirse al clúster que lidera. En caso de aceptación, mediante este mensaje también se indica al nodo qué canal que debe utilizar a partir de ese momento para toda comunicación y durante todo el ciclo. Para evitar posibles interferencias entre clústeres adyacentes cada NL elige una frecuencia para su clúster. Además, en este mensaje también se le asigna al nodo hoja su ranura de tiempo para transmisión intra-clúster, así como su identificador de dirección de nivel 3 en la red. El NL debe recibir el correspondiente ACK por parte del nodo frente a la correcta recepción del mensaje.

Consta de los siguientes campos:

- Código: 0x03
- Admisión: Un valor "1" indica admitido, un valor "0" rechazado.
- Canal: En caso de ser admitido, en este campo se indicará el canal a utilizar para las comunicaciones dentro de éste clúster.
- Slot TDMA intra-clúster asignado al nodo.
- Asignación de dirección para el nodo. Los 'p' primeros bits coincidirán con el prefijo del nodo líder, y el resto de bits serán asignados por este de forma consecutiva.

#### 4.2.2.4 Necesidad de Nodo Líder (NEED\_LEADER) 0x04

Mediante este mensaje se adapta dinámicamente la distribución de la población de nodos líderes para asegurar la cobertura total de la red. Si durante la formación del árbol en la fase de inicialización algún Nodo Líder no tiene a su alcance ningún otro Nodo Líder para encaminar hacia un sumidero entonces difunde este mensaje, de forma que cualquier nodo que lo escuche y esté en cobertura de algún Nodo Líder pueda convertirse en NL, al volver a realizar la elección de rol con más posibilidades, para poder dar conectividad a la rama correspondiente. También puede ser utilizado cuando un nodo líder supera el máximo de peticiones de unión y sería por tanto adecuado otro NL para repartir de forma más adecuada la carga.

En todos los experimentos realizados este mecanismo se ha demostrado totalmente eficaz y sin sobrecoste alguno en aquellos casos, que aunque poco frecuentes, se pueden dar, haciendo que la población de nodos líderes esté perfectamente adaptada a las necesidades concretas de la distribución de la red correspondiente.

Este mensaje consta del campo código:

- Código: 0x04

#### **4.2.2.5 Petición asociación con nodo padre árbol (ASSOC-REQ: ASSOCIATION REQUEST) 0x05**

Mediante este mensaje un nodo líder solicita asociarse con el propio nodo sumidero u otro nodo líder que ya tiene ruta establecida con el sumidero que hubiera resultado seleccionado después de aplicar el algoritmo correspondiente de entre los mensajes de ofrecimiento recibidos.

Los campos del mensaje son:

- Código: 0x05
- Flag UO (única opción): Indica al NL al que se solicita asociación en el árbol que el nodo solicitante no tiene en su área de cobertura más nodos líderes.

#### **4.2.2.6 Respuesta petición de asociación con nodo padre (ASSOC-REPLY: ASSOCIATION REPLY) 0x06**

Es el mensaje que envía un Nodo Líder mediante CSMA para notificar la aceptación o rechazo a la petición de asociación en el árbol a otro nodo líder.

Consta de los siguientes campos:

- Código: 0x06
- Asociación: Un valor "1" indica asociado, un valor "0" rechazado.

#### **4.2.2.7 Información planificación descendiente a nodo padre en árbol (DINFO: DESCENDANT INFORMATION) 0x07**

Mediante este mensaje un nodo líder informa a su nodo padre en el árbol de la composición de su clúster así como de la de todos sus descendientes. La correcta recepción de este mensaje debe ser indicada mediante el correspondiente mensaje SICI (código 0x08) que le enviará posteriormente el nodo padre.

Los campos del mensaje son:

- Código: 0x07
- Slot: Ranura a partir del cual tendrá la información disponible para enviarla al padre en el árbol.

#### 4.2.2.8 Asignación de planificación y dirección lógica desde nodo padre árbol (SICI: SON INTER-CLÚSTER INFORMATION) 0x08

Este mensaje transmitido usando CSMA permite asignar a un nodo hijo en el árbol tanto su ranura de tiempo para transmisión inter-clúster como su identificador de dirección de nivel 3 en la red y la de su descendencia por tanto. No es necesario el reconocimiento (ACK) de este mensaje, puesto que si el nodo hijo no recibe esta asignación volverá a solicitarla mediante un nuevo mensaje DINFO (código 0x07).

Este mensaje consta de los siguientes campos:

- Código: 0x08
- Slot de tiempo asignado para la transmisión al nodo padre en el árbol
- Asignación de dirección para el nodo y su descendencia: El NL asigna la dirección de nivel 3 para el nodo. Los '*p*' primeros bits de dicha dirección identifican la rama en el árbol con lo que el NL los deberá mantener inalterados para todas las direcciones que asigne a sus descendientes, utilizando solo el resto de bits para asignar direcciones por orden a sus hijos en el árbol (sub-ramas). Cada descendiente en el árbol actuará de dicha forma.

#### 4.2.2.9 Notificación nodo sustituto (SN: SUBSTITUTE NOTIFICATION) 0x09

Es el mensaje que utiliza el Nodo Líder para designar los nodos sustitutos para el cargo de nodo líder en ese clúster durante ese ciclo, de forma que ejerzan el cargo en cuanto detecten la caída del Nodo Líder (ausencia de reconocimientos y/o mensajes de consulta). Los nodos sustitutos se encuentran ordenados, siendo el primero el que pasaría a ejercer de nodo líder en caso de caída del NL actual, el segundo en caso de caída del primer sustituto y así sucesivamente. Para la elección de los nodos sustitutos el NL se basa en la cercanía de los nodos y su energía restante comparada con el umbral en ese momento. Al primer nodo sustituto le será asignado el primer slot TDMA, y por orden los siguientes. La correcta recepción de este mensaje debe ser indicada mediante el correspondiente ACK (código 0x0D) al remitente.

Los campos de este mensaje son:

- Código: 0x09
- Nº de orden como sustituto
- Modo: "0" normal. "1" alta fiabilidad. En aplicaciones en las que se requiera una alta fiabilidad y baja latencia en presencia de fallos, el primer nodo sustituto de un clúster deberá permanecer encendido y monitorizando al CH durante todo el periodo de funcionamiento de éste, pasando al modo *sleep* cuando lo haga el NL, de forma que, en caso de fallo del NL en plena ronda de comunicaciones intra o inter-clúster, su

sustitución sea inmediata. Este mecanismo penaliza, evidentemente, el consumo de dicho sustituto, y por tanto acorta la vida útil del nodo.

- Lista hijos en el árbol inter-clúster
- Nodo padre en el árbol
- Planificación intra-clúster: En formato <nodo><slot>
- Planificación inter-clúster: En formato <nodo><slot>

#### 4.2.2.10 Fallo nodo sumidero (SINK-FAILURE) 0x0A

Mediante este mensaje enviado por difusión se notifica que ha caído el nodo sumidero (raíz) del árbol a los nodos líderes integrantes del mismo, para que apliquen el procedimiento de construcción del árbol hacia otro sumidero.

Los campos de este mensaje son:

- Código: 0x0A
- ID sumidero

#### 4.2.2.11 POLL {intra-clúster | inter-clúster} 0x0B

Mediante este mensaje se implementa el mecanismo de encuesta (*polling*) tanto intra-clúster como inter-clúster, para dar la oportunidad de transmitir al nodo indicado. Se envía en el canal intra-clúster indicado por el NL y en el árbol en la frecuencia establecida para el mismo. Este mensaje se emplea, además, para mantener la sincronización, así como para la transmisión de información en sentido descendente (procedente del sumidero o de otro nodo de la red). El nodo que recibe el mensaje de *poll* debe transmitir sus datos a continuación. Nótese que en los anteriores mensajes se utilizan direcciones MAC o de nivel 2, y que en este mensaje se utilizan direcciones de nivel 3 para indicar la fuente y destino final de los datos enviados “hacia abajo” aprovechando el *poll*.

Su formato es:

- Código: 0x0B
- *Time-stamp*
- Datos sentido descendente (opcional):
  - Dir. nodo fuente de los datos
  - Dir. nodo destinatario de los datos
  - Datos en formato: <Id. variable> <VALOR>

#### 4.2.2.12 Transmisión de datos {intra-clúster | inter-clúster} (DATA) 0x0C

Son los mensajes mediante los cuales los nodos transmiten sus datos, correspondientes a la fase de comunicación. En este mensaje también se utilizan direcciones de nivel 3 para indicar la fuente y destino final de los datos enviados. Normalmente, dicho destino será el sumidero, pero el protocolo también permite envíos a cualquier otro nodo de la red, o envíos del sumidero a otro(s) nodo(s).

Puede solicitarse que la correcta recepción de este mensaje sea indicada mediante el correspondiente ACK (código 0x0D) al remitente. Además, el uso de las direcciones lógicas en vez de las direcciones MAC permite que los mecanismos de tolerancia a fallos incluidos sean en la mayoría de los casos transparentes, y se reduzca el tiempo necesario para el correcto funcionamiento de la red tras un fallo, reduciéndose también por tanto el consumo de energía y el tiempo en las comunicaciones.

Su formato es:

- Código: 0x0C
- Dirección Fuente.
- Dirección Destino.
- Flag AR (Ack Requerido)
- DATOS: <Id. variable> <VALOR> (para cada magnitud sensorizada).

#### 4.2.2.13 Reconocimiento {intra-clúster | inter-clúster} (ACK: ACKNOWLEDGMENT) 0x0D

Mediante este mensaje de código 0x0D se reconoce la correcta recepción de un mensaje que lo requiera, y deberá ser transmitido inmediatamente después de la correcta recepción del mensaje.

### 4.3 Operación del protocolo

EDETA es un protocolo adaptativo, basado en una jerarquía de clústeres y árboles, que usa pseudo-aleatorización auto-regulada para distribuir de manera equitativa el gasto energético entre los nodos de la red, con el objetivo de maximizar la vida de la misma. En EDETA los nodos se organizan en clústeres, donde un nodo actuará de *nodo líder* (NL). Los NL tienen la responsabilidad de organizar las comunicaciones dentro de su clúster, opcionalmente realizar agregación de datos tal como contempla el protocolo, y participar en el protocolo inter-clúster para hacer llegar la información al sumidero. El rol de NL se distribuye entre todos los nodos de la red, aumentando significativamente el tiempo de vida de la misma [BOUKER09]. Además, a diferencia de otras aproximaciones [HEINZEL00, DEOSAR09], el NL no debe transmitir

obligatoriamente la información de su clúster directamente al sumidero, lo cual presenta dos ventajas significativas: por un lado, ahorra mucha energía, puesto que no requiere transmisiones de alta potencia que agotan en pocas rondas la batería. Por otro lo hacen muy escalable y apto, por tanto, para grandes redes de sensores. En lugar de realizar dichas transmisiones directas al sumidero, los NL utilizarán una segunda estructura en árbol que se ha demostrado muy eficiente desde el punto de vista energético. Este árbol está formado por todos los NL de la red, y su raíz es un sumidero. Además, el protocolo contempla la posibilidad de realizar compresión y agregación de datos en cada NL, reduciendo por tanto el tráfico en el árbol. Todo ello contribuye a mejorar por tanto las prestaciones y el consumo de energía.

El mecanismo de distribución del rol de nodo líder es totalmente distribuido. Los nodos se auto-proclaman NL durante la fase de elección de una forma pseudo-probabilística cada cierto tiempo, siempre y cuando su nivel de batería esté por encima de un umbral, no siendo necesaria ninguna negociación ni sobrecarga. De esta forma, en cada instante de tiempo se tendrá un conjunto distinto  $C$  de nodos líder. Los nodos que se proclaman NL difunden su estatus a los otros sensores. Cada nodo sensor determina a qué clúster desea pertenecer, eligiendo siempre el que requiera menos energía (esto no significa necesariamente que sea el que está más cerca). Los NL intentarán equilibrar el tamaño de los clústeres del mismo nivel. Sin embargo, los clústeres más cercanos al sumidero serán más pequeños, y cuanto mayor sea la distancia al mismo –en número de saltos– mayor será el tamaño de clúster. De esta forma se compensa la mayor carga que sufren los nodos más cercanos al sumidero debido al reenvío de toda la información de los nodos de niveles inferiores. Esta medida, contribuye también a maximizar la vida de la red, ya que permite un consumo de energía similar en todos los nodos independientemente de su cercanía al sumidero. Una vez se han establecido los clústeres, los NL planifican la actividad del clúster según el número de nodos que lo forman y parámetros de la aplicación (frecuencia de muestreo principalmente), lo cual permitirá a los nodos “hoja” permanecer dormidos la mayor parte del tiempo, solo despertando el intervalo que el NL les ha asignado para la transmisión de sus datos, en la planificación del clúster realizada. Cuando el NL tiene los datos de todos los nodos de su clúster, lleva a cabo la agregación de datos, y mediante el algoritmo inter-clúster envía los datos comprimidos al sumidero.

El diseñador del sistema debe determinar, a priori, el número óptimo de NL que debería haber en la red. Si hay menos de los necesarios podrían haber nodos huérfanos, y además los clústeres serían muy grandes, lo cual afectaría a las prestaciones (*delays*). Por otro lado si el número de NL es excesivo, no se estaría repartiendo la carga que supone ejecutar dicho rol de una forma eficiente. Por los resultados obtenidos en los experimentos realizados, se concluye, como sucede en [HEINZEL00] que el valor óptimo es de un 5% de los nodos totales. Además, como se verá más adelante el protocolo incorpora mecanismos de auto-regulación de la población y distribución de los NL.

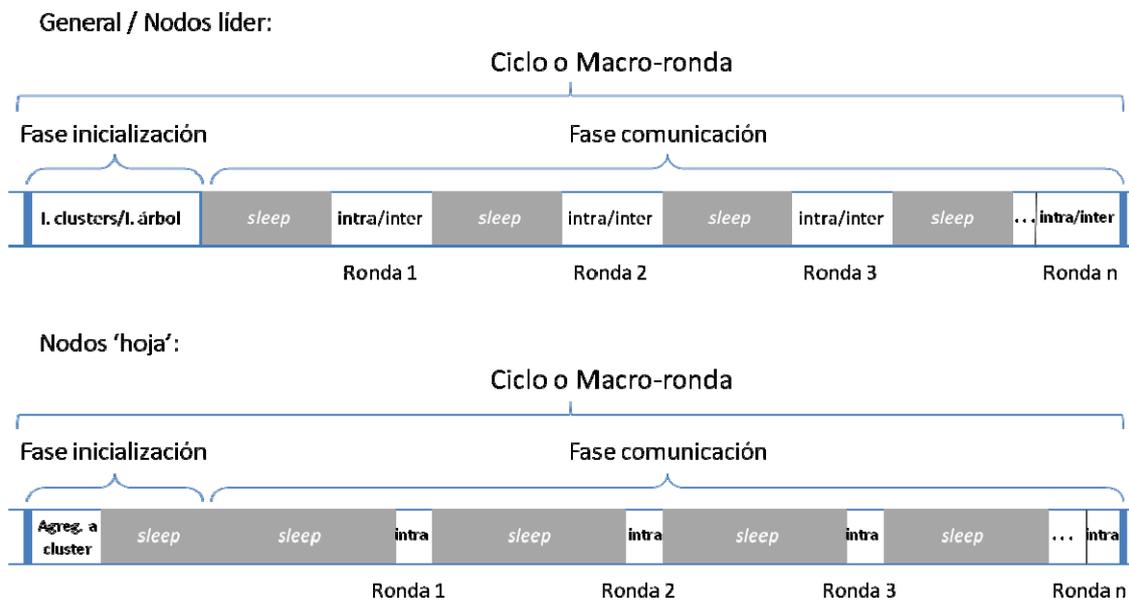


Figura 4.4. Operación EDETA.

El funcionamiento del protocolo, como podemos observar en la figura 4.4, está dividido en macro-rondas, constando cada una de las siguientes fases:

- **Fase de inicialización:** en la que se configura la red.
- **Fase de operación (ó comunicación):** más larga que la de inicialización para minimizar sobrecarga, en ella se suceden una serie de rondas de comunicación. Según los requisitos de la aplicación (frecuencia de las medidas, etc.) se ajustarán los tiempos de ronda y descanso (*sleep*) para minimizar el gasto energético.

### 4.3.1 Fase de inicialización

Esta fase se compone de las siguientes etapas:

#### 4.3.1.1 Elección de rol

En esta fase los nodos aptos para proclamarse NL (elegibles) han de determinar si van a ejercer o no el papel de NL en la ronda actual. Para tomar esta decisión, un nodo debe comprobar que su energía restante está por encima de un umbral dinámico  $E(n)$  calculado según la expresión (4.1) ajustada mediante simulación.

$$E(n) = E_{NL} \times \frac{2T_{inic}}{2T_{inic} + NR \times T_{ronda}}, \quad n \in N \quad (4.1)$$

Donde  $E_{NL}$  es la energía media restante en los NL escuchados recientemente, dado que los NLs en su mensaje de notificación de rol deben indicar su nivel de energía,  $T_{inic}$  es la duración de la fase de inicialización,  $NR$  es el número de rondas que componen una macro-ronda ó ciclo y  $T_{ronda}$  es la duración de cada ronda. La primera vez que se ejecuta este cálculo se toma como valor de energía restante de los NL el máximo. La motivación de este umbral es evitar que un nodo líder se quede sin batería en el ejercicio de dicho rol. Además, dicho umbral debe ser dinámico pues de lo contrario una vez las baterías bajaran de cierto nivel ya ningún nodo podría ser NL y por tanto la red dejaría de funcionar.

Si la energía del nodo supera esta  $E(n)$ , éste elige un número aleatorio entre 0 y 1, y si este número está por debajo de un umbral  $T(n)$  calculado según (4.2), entonces el nodo ejercerá de NL en la ronda actual.

$$T(n) = \frac{c}{|N| - 2c} \times t, \quad n \in N \quad (4.2)$$

Donde  $c$  es el número de clústeres que deseamos en la red,  $N$  es el conjunto de nodos de la red, y  $t$  es el parámetro temporal que dependerá del instante en el que se evalúa la ecuación, para la primera configuración al inicio valdrá  $t = 1$ . Como se explicará más adelante, durante la fase de inicialización, existe un mecanismo posterior para adaptar la población y distribución de nodos líderes, con lo que un nodo hoja inicial podría convertirse en nodo líder, y para ello, al evaluar la expresión (4.2) se tomarán valores de “ $t$ ” mayores, de forma que se incremente la probabilidad de ser NL al re-evaluarse más veces esta expresión en un mismo ciclo. Esta variable se reinicializa después del conjunto de rondas del ciclo.

Con todo ello, la expresión (4.2) propicia que un nodo que ha sido nodo líder no vuelva a serlo al menos hasta dos rondas después, por esto tiene el término  $2c$  en el denominador. De esta forma reparte la tarea de Nodo Líder, distribuyéndose el consumo de energía de forma equilibrada entre todos los nodos de la red, al ajustarse la población de nodos líderes a la proporción que se ha demostrado suficiente para proporcionar los servicios en la red y no excesiva para minimizar el impacto energético.

Además, existe otro mecanismo posterior, basado en el mensaje `NEED_LEADER` (código 0x04) para terminar de adaptar la población de nodos líderes a las necesidades exactas de la distribución geográfica de la red.

Por otro lado, el protocolo también contempla la posibilidad de fijar en un nodo (aquel que, por ejemplo, posea sistema de alimentación perpetua en redes heterogéneas) la auto-elección como NL en todas las macro-rondas.

Como hemos visto pues, y a diferencia de otras propuestas [ALKARAK00, DEOSAR08], EDETA considera la energía restante del nodo en comparación a los de su entorno para tomar la decisión del rol a ejercer, de forma que se pueden desplegar en la red bajo el mismo protocolo nodos con capacidades diferentes. Esta posibilidad hace que el protocolo no esté limitado a WSN homogéneas donde todos los nodos son iguales, sino que también es aplicable a WSN heterogéneas, cada vez más en auge [BOUKER09], puesto que se elimina la asunción que

hacen otras aproximaciones de que el ejercicio del rol de NL consume la misma cantidad de energía en todos los nodos. Así también se evita que un NL se quede sin batería ejerciendo el dicho rol, que aunque es una hipótesis de fallo contemplada por el protocolo, no se producirá mientras a su alrededor queden nodos con mayor reserva energética. En caso de que existan estos nodos con una fuente perpetua de energía, mediante la expresión (4.2) se consigue que el resto de nodos a su alrededor no adquieran nunca el rol de NL, y por tanto prolonguen la vida de sus baterías.

Nótese que todos los mecanismos de regulación para maximizar la vida de la red son totalmente automáticos, es decir, los propios nodos se configuran según la forma en la que hayan sido desplegados y sus capacidades sin necesidad de intervención externa (transparencia), y son capaces de adaptarse a los cambios conforme van ocurriendo en la red (caída de nodos, aparición de nuevos con batería recargada, nodos con diferentes capacidades y fuentes de energía, etc.).

#### 4.3.1.2 Notificación y formación del árbol

Cada nodo que se haya proclamado NL en el ciclo actual elegirá aleatoriamente, de entre los posibles canales de radiofrecuencia disponibles –exceptuando el canal definido para las comunicaciones en árbol- qué canal utilizará para las comunicaciones en su clúster. Cuando el NL recibe algún mensaje LEADER de otro NL o de un sumidero, y puedan con dicha información recibida indicar cuál será su nivel en el árbol, notificará su rol y el canal elegido mediante la difusión periódica del mensaje LEADER (0x01), indicando su nivel en el árbol. Si un NL escucha, antes de transmitir su notificación, que otro NL con bastante potencia de señal va a utilizar el mismo canal que él ha elegido, volverá a elegir otro canal.

En esta etapa, los NL usan el protocolo MAC CSMA. Los nodos que no son NL (nodos hoja por tanto) deberán mantener sus receptores en marcha durante esta etapa para escuchar las notificaciones de los NL. Cuando finalice esta etapa cada nodo hoja decidirá el clúster al que pertenecer en la ronda actual. Esta decisión se basa en la calidad de la señal recibida en la trama de notificación. El NL cuya notificación se haya recibido con la mayor calidad de señal es el NL con quien se puede comunicar gastando menos energía. En caso de empate se escogerá el nodo líder con nivel más pequeño (menos saltos hasta el sumidero) con lo que además reduciremos latencias.

En caso que un NL no escuche la notificación de ningún otro NL deberá emplear el mecanismo *NEED\_NL* para obtener en su área de cobertura un NL con conectividad en el árbol hacia el sumidero (raíz). Dicho mecanismo consiste en el envío del mensaje *NEED\_NL* (código 0x04), para que cualquier nodo hoja (que durante esta fase están en estado de escucha para decidir qué NL escoger), que esté dentro de la cobertura de un NL con conectividad en el árbol y escuche dicho mensaje *NEED\_NL* pueda convertirse en NL, al volver a realizar la elección de rol con un valor de  $t$  mayor (con lo que tendrá mayor probabilidad). Gracias a este mecanismo conseguimos adaptar la población y distribución de NL de forma dinámica para satisfacer las necesidades concretas de la topología de la red en cada momento. En cuanto un nodo pase a

ser NL deberá notificar inmediatamente su nuevo rol para ofrecerse al NL aislado y para indicar a otros nodos que pudieran haber recibido el mensaje y cumplan los requisitos que ya se ha resuelto la petición.

Si un NL durante esta fase no escucha nada, u observa tráfico normal de la red (rondas intra/inter clúster), decide que se ha puesto en marcha cuando la red ya está en funcionamiento (debido, por ejemplo, a que se ha recuperado de un fallo o a que ha sido añadido con posterioridad) y por tanto debe pasar a ejercer el papel de nodo hoja e intentar integrarse en algún clúster.

En paralelo con la notificación de rol se lleva a cabo la construcción de los árboles de nodos líderes según el diagrama de estados de la figura 4.5, puesto que al difundir el rol, los nodos líderes también proporcionan toda la información necesaria para llevar a cabo dicha tarea. Este mecanismo resulta muy adecuado, eficiente e inmediato.

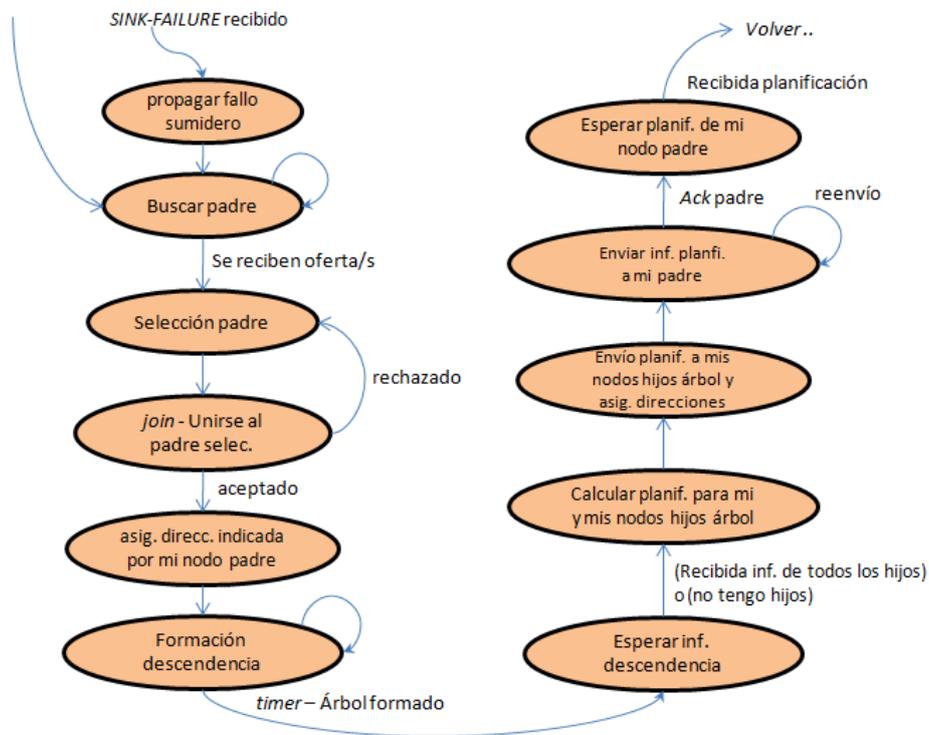


Figura 4.5. Diagrama de estados NL: procedimiento de formación árbol

Para la formación de cada árbol, los nodos NL se asociarán con el sumidero o con otro nodo líder que ya tenga ruta hacia el sumidero que resulte más apropiado para minimizar el consumo y reducir tiempos de transmisión hacia el sumidero. Para ello, tanto el sumidero como todo NL ya asociado difundirán periódicamente un mensaje LEADER (tipo 0x01), como se ha explicado anteriormente, indicando en el mismo su nivel, número de hijos y estado de la batería, ofreciéndose de esta forma como nodo padre en el árbol y facilitando la información necesaria para que otros nodos puedan seleccionar acertadamente su nodo padre.

Para decidir con qué nodo asociarse en el árbol, los nodos líderes, tras un tiempo, y si se han recibido ofertas, proceden a ordenarlas aplicando en la expresión (4.3) la información proporcionada por cada potencial padre, en la que se pondera tanto el número de saltos al sumidero que presenta el posible padre (nivel), como la calidad de la señal recibida (LQI), el número de hijos que ya tiene (nº hijos) y su nivel de batería (Energía). Esta expresión presenta determinados índices (k, k1, k2) que van a condicionar el peso de cada uno de estos factores en la decisión final. Se han obtenido los valores óptimos de estos índices mediante simulación.

$$k_1 \left( \frac{1}{\text{nivel} + 1} \right) + (k_2 \times LQI) + k_3 \left( \frac{\text{Energía}}{n^\circ \text{ hijos} + 1} \right) \quad (4.3)$$

Una vez seleccionado el nodo padre más adecuado (el primero en la lista ordenada construida) se le envía un mensaje ASSOC-REQ (tipo 0x05) –inicialmente con el *flag* UO (“única opción”) desactivado–, que el nodo padre evaluará, y responderá con un mensaje ASSOC-REPLY (tipo 0x06). Mediante este mensaje el nodo padre puede rechazar la solicitud (por ejemplo, si ya ha alcanzado su número máximo de hijos), o aceptarla, en cuyo caso el nodo solicitante ya forma parte del árbol (con nivel más que el nodo padre) y puede aceptar a su vez nodos hijos. En caso que haya sido rechazado intentará asociarse con el siguiente nodo padre en la lista priorizada y así sucesivamente. Si finalmente no es aceptado por ningún posible padre, lo intentará de nuevo pero con el *flag* UO activado, de forma que el nodo a quien lo dirija no puede negarse a asociarlo. Para mejorar la seguridad del protocolo, los nodos pueden incluir opcionalmente una *huella*, que se utilizará en la petición de asociación para que el nodo padre pueda verificar que se trata de un nodo “legítimo” y en caso contrario denegarle la asociación.

### 4.3.1.3 Establecimiento y planificación de los clústeres

Una vez cada nodo decide a qué clúster desea agregarse (según calidad de señal y nivel), debe informar al nodo líder (NL) de dicho clúster que desea ingresar en el mismo mediante el mensaje JOIN-REQ (0x02). Los nodos transmiten dicha información al NL utilizando el protocolo MAC CSMA. Durante esta fase los NL deberán mantener sus receptores encendidos. Con el objeto de acotar y equilibrar el tamaño de los clústeres, los NL admitirán un máximo predeterminado de nodos, de forma que, cuando reciba peticiones enviará los mensajes de aceptación o rechazo a los nodos solicitantes (no pudiendo rechazar las solicitudes con el *flag* UO activado para evitar nodos huérfanos). Además, el tamaño de los clústeres no será uniforme en todos los niveles de un árbol, para compensar la carga extra que supone ejercer de NL más cerca del sumidero en cada posible árbol de la red, dado que tiene que retransmitir todo el tráfico de todos los niveles inferiores hacia el sumidero. Por tanto, los clústeres serán más pequeños cuanto más cercan estén del sumidero. De esta forma, el número máximo de nodos en un clúster en función de su nivel vendrá dada por la expresión (4.4), que ha sido verificada y ajustada mediante simulación:

$$TC(\text{nivel}) = T_{\text{mín}} + [T_{\text{mín}} \times (0,05 \times \text{nivel})] \quad (4.4)$$

Donde TC es el Tamaño de Clúster en función del nivel del nodo NL y  $T_{\min}$  es el tamaño mínimo predeterminado de clúster.

De esta forma se consigue repartir la carga equitativamente entre los NL, lo cual facilita la sincronización entre clústeres. Esta mejora en la sincronización resulta, además de beneficiosa para los retardos de comunicación, muy conveniente desde el punto de vista energético, ya que todos los nodos soportarán una carga de trabajo similar, y por tanto agotarán su energía de forma pareja.

Un NL notifica la pertenencia al clúster a un nodo hoja solicitante respondiendo a su mensaje JOIN-REQ (código 0x02) con un mensaje JOIN-REPLY (código 0x03). En dicho mensaje JOIN-REPLY el NL le indicará al nodo hoja cual es su *slot* TDMA intra-clúster y el canal de radio en dicho clúster, al cual deberá conmutar a continuación. Los NL asignarán *slots* consecutivos a los nodos hoja que va asociando, dejando ventanas temporales para compensar posibles derivas. En cuanto un nodo hoja es aceptado en un clúster pasa al estado *sleep* durante el tiempo indicado por su NL. Al igual que en el caso de la formación del árbol, y como mecanismo de seguridad del protocolo, opcionalmente los nodos pueden incluir una *huella*, que utilizarán los nodos hoja en la petición de unión al clúster, de forma que el NL pueda verificar que se trata de un nodo “legítimo” y en caso contrario rechazarlo en el clúster. Indicar que los NL podrán seguir admitiendo nodos en sus clústeres una vez finalizada la etapa de configuración, para poder conectar nuevos nodos añadidos o nodos que se han recuperado de cualquier tipo de fallo. Por razones de eficiencia, los NL reutilizan *slots* de nodos que han abandonado el clúster por cualquier razón, de forma que la vuelta siempre se hace como un nuevo nodo para el clúster. En caso de no disponer de *slots* que reutilizar los NL siempre añadirán a posibles nuevos miembros en *slots* posteriores al último asignado, de forma que no afecte en ningún caso la planificación del resto de nodos del clúster.

Si un nodo hoja no observa ningún tipo de tráfico durante un tiempo determinado, supondrá que, o bien se encuentra geográficamente aislado de la red, o bien que se ha conectado cuando la red ya está en funcionamiento (debido a que se recupera en ese momento de un fallo o ha sido incorporado con posterioridad). Para determinar en qué situación se encuentra, el nodo emitirá una señal en determinada frecuencia que es capaz de excitar los circuitos pasivos que deben habilitar los nodos líderes para despertarse en caso de recibir dicha señal (*radio triggered wake up*) [GU05], de forma que pasen a ofrecerse como NL's y permitan la integración del nodo en cuestión, asignarle dirección y *slot*, y pasar de nuevo (tanto el nodo NL como el nodo hoja agregado) al estado *sleep*. Si se hubiera despertado más de un NL, cuando observe/n que le solicita asociación a otro NL o pase un tiempo determinado sin noticias del nodo vuelven al estado *sleep*. Si después de la transmisión de las señales indicadas el nodo sigue sin observar ningún tráfico asume que está aislado del resto, situación ante la cual, como mal menor, dormirá periodos pseudo-aleatorios de un determinado número de rondas para ver si más adelante es posible integrarse en la red.

#### 4.3.1.4 Planificación del árbol

Una vez formados los árboles y clústeres, los nodos hoja han comenzado su periodo de *sleep* y tan solo quedan despiertos los NL's, que proceden a configurar las comunicaciones TDMA-poll

en árbol. Para ello, los nodos esperan la información sobre su descendencia (recepción de mensajes DINFO – tipo 0x07 –), que contiene el tiempo que necesita cada hijo para tener listos tanto los datos de su clúster como los datos de sus hijos en el árbol. Con esta información pueden indicar a su padre cuánto tiempo necesitan para tener listos sus datos y los de su descendencia. De esta forma, cuando un NL haya recibido los mensajes DINFO de todos sus hijos en el árbol será capaz de asignarles de forma oportuna un *slot* TDMA acorde con sus necesidades, notificadas mediante mensajes SICI (código 0x08), y estará en condiciones de transmitir su mensaje DINFO a su nodo padre en el árbol para que éste a su vez le pueda asignar *slot*. Para ello, el padre considerará los costes temporales de recolección en cada uno de sus hijos, y sus necesidades, para dormir, por ejemplo, y dejará siempre ventanas temporales de seguridad. Una vez contempladas estas actividades, el nodo padre puede pasar a programar los *timers* según la planificación realizada. Mediante este método la planificación en árbol se realiza de forma totalmente distribuida, evitando los sobrecostes (tanto en tiempo como en energía) de otras aproximaciones centralizadas en las que se necesita llevar toda la información de los nodos hasta el sumidero para que éste de forma centralizada realice la planificación, y luego la difunda a cada nodo.

Los slots asignados no quedarán vacíos dado que aunque caiga el NL hijo que lo tuviera asignado lo tomará el nuevo NL-sustituto de aquel. Los nodos líderes en el árbol pueden añadir más hijos (NL en el árbol) después del periodo de configuración, para adaptar la red a caídas de sumideros, gracias al estado de notificación de rol y escucha tras las rondas intra e inter clúster. Cuando se añade un nuevo hijo en el árbol de esta forma se añade en slots previos al primer slot asignado, de forma que no afecte a la planificación del resto de nodos hijos en el árbol ni a la planificación para el envío al respectivo nodo padre, dado que el NL que haya adoptado nuevos hijos seguirá teniendo disponible la información en el mismo instante de tiempo que antes.

### 4.3.2 Fase de operación

Una vez se han creado los árboles y clústeres y se ha establecido la planificación comienza la fase de operación. Esta fase será más larga que la de inicialización para minimizar la sobrecarga. Opcionalmente todas las comunicaciones en la red de sensores (tanto a nivel de clúster como de árbol) podrán ser cifradas con el fin de proporcionar una comunicación segura.

#### 4.3.2.1 Protocolo de comunicación *intra-cluster*

Dentro de los clústeres, los nodos hoja se despertarán con la antelación necesaria para leer sus sensores y estar dispuestos a la transmisión al nodo líder de los datos obtenidos en su *slot* intra-clúster, utilizando para ello su dirección de nivel 3, como puede observarse en el DTE de los nodos hoja en la figura 4.6. El destinatario final de dicha transmisión será usualmente el sumidero, pero, gracias al mecanismo de comunicación bidireccional que incluye el protocolo, un nodo puede enviar información a cualquier otro nodo de la red, indicándolo mediante la correspondiente dirección destino de nivel 3.

Es destacable que para los tiempos y granularidad de la mayoría de aplicaciones de redes de sensores, en las que resulta habitual que las mediciones se realicen cada varios minutos o incluso horas, el reloj interno (disponible en prácticamente todos los microcontroladores más usuales) es suficiente para la sincronización dentro del TDMA intra-clúster. De forma adicional se prevén en dichas rondas intra-clúster la utilización de ventanas temporales para equilibrar posibles derivas de los mismos. No obstante, si la aplicación requiriera mayor precisión y granularidad podría incorporarse un reloj de tiempo real en los nodos (RTC), que se sincronizaría nuevamente en cada ronda. Para la transmisión de los datos medidos hacia el NL se requiere mínima energía posible, ya que se ha elegido el NL que se alcanza con la mínima potencia, y el nodo vuelve inmediatamente al estado *sleep* después de finalizar con éxito dicha transmisión. Además, el protocolo dispone de un mecanismo de sincronización basado en el mensaje de *poll* transmitido por los nodos líderes, y que permite ir compensando las diferencias entre el reloj del NL y el reloj local de los nodos para homogeneizar la duración de los intervalos de tiempo medidos.

El NL notificará en la primera ronda de comunicación de cada ciclo al nodo o nodos seleccionados como sustitutos su elección para tal papel, mediante un mensaje SN (código 0x09), donde le aporta toda la información necesaria para ejercer el rol de líder (planificación intra e inter, nodo padre en el árbol y lista de nodos hijos en el árbol). Ello permitirá tolerar un fallo del NL actual, ya que la operación seguiría sin complicaciones por parte del sustituto. Se realiza en este momento debido a que hasta que el NL no haya completado el clúster no puede decidir cuál es el mejor o mejores candidatos a nodo sustituto, maximizándose además de esta forma el ahorro energético debido a que los nodos hoja correspondientes no han tenido que estar en marcha más tiempo del normal (más tiempo que cualquier otro nodo hoja). Además, el NL con la notificación otorga el primer slot al primer nodo sustituto (así como con los sucesivos slots a los sustitutos de reserva). Para ello debe cambiar al nodo o nodos que ocupaban estos slots, acción que se realiza mediante un *flag* en el propio ACK de datos.

La selección del nodo líder sustituto se basa en elegir a aquellos nodos del clúster que estén más próximos al NL (aquellos que haya visto con mejor calidad de señal), puesto que cuanto más cercano se encuentre menos variarán las condiciones -distancia al resto de nodos del clúster y a los NL's padre e hijos del NL actual- en caso de fallo del NL. Los nodos sustitutos ocuparán por orden los primeros slots de las rondas, de forma que serán los primeros en detectar la caída del nodo líder para asumir sus funciones (con su dirección de nivel 3) siendo este cambio transparente al resto de nodos. Adicionalmente, el protocolo contempla mecanismos que se explicarán más adelante para resolver situaciones en las que algún nodo -bien nodo hoja o bien otro nodo líder en el árbol- no alcanzara al sustituto en caso que éste entrara en acción.

Si a pesar del mecanismo basado en nodo sustituto, un nodo hoja de un clúster no consigue transmitir a su nodo líder tras dos rondas consecutivas, se prevé un mecanismo para que dicho nodo trata de unirse a otro clúster, pasando al estado de escucha de ofertas de NL. Dado que los NL al final de sus rondas intra e inter clúster notifican su rol para ofrecerse a posibles nodos hoja, estos nodos podrán en ese instante seleccionar y solicitar adhesión a otro clúster.

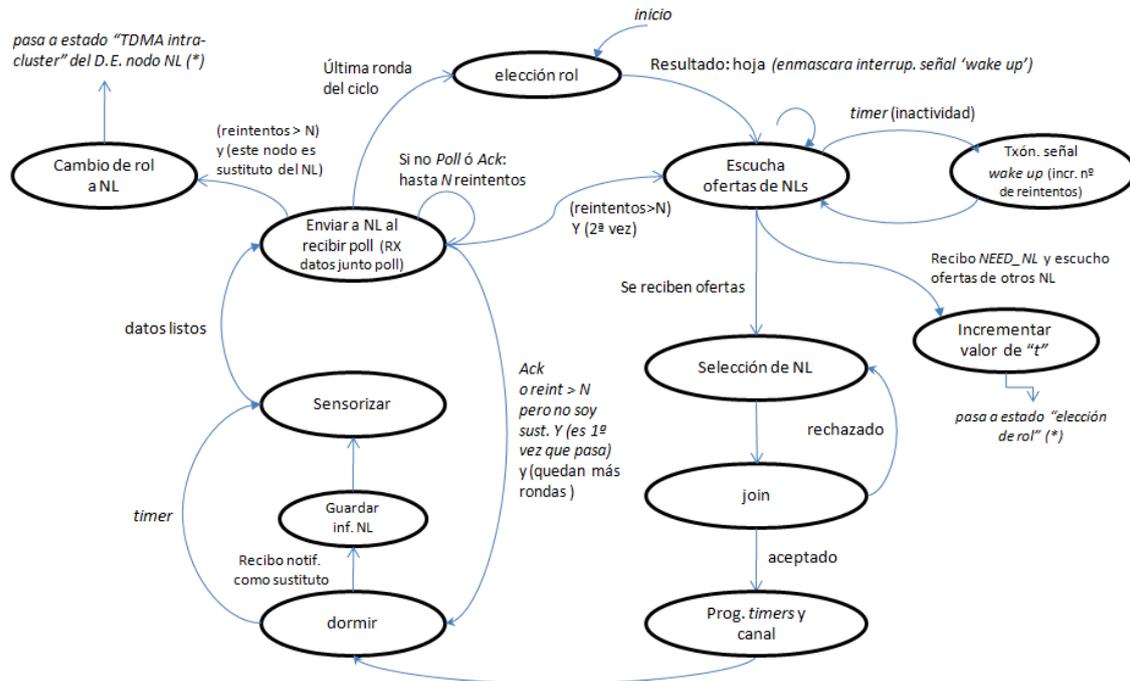


Figura 4.6. Diagrama de estados nodo normal.

Opcionalmente se puede activar para las comunicaciones intra-clúster un mecanismo de *polling*, de forma que el esquema de comunicaciones sería TDMA-polling, en el que los nodos se comportan como se ha descrito anteriormente, pero cuando llega su slot transmitirán una vez se lo indique el NL. Además dicho mensaje de *poll* será utilizado para mantener la sincronización con mayor precisión, implementándose adicionalmente la comunicación bidireccional, dado que el mensaje de poll que envía el NL hacia el nodo hoja puede contener datos destinados para el nodo encuestado procedentes del sumidero u otro nodo de la red. Por otro lado, el mensaje de *polling* podrá ser cifrado para mejorar la seguridad de la red.

Una vez finalizada la transmisión al NL y recibido el ACK correspondiente, tanto el subsistema de radio de cada nodo hoja como el resto de los módulos del nodo pueden apagarse hasta el próximo slot de tiempo asignado, es más, no solo puede apagarse la radio sino todos los módulos del nodo para maximizar el ahorro de energía. El NL debe mantener su receptor encendido para recibir todos los datos de los nodos de su clúster. Cuando ha recibido todos los datos, el NL puede efectuar agregación de datos (opcional según la aplicación) o compactarla toda ella en el menor número posible de tramas. Los nodos NL una vez terminan la ronda TDMA intra-clúster, pasan a ejecutar el *inter-clúster routing*.

Para minimizar las interferencias y colisiones entre clústeres adyacentes, y tal como se ha indicado anteriormente, en cada clúster se utilizará un canal diferente que el NL elegirá pseudo-aleatoriamente cuando se forma el clúster. Para las comunicaciones inter-clúster, los nodos NL utilizarán un canal común predeterminado. Se descarta garantizar la asignación de canales diferentes para clústeres adyacentes dado que es un problema NP completo [SCOTT96] incluso cuando hubiera un centro de control central que pudiera ejecutar los

algoritmos necesarios. Usando la opción propuesta se soluciona el problema de una forma distribuida y sencilla, que se ha demostrado eficaz en la experimentación llevada a cabo hasta la fecha.

#### 4.3.2.2 Protocolo de comunicación *inter-cluster dynamic tree*

Una vez un NL ha terminado su ronda TDMA intra-clúster, inicia la fase de comunicaciones en el árbol. Deberá conmutar al canal de comunicaciones preestablecido para el árbol, donde las comunicaciones emplearán obligatoriamente la técnica de TDMA-polling para mantener el sincronismo en cada ronda y tolerar fallos en el árbol con flexibilidad. El diagrama de estados para este protocolo correspondiente a los NL se presenta en la figura 4.7. El impacto en el consumo del mecanismo de *polling* en el árbol comparado con sus beneficios es extremadamente bajo. Además, considérese que, puesto que la población de NL's en la red será, aproximadamente, de un 5% de los nodos, el número de nodos que un NL tiene a su cargo en un clúster es muy superior al número de nodos que tendrá que gestionar en el árbol. Por lo tanto, en comparación con las comunicaciones intra-clúster, la aplicación del polling en las comunicaciones inter-clúster a muy pocos nodos NL hijos no afecta significativamente al consumo, y permite una pronta detección de fallos, una comunicación eficiente, acotada en el tiempo y sin colisiones en el árbol (el TDMA evita colisiones de otros nodos del árbol, y la utilización de una frecuencia distinta a las que se utilizan en los clústeres evita las colisiones con nodos hoja). Opcionalmente los mensajes de *poll* en el árbol podrán cifrarse para mejorar la seguridad de la red.

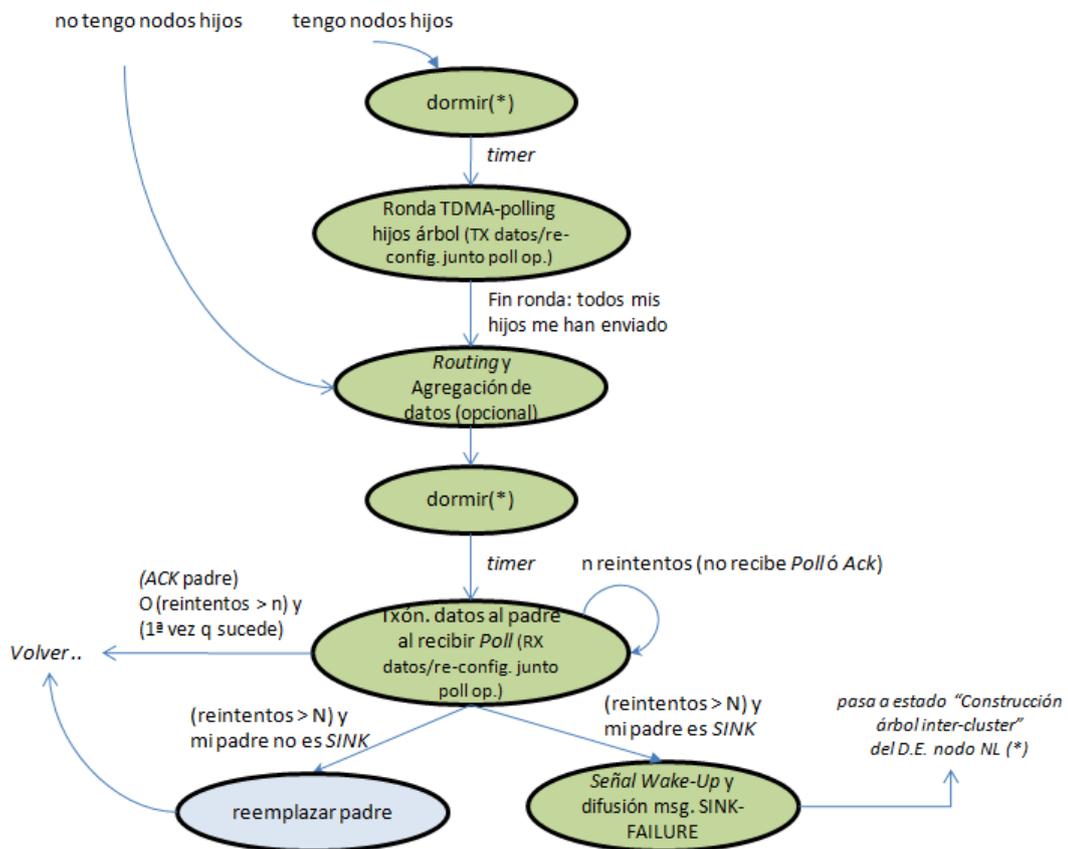


Figura 4.7. Diagrama de estados NL: inter-clúster routing.

La primera fase dentro del protocolo inter-clúster consiste en recibir de cada uno de los nodos hijos en el árbol la información que han recabado de sus clústeres (durante la fase intra-clúster) y la que ha recibido a su vez de sus nodos hijos en el árbol. Así pues, el NL en cuestión hará *polling* en el slot correspondiente a cada uno de sus nodos hijos en el árbol. Una vez se disponga de los datos de las ramas inferiores del NL, se procederá a la agregación de datos junto con los datos recabados en su propia ronda TDMA intra-clúster, preparando dicho NL la información para el siguiente salto hacia su destino, en el que será transmitida en el slot correspondiente una vez su nodo padre en el árbol se lo señale (*poll*), y así irá subiendo por el árbol hasta llegar al sumidero. Obsérvese que se trata de un mecanismo de encaminamiento sencillo y rápido, que no requiere efectuar ninguna operación complicada en los nodos ni requiere cantidad de memoria para almacenar tablas de enrutamiento, etc. Y por otro lado, ofrece además tiempos acotados en las comunicaciones, pues es posible establecer una cota superior del tiempo que tardará una información en llegar al sumidero desde que sale de un nodo, dado que tanto en la fase intra-clúster como en el árbol (inter-clúster) las comunicaciones están acotadas en el tiempo.

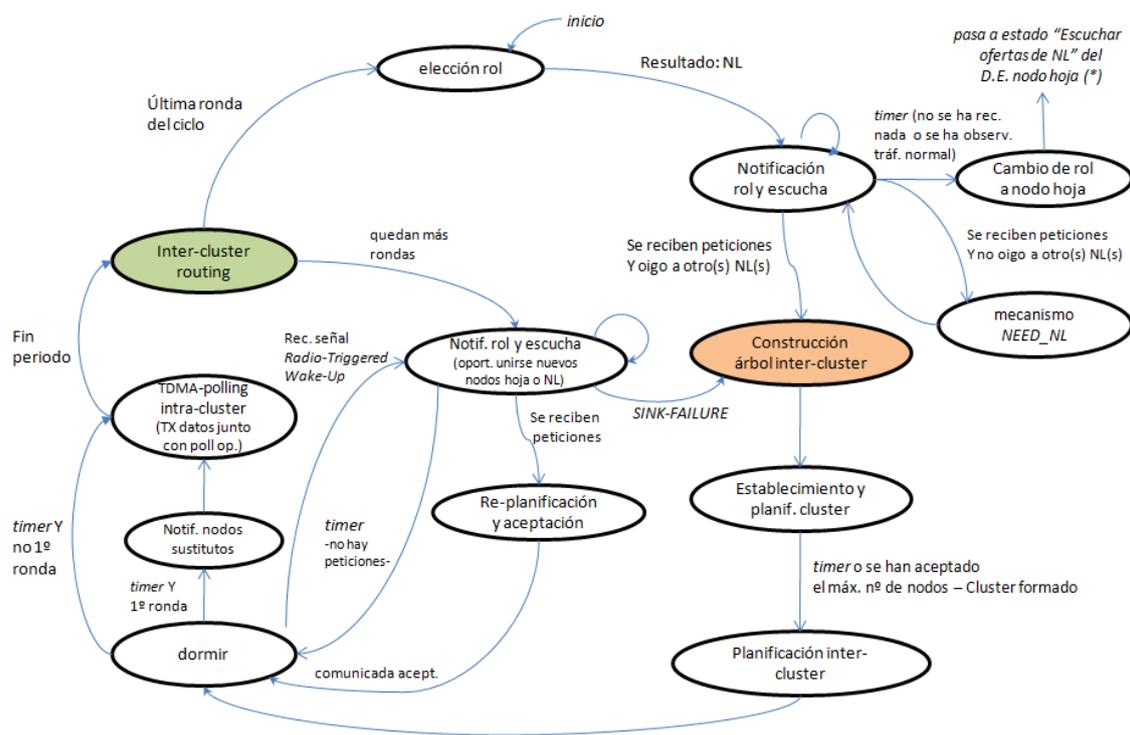


Figura 4.8. Diagrama de estados general Nodo Líder

El mensaje de *poll* que el NL padre envía a sus NL's hijos se utiliza también para mantener la sincronización y además puede contener información (procedente del sumidero u otro nodo

de la red) en sentido descendente para dicho NL, o para alguno de sus nodos hoja en el clúster o para alguno de sus nodos hijos en el árbol.

El mecanismo de enrutamiento de la información “descendente” es también sencillo y sin costes adicionales, y sin requisitos de grandes cantidades de memoria ya que no requiere de tablas de encaminamiento, dado que, según el mecanismo de asignación de direcciones descrito, un nodo líder puede encaminar en función de los “prefijos” (“p” primeros bits concatenados al principio de la dirección) por alguna de sus ramas inferiores si reconoce el prefijo como el de uno de sus hijos o hará entrega directa en algún nodo de su clúster si determina por la dirección que es un nodo asociado con él, o en caso contrario encaminará “hacia arriba” la información ya que el nodo destino está por otra rama.

Una vez concluidas estas operaciones el NL pasa al estado *sleep*, programándose el temporizador para despertarse en la siguiente ronda TDMA intra-clúster, empezando de nuevo el ciclo. El protocolo establece además, que si el tiempo que debe esperar un NL hasta hacer polling a alguno de sus hijos en el árbol una vez ha terminado su ronda TDMA intra-clúster supera un determinado umbral, el NL puede pasar hasta entonces a estado *sleep*. Igualmente sucede si el tiempo que debe esperar desde que tiene los datos preparados para enviar a su padre hasta el inicio del slot asignado para dicha transmisión excede un umbral el NL esperará ese tiempo en modo *sleep*, programándose el temporizador para estar listo antes del inicio del slot correspondiente para compensar posibles derivas del reloj local. Finalmente, hacer notar que como se ha podido comprobar, es posible tener una cota superior (en ausencia de fallos) para el tiempo que tardan los datos en llegar al sumidero desde que se tienen listos en un nodo.

Por último, una vez hecho todo lo anterior, el NL pasa durante un breve periodo al estado de notificación de rol y escucha, para dar la oportunidad de unirse a nuevos nodos (o nodos que se recuperan de algún fallo) que lo soliciten. Todos estos procedimientos pueden observarse en el DTE general de los NL en la figura 4.8.

En caso de fallo en un nodo líder del árbol, se aplicarán los mecanismos explicados en el apartado 4.4, no obstante, indicar que si a pesar de la entrada en funcionamiento del nodo sustituto, algún nodo en el árbol no alcanza a su nodo padre en el árbol, el protocolo presenta un mecanismo (figura 4.9) para adaptarse y elegir un nuevo padre, gracias a la notificación de rol que deben hacer todos los NL al final de la ronda intra e inter clúster, y por tanto oportunidad para asociar un nuevo hijo en el árbol. En este caso habrá que indicarle al padre en el momento de la asociación la ranura a partir de la cual se puede tener lista la información de su clúster y de sus hijos en el árbol, mediante el mensaje ASSOC-REQ (código 0x05), para que el nodo padre le asigne slot (que no modificará la planificación de los hijos adoptados con anterioridad, dado que siempre se inserta a los nodos nuevos en los slots posteriores) mediante el mensaje ASSOC-REPLY (código 0x06), y así re-planificarse. El nuevo padre también le dará una nueva dirección, que se asignará al nodo, y que servirá para recalcular las direcciones de los nodos hijos, que las asignará en las siguientes rondas mediante mensajes SICI (código 0x08).

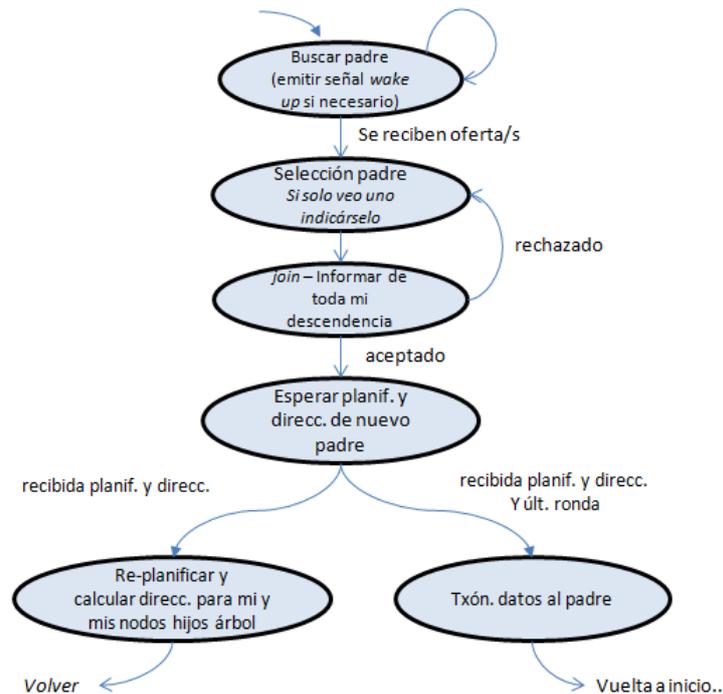


Figura 4.9. Diagrama de estados NL inter-clúster routing: Reemplazar padre

#### 4.3.2.3 Duración de los clústeres

La duración de los periodos óptima dependerá de la aplicación concreta. Deberá ser de cierto tamaño para sacarle provecho a la sobrecarga que supone la formación de los clústeres, y adecuado para repartir de forma eficiente la carga del rol de NL entre todos los nodos. En la última ronda, el NL avisa que es la última ronda y que cuando finalicen su transmisión no deben dormirse, pues se iniciará de nuevo el proceso de formación de clústeres.

### 4.4 Robustez del protocolo

A pesar que uno de los puntos fuertes en redes de sensores es precisamente la tolerancia a fallos debido a la alta replicación, ya que se disponen de muchos nodos en la red, el presente protocolo presenta una serie de mecanismos de tolerancia a fallos que lo hacen robusto y estable a pesar de la caída de nodos que desarrollen un papel importante en la red.

Todos los mecanismos propuestos se han demostrado eficaces en los experimentos realizados tanto en nodos reales como en el modelo simulado, además de presentar un impacto ligero en el consumo energético.

Hipótesis de fallo contempladas:

- Cae un nodo ejerciendo rol de NL
- Cae un nodo ejerciendo rol de hoja
- Cae nodo sumidero
- Nodos huérfanos

#### 4.4.1 Fallo de un nodo líder

Este fallo se soluciona mediante los “nodos sustitutos”. En principio, cualquier nodo puede ser “nodo sustituto”, pero si se siguen determinados criterios a la hora de elegirlos se minimizarán posibles efectos colaterales. En un clúster los mejores candidatos a nodo sustituto son los nodos más próximos al NL actual, dado que si los nodos que forman el clúster han elegido a dicho nodo por potencia de señal, un nodo que esté próximo a dicho nodo seguirá siendo una buena opción para los nodos que conforman el clúster, para poder seguir transmitiendo con la mínima potencia. Además, otro criterio para elegir nodo sustituto será que esté por encima de del umbral actual de energía restante, dado que no tiene sentido elegir un nodo sustituto sin energía. La elección de nodo sustituto la hace el NL cuando constituye el clúster con las peticiones que ha recibido. El protocolo permite que se establezca una lista priorizada de “nodos sustitutos” para reforzar la fiabilidad. Los nodos sustitutos son notificados expresamente en la etapa de inicialización del clúster, transfiriéndoles el NL toda la información que necesitan para ejercer en dicho clúster. Se les reserva siempre la primera ranura TDMA intra-clúster para que sean los primeros en detectar la caída del NL (no reciben ACK del NL tras varios intentos) y pasar a sustituirles de inmediato de forma que el resto de nodos del clúster no lleguen ni a enterarse de dicho fallo, dado que los nodos del clúster envían a la dirección lógica del clúster no a una MAC de un nodo concreto, siéndoles por tanto transparente, resolviéndose el problema sin pérdida de tiempo, lo cual es ahorro de energía, y manteniéndose por tanto las cotas temporales aún en presencia de fallos importantes como es la caída de un NL.

En cuanto a las comunicaciones inter-clúster, el nodo sustituto al terminar la ronda TDMA intra clúster, como conoce la planificación en el árbol pasará a ejecutar el protocolo inter-clúster, en el que al igual que en el caso anterior, al utilizarse para la transmisión de los datos las direcciones lógicas, será totalmente transparente el fallo en el NL.

En caso que tras la caída y toma de posesión del nuevo NL hubiera quedado algún nodo hoja huérfano (que no alcanza al nuevo NL) –circunstancia muy poco probable considerando que el nuevo NL es el nodo más próximo al anterior NL, y que los miembros del clúster habían elegido pertenecer a dicho clúster debido a que el NL que lo dirigía es el que veían con mayor potencia- el nodo buscará otro clúster (los NL dan la oportunidad al final de cada ronda de unirse). También está contemplada esta rara posibilidad en la estructura de árbol, de forma

que el nodo líder que no alcanzara al nodo padre sustituto puede también asociarse a otro NL en el árbol.

Para sistemas con requisitos de alta confiabilidad, se puede habilitar el funcionamiento del primer nodo sustituto durante todo el tiempo que está en marcha el nodo líder, es decir, que solo duerma cuando lo hace el líder, de forma que esté monitorizando continuamente su actividad y pueda detectar cualquier fallo inmediatamente, suceda en el momento que suceda.

Si por error, en un momento hubiera dos nodos líderes en activo en el clúster, el primero que lo detectara pasara al estado de reserva.

#### 4.4.2 Fallo de un nodo hoja

Este tipo de fallos es menos grave que el anterior, dado que la replicación de nodos en una red inalámbrica de sensores por sí sola es mecanismo suficiente para tolerar la caída de nodos hoja puntuales. Además, el protocolo contempla la posibilidad de fallo transitorio en un nodo, de forma que si vuelve a funcionar más tarde, el nodo intentará unirse al NL que escuche en el momento en que vuelve a funcionar. Téngase en cuenta que no tiene porque ser su NL anterior, dado que en el tiempo en que ha estado averiado puede haberse reconfigurado la red. Los NL, como se ha expuesto anteriormente, una vez han ejecutado el algoritmo intra e inter dedican unos instantes a difundir su disposición a asociar al clúster a nodos que lo requieran. Este es el mecanismo que dispone el protocolo y que sirve para unir a la red nuevas estaciones que se pongan en marcha o añadan con posterioridad a la misma. Además, si el nodo tras un intervalo de tiempo no observa tráfico normal o notificaciones de rol por parte de nodos líderes, asume que los nodos de la zona están en modo *sleep* recurriendo a la técnica de la señal *Wake-Up* para despertar al/los NLs de la zona para unirse a la red.

#### 4.4.3 Fallo de nodo sumidero

Este es sin lugar a dudas uno de los fallos más graves que pueden ocurrir en una red. La forma de tolerarlo es mediante replicación del mismo, lo cual además es recomendable no solo por tolerancia a fallos sino también por prestaciones como se ha comentado, dado que el sumidero finalmente es el cuello de botella de una red de sensores, con lo que se hace necesario a partir de cierto tamaño de la red disponer de varios sumideros, que conformarán, como se ha explicado, varios árboles de *clústeres*.

Los primeros nodos en detectar el fallo de un sumidero serán los NL de primer nivel (los que se comunican directamente con el sumidero), dado que no obtendrán contestación del mismo, estos nodos deberán comunicarlo al resto de nodos del árbol mediante el mensaje SINK-FAILURE (0x0A) indicando el sumidero que ha caído y buscar una nueva ruta (construir un nuevo árbol) hacia otro sumidero, pero el problema es que el resto de NL por debajo en el árbol ya habrán pasado a estado *sleep*, al haber concluido la comunicación inter-clúster, con lo que habrá que despertarlos mediante la transmisión de la señal *wake up* que permite

despertar los NLs como se ha explicado anteriormente (estos a su vez irán propagando en sus ramas la caída del sumidero correspondiente), de forma que se despierte al resto de nodos líderes con conectividad en otro/s árbol/es que pasarán a ofrecerse mediante la notificación de rol (mensaje LEADER 0x01) y los nodos dependientes del sumidero que ha fallado podrán solicitar unirse a ellos (mensaje ASSOC-REQ 0x05), de forma que se irá extendiendo el árbol previo (siguiendo el mecanismo explicado con anterioridad de formación de árbol) y proporcionando por tanto conectividad hacia el nuevo sumidero a todo el árbol que estaba sin conectividad. Dependiendo de la cantidad de nodos y sumideros y su distribución es posible que el antiguo árbol se reparta entre varios árboles, todo ello se realizará de forma automática y aplicando la expresión (3) para obtener la mayor eficiencia en la ruta. Hacer notar que todo este proceso es transparente y por tanto sin coste energético alguno para los nodos hoja, que constituyen la gran mayoría de la población de nodos de la red.

#### 4.4.4 Nodos huérfanos

Si un nodo resulta huérfano porque lo han rechazado el/los CH que tenía a su alcance –por *hardlimit*–, el nodo se programará el temporizador para dormir hasta el siguiente ciclo.

Si un nodo resulta huérfano después de haber sufrido una avería, porque se ha encendido en ese momento, o por cualquier otra razón, en cualquier caso no ejercerá rol de NL, escuchará para analizar posibles ofertas de NL para unirse a ellos, y en caso que no las hubiera (y no observa ningún tipo de tráfico) asume que está en tiempo de dormir de la red, por lo que empleará la técnica de la señal “*radio triggered Wake-up*” para despertar al/los NL/s que esté/n cerca (los nodos hoja tienen enmascarada la interrupción por señal *wake-up*) y así recibir sus ofertas y unirse al que vea con mejor señal (si se trata de la instalación de nuevos nodos en la red, la señal *wake-up* puede ser emitida externamente por el instalador para despertar a los NLs de la zona y permitir la inserción de los nuevos nodos, ahorrándoles energía). El NL replanificará para asignarle slot temporal, y para reprogramar sus *timers*, pero la re planificación no afectará a nadie más (ni en el clúster ni en el árbol), ya que los nuevos nodos se añadirán siempre al final de la ronda, y en todo caso lo que modificará será el tiempo de dormir –margen– que tenga el NL para iniciar luego la comunicación inter-clúster, pero nada más.

## 4.5 Conclusiones

Se ha propuesto una nueva arquitectura para redes inalámbricas de sensores, denominada *Energy-efficient aDaptative hiErarchical and robusT Architecture* (EDETA), que permite realizar un encaminamiento muy eficiente desde el punto de vista energético, a la vez que robusto y con tiempos acotados, en redes de sensores de cualquier tamaño y sin requisitos de infraestructura ni de nodos especiales, facilitando por tanto la implantación de las WSN y abriendo ampliamente el abanico de posibles aplicaciones.

No tiene *overhead* de cálculo de rutas, pero estas son dinámicas, adaptables a cambios y fallos, y óptimas, teniendo tiempos acotados y ofreciendo el máximo ahorro energético posible, sin necesidad de que los nodos sepan su posición (necesidad de GPS protocolos basados geográficos) y tolera a fallos sin hacer inundaciones lo cual es muy ineficiente desde todos los puntos de vista (protocolos *multicast*).

La evaluación de prestaciones del protocolo EDETA muestra un funcionamiento estable y óptimo de las estructuras de clústeres y árboles, con un *overhead* muchísimo menor que otras aproximaciones que realizan cálculos de rutas o difusiones para hacer llegar la información al sumidero, todo ello de forma automática. Además presenta una detección temprana de fallos en estaciones gracias al uso de esquemas TDMA. Por otro lado todos los mecanismos de tolerancia a fallos propuestos han sido evaluados, probados y ajustados tanto mediante simulación como en hardware real.

Los experimentos realizados, basados en medidas reales, demuestran que EDETA reduce en un factor de 8x la energía consumida comparado con los protocolos *power-aware* más populares, multiplicando por tanto la duración de la vida de la red, y ofreciendo además características novedosas en redes de sensores, a la par que necesarias para determinados nuevos campos de aplicación, como son mecanismos de tolerancia a fallos y tiempos acotados.

## Capítulo 5

### Experimentación

#### 5.1 Metodología de evaluación

Todos los mecanismos propuestos en la presente tesis se han evaluado mediante simulaciones. El simulador utilizado ha sido el NS-2 (network simulator versión 2) [NS09]. En la actualidad, la inmensa mayoría de grupos de investigación que trabajan en la línea de redes inalámbricas de sensores utilizan el NS-2.

El método de evaluación mediante simulación tiene numerosas ventajas especialmente cuando se utiliza un simulador estándar ampliamente utilizado en la comunidad científica, al tratarse de una herramienta suficientemente probada y que es continuamente corregida y actualizada, de la que se dispone de mucha documentación que facilita su uso y extensión. Además, resulta inmediato realizar comparaciones de diferentes propuestas de una forma rápida y exacta, y que permite reutilización de código, al poder reutilizar niveles OSI ya existentes para la simulación de nuevas propuestas que solo afecten a determinados niveles de la arquitectura.

##### *5.1.1 Proceso de simulación*

El simulador está escrito en lenguaje de programación C++ y en lenguaje de script OTCL (Object Tool Command Lenguaje). El porqué de dos lenguajes tiene su justificación en la naturaleza de los propios lenguajes. C++ encaja perfectamente en el desarrollo de aplicaciones detalladas, es decir, todo lo que conlleva programación de sistemas a un nivel de detalle alto. La

manipulación a nivel de byte, procesamiento de paquete, implementación de algoritmos, etc., es sencilla y su ejecución muy rápida. El único punto en contra es la necesidad de recompilación asociada a los cambios que se deseen hacer sobre el código. Por su parte, *OTcl* se encarga de la parte de variación de parámetros, o aspectos de simulación. La definición de modelos o topologías para una rápida exploración de los escenarios así como los cambios en los mismos, justifica enormemente el uso de un lenguaje interpretado. De esta forma, el lenguaje C++ se utiliza para la implementación de los diferentes protocolos de red, y el lenguaje OTCL utiliza un lenguaje de script que permite configurar de forma rápida y sencilla un gran número de escenarios de simulación. El simulador utiliza como ficheros de entrada los siguientes ficheros OTCL:

- Fichero de escenario de red: que describe el área y topología de la red.
- Fichero de patrón de tráfico de red: que describe el patrón de tráfico de la WSN, pudiéndose utilizar diversos tipos de tráfico sintético así como trazas reales.
- Fichero de configuración de los nodos: que permite especificar los protocolos de los diversos niveles de la arquitectura y sus parámetros de funcionamiento.

Cabe destacar que el número de ficheros necesarios para estas definiciones no es fijo, puede haber distintos scripts de configuración que realicen distintas tareas y se ejecuten secuencialmente por lo que se crea un núcleo de scripts para tal efecto. Rodeando a este núcleo de scripts se encuentran otros scripts auxiliares ya sea de generación de tráfico, generación de escenarios que posee *NS-2* en su distribución. Se pueden emplear estos scripts para generar automáticamente y de forma sencilla escenarios de simulación específicos.

Estos scripts actúan previamente a la ejecución de la simulación. Una vez el *NS-2* ha simulado el escenario, genera un fichero de traza con todos los datos relativos a la simulación y opcionalmente un fichero de traza para la ejecución de la herramienta visual *NAM*. En ambos casos se tienen las siguientes posibilidades:

- *Fichero de traza del NS*: este fichero tiene un formato específico y definido por eventos, *tags*, y *flags*:

Con el consecuente *parsing* por medio de un script creado por el usuario en *Perl* o *Python*, se pueden obtener los índices de prestaciones a evaluar.

- *Fichero de traza para NAM*: el *NAM (Network Animator)* es la herramienta que permite visualizar de forma gráfica la simulación, pudiendo observar todo el intercambio de paquetes entre nodos y pérdidas producidos durante la simulación. Únicamente hay que ejecutar el *NAM* con entrada este fichero de traza para lanzar la visualización.

### 5.1.2 El simulador NS-2

El simulador *NS-2* es un simulador dirigido por eventos para la evaluación de protocolos de red, para el cual se han desarrollado medios físicos inalámbricos que utilizan modelos

específicos de propagación para evaluar el rango de las transmisiones inalámbricas. Dichos modelos incluyen características como retardos de propagación, efectos de colisiones y detección de la señal portadora entre otras. Por otro lado, se incluyen implementaciones de protocolos de acceso al medio como p.e IEEE 802.11 y 802.15.4 utilizado en redes inalámbricas de sensores, así como se han modelado interfaces de red que incluyen características relacionadas con la modulación y la transmisión inalámbrica de la señal. Además también incorpora diversos protocolos de enrutamiento como p.e DSR, AODC, DSDV y TORA, a parte de los que los investigadores van desarrollando y dejando disponibles a la comunidad.

Un recurso clave del *NS-2* es su extensible librería y su modelo compuesto de componentes. Debido a ello, el uso del *NS-2* requiere experiencia en lenguajes de programación orientados a objetos, análisis y diseño. Esto es porque para tener un conocimiento global del dominio y las abstracciones del modelo de eventos discretos se necesita desarrollar los conceptos en términos de la clase *TclObject* y muchas otras. La clase *TclObject* es la superclase de todos los objetos de la librería *OTcl* ya sean planificadores, componentes de red, temporizadores u objetos relacionados con *NAM*. La clase *NSObject* es la superclase de todos los objetos componentes básicos de red que manejan paquetes, los cuales componen objetos compuestos de red superiores como pueden ser los nodos o los enlaces. Estos componentes básicos a su vez están divididos en las clases Conectores y Clasificadores basándose en el posible número de rutas de salida. Los objetos de red básicos que poseen una única ruta de salida de datos se encuentran bajo la clase Conectores mientras que aquellos objetos que tienen múltiples rutas de salida de datos están bajo la clase Clasificadores.

Una visión parcial de la jerarquía de clases del *NS-2* se puede ver en la figura 5.1:

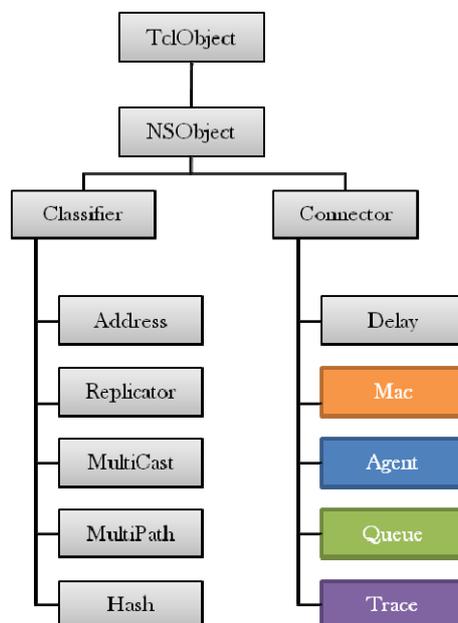


Figura 5.1. Jerarquía de clases parcial de NS-2

A continuación se describirá brevemente cada rama de objetos indicando su correspondiente funcionalidad:

- **Classifier:** La función de un nodo cuando recibe un paquete es examinar los campos del paquete, normalmente su dirección de destino, y en ocasiones, su dirección origen. Posteriormente, debería mapear dichos valores hacia una interfaz de salida de un objeto, que es el siguiente recipiente que debe analizar el paquete. En el *NS-2*, esta tarea la realiza un simple objeto clasificador. Una serie de múltiples objetos clasificador, cada uno analizando una porción del paquete, realizan el forward del paquete a través del nodo. Un nodo en *NS-2* emplea diferentes tipos de clasificadores para diferentes efectos.

El clasificador provee una forma de *matching* entre paquetes apoyándose en criterios lógicos y devuelve una referencia a otro objeto de la simulación basándose en los resultados del *matching*. Cada clasificador contiene una tabla de objetos de la simulación indexada por número de slot. El trabajo de un *clasificador* es determinar el número de slot asociado con un paquete recibido y realizar el forward del paquete hacia el objeto referenciado por ese slot en particular. La clase *C++ Classifier* (*~ns/classifier.h*), cuyo perfil se puede observar en la figura 5.2, es la clase desde la cual se derivan otras clases clasificadoras.

```
class Classifier : public NsObject {
public:
    ~Classifier();
    void recv(Packet*, Handler* h = 0);
protected:
    Classifier();
    void install(int slot, NsObject*);
    void clear(int slot);
    virtual int command(int argc, const char*const* argv);
    virtual int classify(Packet *const) = 0;
    void alloc(int);
    NsObject** slot_;      /* table that maps slot number to a NsObject */
    int nslot_;
    int maxslot_;
};
```

Figura 5.2. Clase *Classifier*

Cuando un clasificador recibe un paquete mediante el método *recv()*, lo maneja mediante el método *classify()*. Este método está definido de manera distinta en cada tipo de clasificador derivado de la clase base. El formato usual del método *classify()* es determinar y devolver el índice de slot en la tabla de slots. Si el índice es válido, y apunta hacia un objeto *TcObject* válido, el clasificador manejará el paquete con dicho objeto usando el método *recv()* del propio objeto. Si por el contrario el índice no es válido, el clasificador invocará una instancia del procedimiento llamado *no-slot{}* para intentar diseminar la tabla correctamente. Aun así, el propio procedimiento *Classifier::no-slot{}* imprime un mensaje de error por pantalla y termina la ejecución.

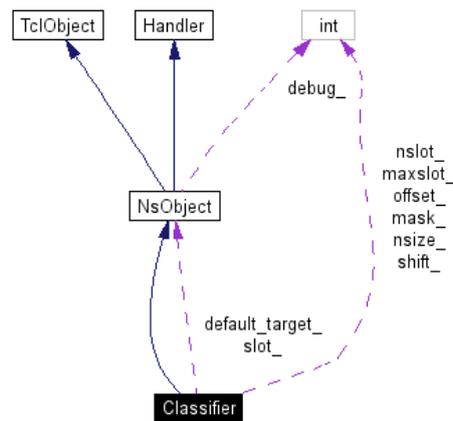


Figura 5.3. Diagrama de colaboración de la clase *Classifier*

Como se observa en la figura 5.3, las clases que derivan de la clase *Classifier* son:

- **Address:** El clasificador de direcciones se emplea para soportar el reenvío unicast de los paquetes. Éste aplica un cambio inteligente de bits (*bitwise shift*) y operaciones de enmascaramiento a la dirección de destino del paquete para producir el número de slot. El diagrama de relación se puede observar en la figura 5.4.

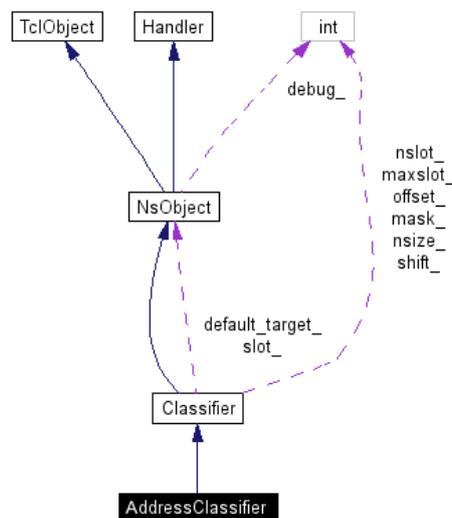


Figura 5.4. Diagrama de colaboración de la clase *Address*

- **MultiCast:** El clasificador *multicast* clasifica los paquetes en función tanto de su dirección de origen como destino (de grupo). Mantiene una tabla (hash enlazada) que mapea pares de grupo fuente a números de slot. Cuando un paquete que contiene un grupo-fuente desconocido llega al clasificador, éste invoca un procedimiento `OTcl Node::new-group{}` que añade la entrada a la tabla. Este procedimiento `OTcl` emplea el método `set-hash` para añadir la nueva 3-tupla (*source, group, slot*) a la tabla del clasificador. Su diagrama de colaboración se puede observar en la figura 5.5.

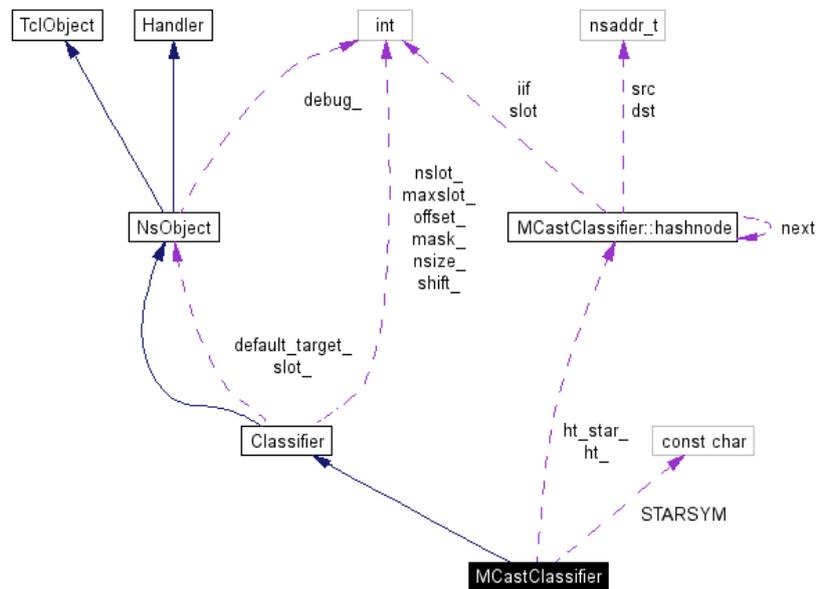


Figura 5.5. Diagrama de colaboración de la clase *MultiCast*

- **MutiPath:** Este objeto está diseñado para soportar el envío *multipath* cuando los costes son iguales, esto es, cuando los nodos poseen rutas de igual coste hacia el mismo destino y quieren emplear todas esas rutas simultáneamente. Este objeto no analiza ningún campo del paquete. Con cada paquete correcto, simplemente devuelve el siguiente slot relleno aplicando la planificación *Round Robin*. Su diagrama de colaboración se puede observar en la figura 5.6.

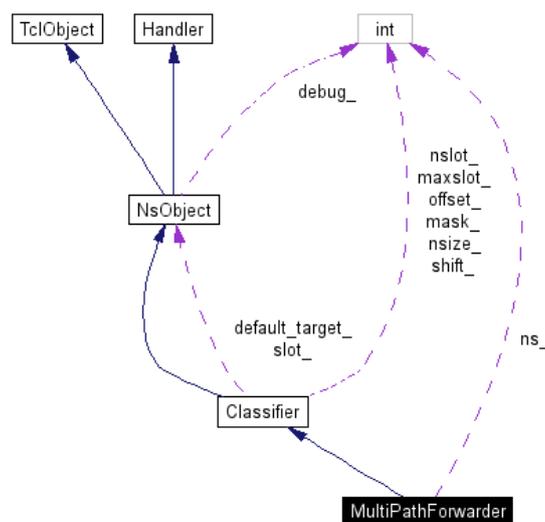


Figura 5.6. Diagrama de colaboración de la clase *MultiPath*

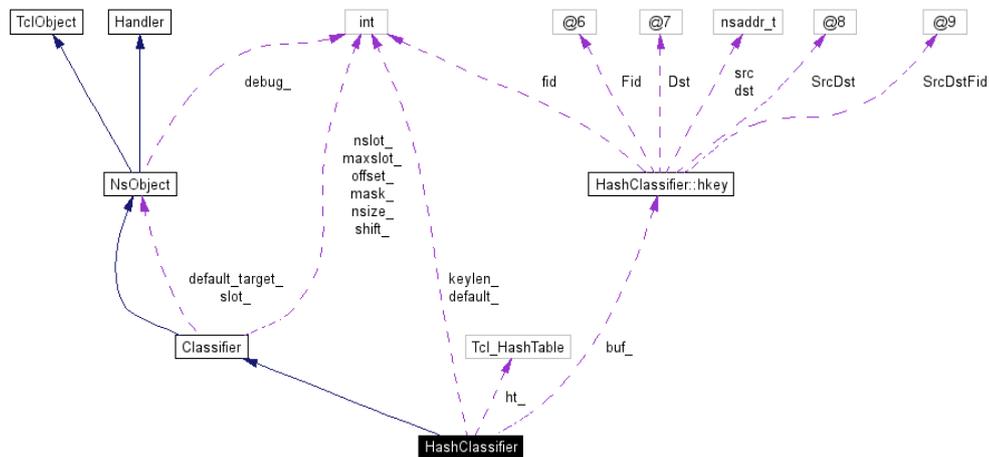


Figura 5.7. Diagrama de colaboración de la clase *Hash*

- **Hash:** Este objeto se usa para clasificar un paquete como miembro de un flujo particular. Como su nombre indica, los clasificadores hash emplean una tabla hash interna para asignar paquetes con flujos. Estos objetos se usan donde se requiere información de nivel-flujo, como por ejemplo en ciertas disciplinas de *queuing flow-level*. Su diagrama de colaboración se puede observar en la figura 5.7.
- **Replicator:** es diferente a cualquier otro clasificador descrito anteriormente por el mero hecho de que no emplea la función *classify*. En vez de eso, simplemente usa el clasificador como tabla de *n* slots; sobrecarga el método *recv()* para producir *n* copias de un paquete, que son entregadas a todos los objetos referenciados en la tabla.
- **Connector:** Los conectores, a diferencia de los clasificadores, únicamente generan datos para un destino; o se entrega el paquete al *target\_* del vecino o se descarta enviándolo a *drop-target*. Su diagrama de relación se puede observar en la figura 5.8. Un conector recibe un paquete, realiza algunas funciones y entrega dicho paquete a su vecino o lo elimina. Existen diferentes tipos de conectores en *NS-2*, cada uno de los cuales realiza diferentes funciones:
  - ✓ *DynaLink*, objeto que actúa como puerta de entrada del tráfico dependiendo de si el enlace está activo o inactivo.
  - ✓ *DelayLink*, objeto que modela la latencia del enlace y las características de ancho de banda.
  - ✓ *Queues*, modela los buffers de salida adjuntos a un enlace en un enrutador real de una red.

- ✓ *TTLChecker*, que decreenta el *TTL* de cada paquete que recibe.

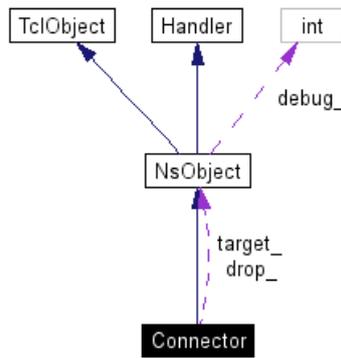


Figura 5.8. Diagrama de colaboración de la clase *Connector*

Derivando de esta clase base tenemos:

- **Delay:** La clase *LinkDelay* deriva de la clase base *Connector*. La latencia representa el tiempo requerido por un paquete para atravesar el enlace. Una forma especial de este objeto, el *dynamic link*, también captura la posibilidad de fallo en el enlace. La cantidad de tiempo requerida por un paquete para atravesar el enlace se define como  $s/b + d$ , donde  $s$  es el tamaño del paquete (registrado en su cabecera),  $b$  es la velocidad del enlace medida en *bits/sec* y  $d$  es el retardo del enlace en segundos.

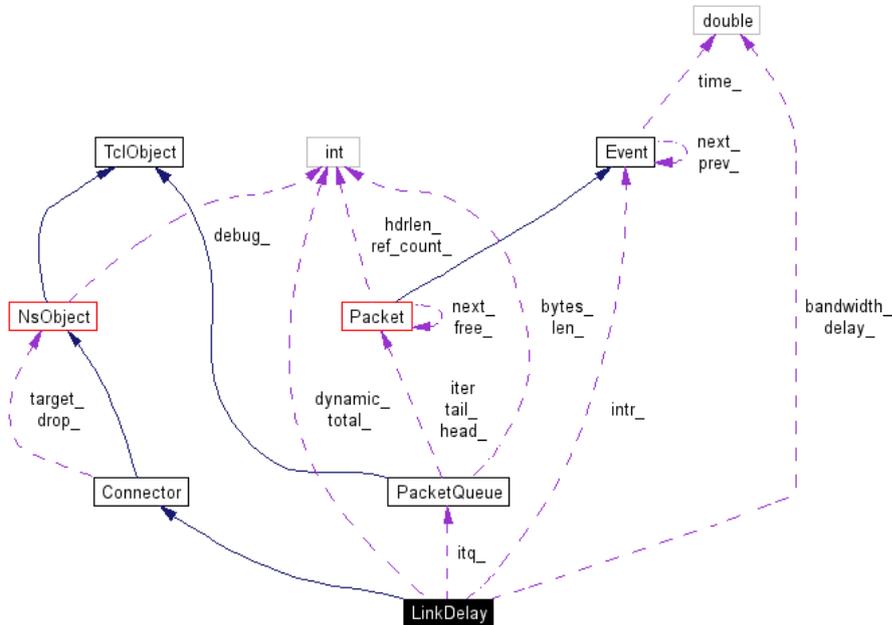


Figura 5.9. Diagrama de colaboración de la clase *LinkDelay*

En la *figura 5.9* se aprecia como la clase *LinkDelay* hace uso de la clase Paquete, Cola y Evento como influencias para plasmar la latencia del enlace. La implementación de *delay links* está íntimamente relacionada con los procedimientos bloqueantes en *NS-2*. En la función *recv()* de la clase, se sobrecarga la versión de la clase base para controlar la variable *dynamic\_*, la cual

determina cuando en enlace es dinámico o no (es propenso a fallar o recuperarse en ciertos instantes de tiempo). El comportamiento interno del enlace es distinto en cada caso.

- **Mac** (figura 91): El objeto *Mac* simula los protocolos de acceso al medio que son necesarios en un medio compartido como el medio *wireless* o *LAN*. Debido a que los mecanismos de envío y recepción están perfectamente acoplados en la mayoría de tipos de las capas *MAC*, es esencial que el objeto *Mac* sea *duplex*.

En el lado del envío, el objeto *Mac* es el responsable de añadir la cabecera *MAC* y transmitir el paquete por el canal. En el lado de la recepción, el objeto *Mac* asincrónicamente recibe paquetes desde el clasificador de la capa física. Después del procesamiento realizado por el protocolo *MAC*, pasa el paquete a la capa de enlace.

Además de la propia clase *Mac*, existen parámetros asociados con el procesamiento *MAC*:

- *Mac State*: La clase *C++* contiene suficiente estado interno como para simular el particular protocolo *MAC*. También exporta los siguientes parámetros de configuración *OTcl*:
  - o *bandwidth\_*, ratio de modulación del *MAC*
  - o *hlen\_*, bytes adicionales añadidos al paquete para la cabecera *MAC*
  - o *label\_*, la propia dirección *MAC*
- *Mac Methods*: La clase *Mac* añade varios métodos *Tcl* para configuración, en especial, enlace con otros objetos de la simulación.
  - o *channel*, especifica el canal para la transmisión
  - o *classifier*, el clasificador que entrega los paquetes para que los reciba la *MAC*
  - o *maclist*, una lista de enlace de interfaces *MAC* en el mismo nodo

En *C++*, la clase *Mac* (figura 5.10) deriva de la clase *Connector* como ya se comentó. Cuando el método *recv()* obtiene un paquete, identifica la dirección del paquete basándose en la presencia de un manejador de *callback*. Si existe tal manejador, el paquete es saliente, si no, es entrante. Cuando un objeto *Mac* recibe un paquete vía su método *recv()*, comprueba si el paquete es saliente o entrante. Para un paquete declarado como saliente, se asume que la capa de enlace del emisor ha obtenido la dirección *MAC* del destinatario y la ha rellenado en el campo *macDA\_* de la cabecera *MAC*, *hdr\_mac*. El objeto *Mac* rellena el resto de la cabecera *MAC* con la dirección del *source* y el tipo de *frame*. Es entonces cuando pasa el paquete a su método *send()*, el cual realiza el *carry* del medio mediante el protocolo de acceso. Para el objeto básico *Mac*, el método *send* llama a *txtime()* para computar el tiempo de transmisión y luego invoca *Channel::send* para transmitir el paquete. Finalmente, se auto-planifica de nuevo para cuando el tiempo de transmisión haya transcurrido. Para un paquete entrante, el objeto *Mac* realiza su protocolo de procesamiento y pasa el paquete a la capa de enlace.

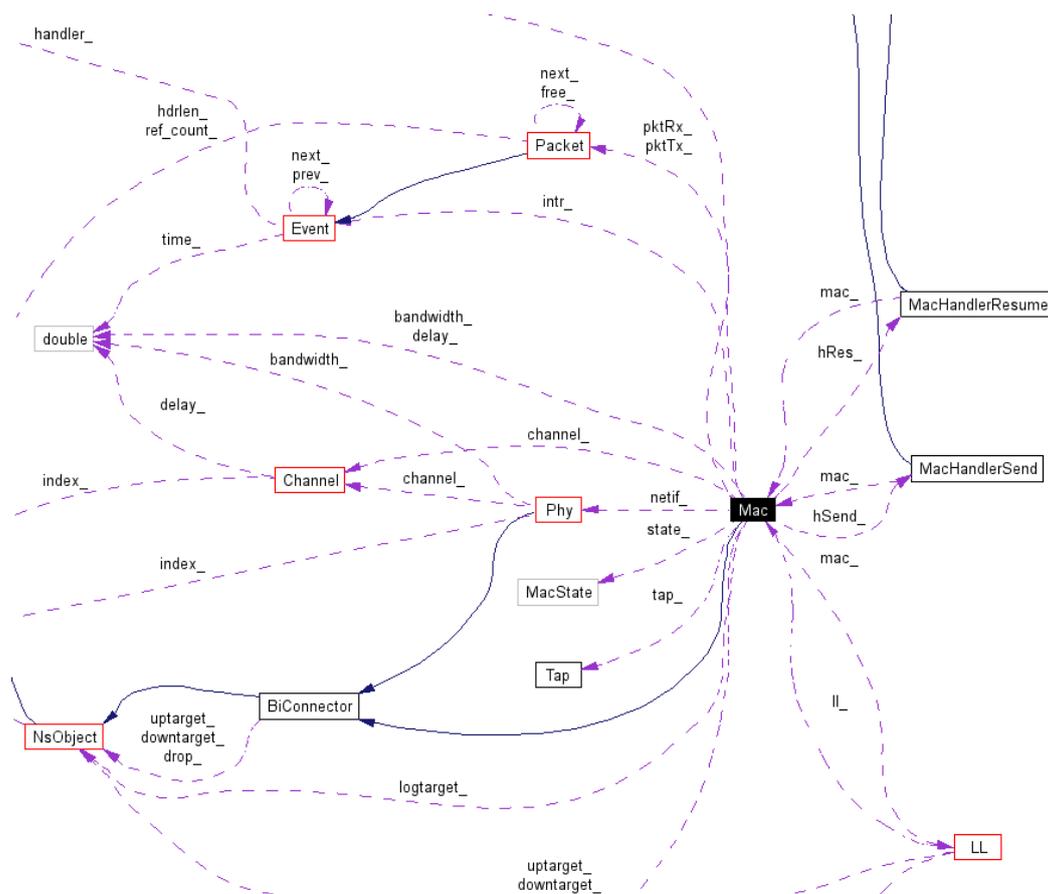


Figura 5.10. Diagrama de colaboración de la clase *Mac*

- **Queue:** Las colas representan lugares donde los paquetes pueden situarse o descartarse. Asociado a este concepto está el tratamiento de los buffers, que se refiere a la disciplina en particular que regula la ocupación de una cola en particular. La clase *Queue* provee una clase base usada para un tipo en particular de clases de colas (derivadas), así como funciones de *call-back* para implementar bloqueos. Actualmente, se soporta *drop-fail queueing (FIFO)*, *CBQ*, que incluye planificaciones *RR* y por prioridad, *Stochastic Fair Queueing (SFQ)* y *Deficit Round-Robin*, entre otros.

Como se puede observar en la figura 5.11 los elementos principales de los que hace uso esta clase son la propia clase *Paquete* y estructuras de datos asociadas el manejo de la propia cola, como el tipo *QueueElem* o el manejador de la cola *QueueHandler*.

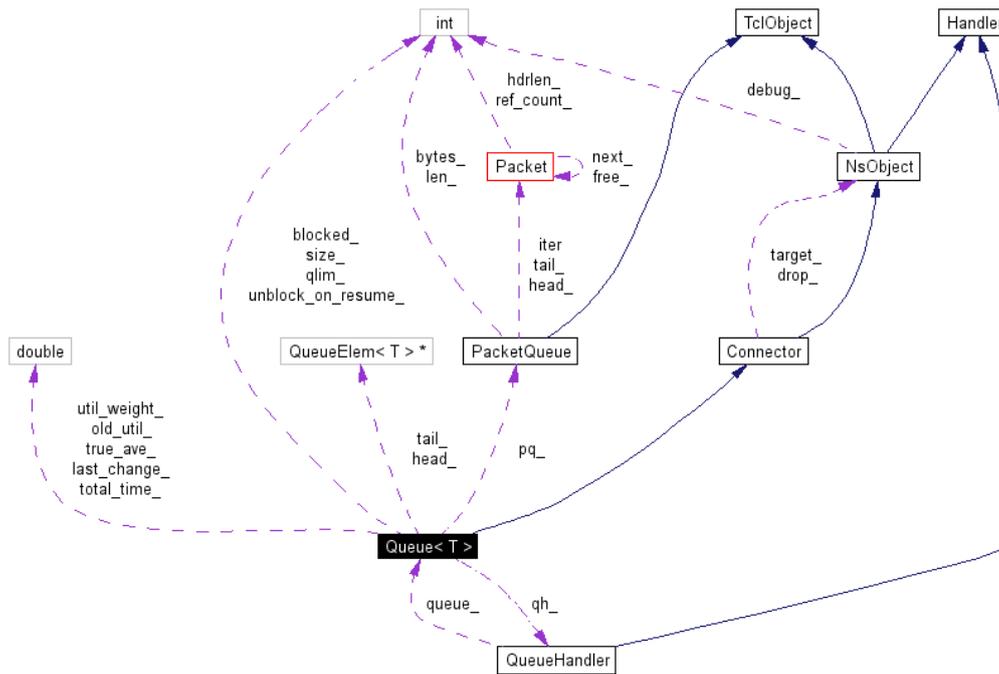


Figura 5.11. Diagrama de colaboración de la clase *Queue*

- **Trace:** La clase *Trace* se utiliza para implementar las clases *OTcl Trace/Hop*, *Trace/Enque*, *Trace/Deque* y *Trace/Drop*. Cabe destacar que el método `Trace::format()` define el formato dl fichero de traza producido por esta clase. Su diagrama de relación puede observarse en la figura 5.12.

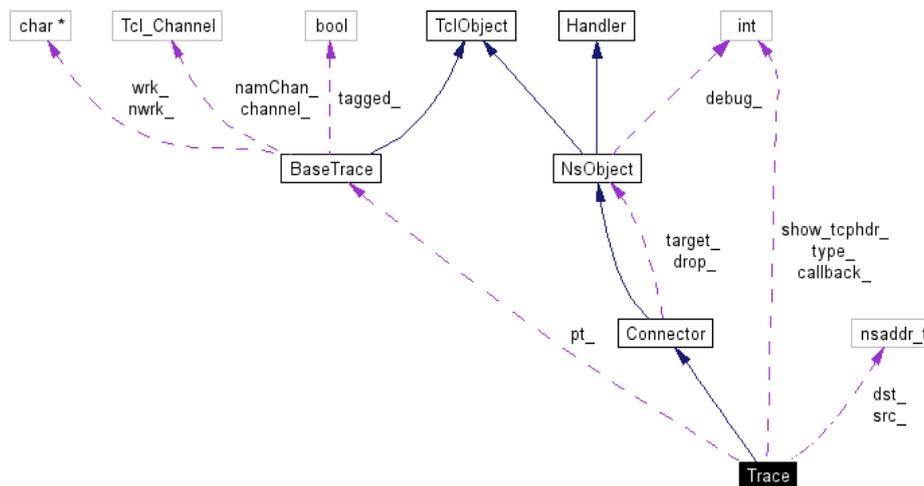


Figura 5.12. Diagrama de colaboración de la clase *Trace*

Cabe destacar el uso de la clase *BaseTrace* que enlaza con parámetros y con la propia clase *TclObject* y el empleo típico tanto del tipo *addr* como de variables asociadas a los *NsObject* de los cuales se estén realizando trazas.

En lo que al modelo de nodo se refiere, el procedimiento de creación de una instancia de un nodo construye un nodo a partir de clasificadores más simples. El nodo por sí mismo es una clase independiente en *OTcl*. Sin embargo, la mayoría de los componentes de un nodo son *TclObjects*. La típica estructura de un nodo *unicast* se muestra en la figura 5.13.

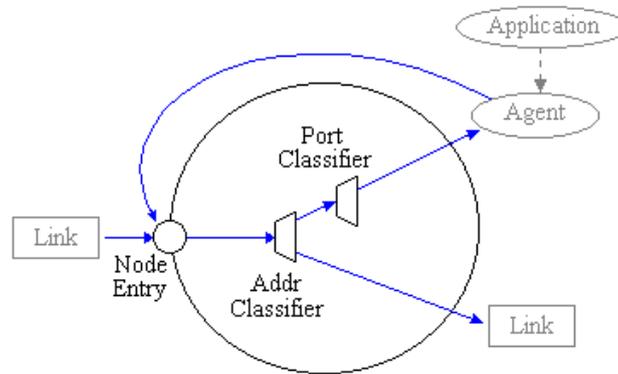


Figura 5.13. Estructura de un nodo *unicast*

Esta simple estructura consta de dos *TclObjects*:

- Un *Address Classifier* o Clasificador de Direcciones, *classifier\_*
- Un *Port Classifier* o Clasificador de Puertos, *dmux\_*

La función de estos clasificadores es distribuir los paquetes entrantes a los agentes o enlaces correspondientes. Por defecto, los nodos en *NS-2* se construyen para simulaciones *unicast*. Para activar las simulaciones *multicast*, la simulación debe ser creada con la opción *-multicast on*. La estructura de uno de estos nodos *multicast* se muestra en la figura 5.14.

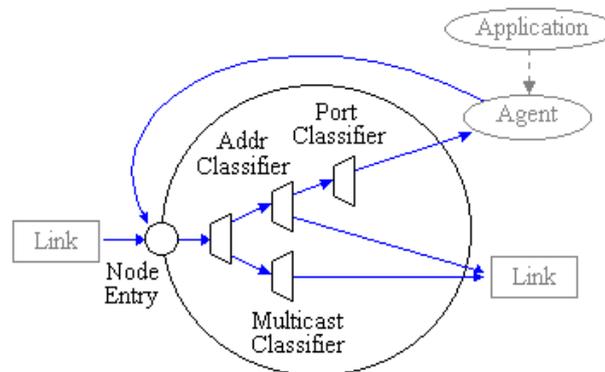


Figura 5.14. Estructura de un nodo *multicast*

Cuando una simulación emplea *multicast routing*, el bit de mayor peso de la dirección indica cuando es una dirección *multicast* o es una dirección *unicast*. Si el bit es 0, la dirección representa una dirección *unicast*, sino, se trata de una dirección *multicast*.

Como se puede observar en ambas figuras, el objeto *Link* (enlace) derivado de la clase *LL* también es un objeto de gran importancia en *NS-2*. Estos enlaces son el segundo aspecto en la creación de la topología, que actúan como conectores de los nodos y completan la formación de la topología. *NS-2* soporta tanto la creación de enlaces punto por punto simples como la creación de enlaces bidireccionales, emulación de *multi-access LAN* u otros tipos de conexiones *wireless* o media.

Tal y como se ha visto, los nodos son una composición de clasificadores. Un enlace sería una secuencia de conectores. La clase *Link* es una clase independiente en *OTcl*, que provee una serie de primitivas simples. La clase *SimpleLink* ofrece la oportunidad de conectar dos nodos con un enlace punto por punto mediante el procedimiento instanciado *simplex-link{}*, formando de esta forma un enlace unidireccional entre ambos nodos. El enlace se encuentra en la clase *SimpleLink*. Para el caso de enlaces bidireccionales, el procedimiento adecuado sería *duplex-link{}*, que construiría un enlace bidireccional entre nodos mediante dos enlaces simples.

Este objeto capa de enlace es el responsable de simular los protocolos de enlace de datos. Muchos protocolos pueden ser implementados en esta capa como fragmentación de paquetes y reensamblaje o protocolos de enlaces fiables.

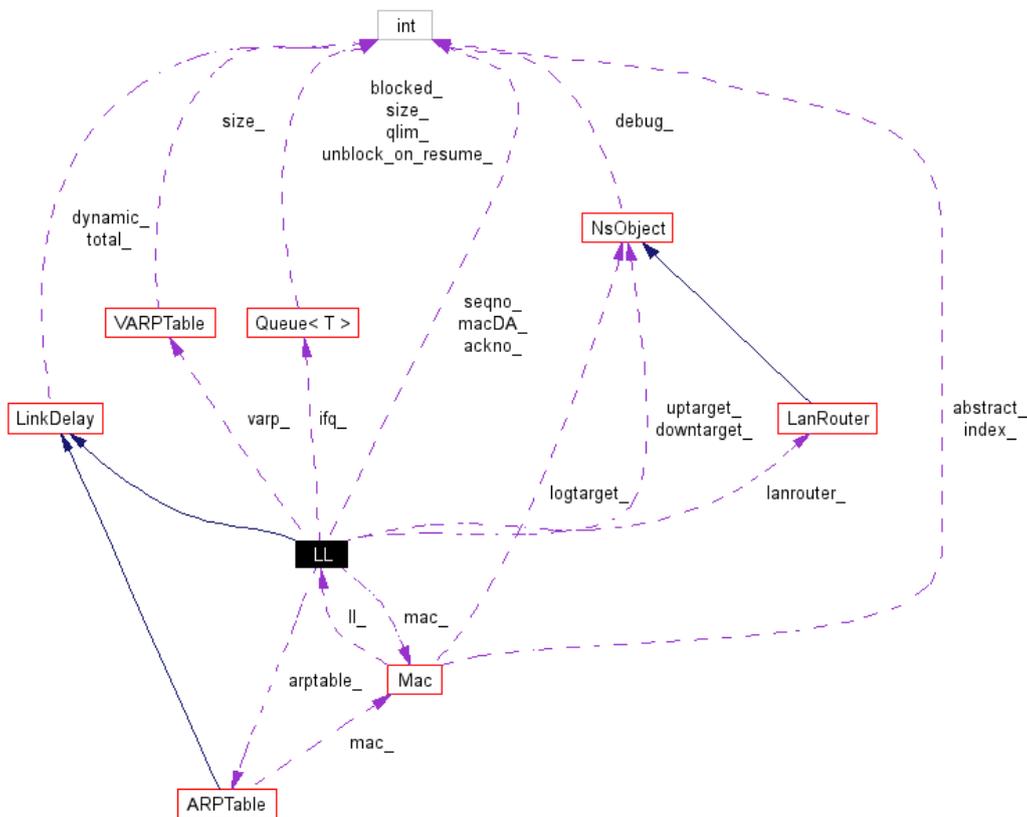


Figura 5.15. Diagrama de colaboración de la clase *LL* (enlace de datos)

Otra función importante de la capa de enlace es configurar la dirección *MAC* de destino en la cabecera *MAC* del paquete (figura 5.15).

Por tanto el flujo de datos/eventos en *NS-2* se desarrolla como se ha explicado según vemos en la figura 5.16, donde podemos observar un ejemplo de tal flujo, dónde un nodo con tráfico de *FTP* envía a otro nodo información.

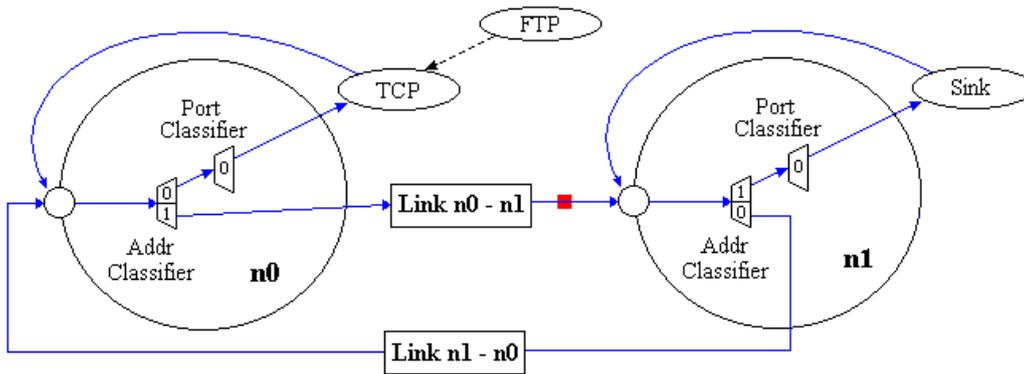


Figura 5.16. Ejemplo de flujo

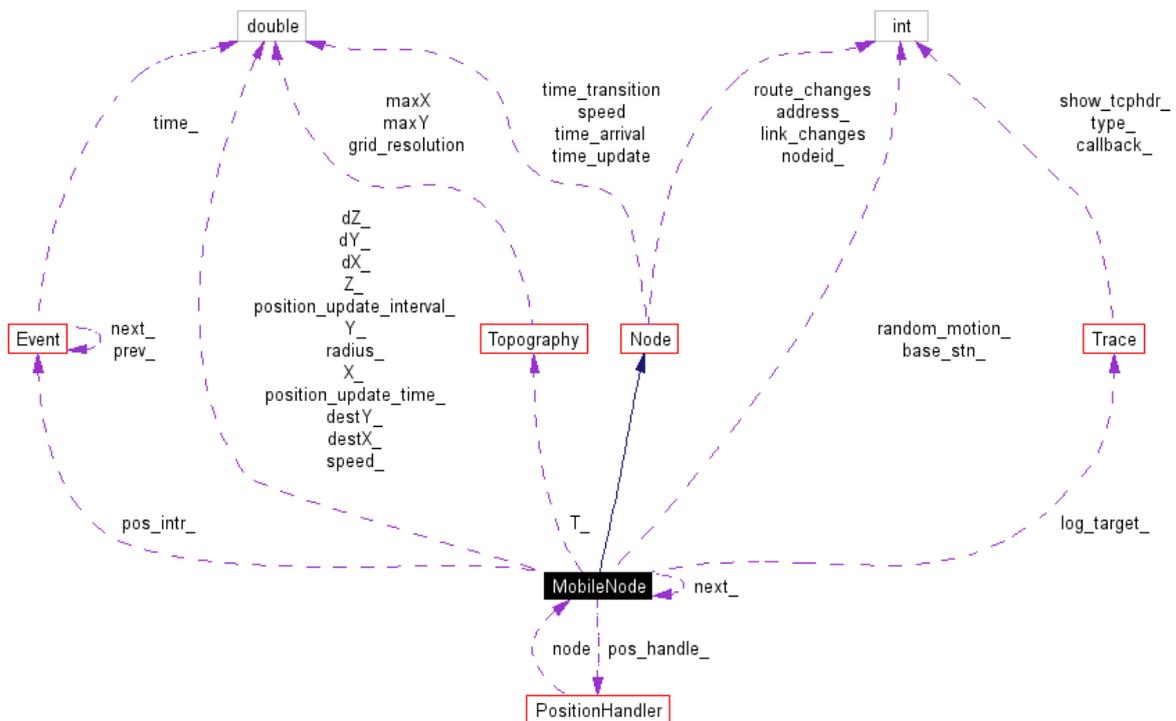


Figura 5.17. Diagrama de colaboración de la clase *MobileNode*

El modelo *wireless* esencialmente está compuesto en su núcleo por *MobileNode*, que soporta características adicionales que permiten simulaciones de redes *ad-hoc multi-hop*, *wireless LANs*, etc. El objeto *MobileNode* es un objeto fraccionado. Como se puede observar en la

figura 5.17, la clase *C++ MobileNode* deriva de la clase padre *Node* y por ello un *MobileNode* es el objeto básico *Node* más un conjunto de funcionalidades *wireless* y de movilidad que le otorgan habilidades como moverse a través de una topología determinada, recibir y transmitir señales por un canal *wireless*, etc. La mayor diferencia entre los nodos normales y un *MobileNode* es que los *MobileNode* no están conectados mediante enlaces los unos con los otros.

A la hora de configurar los parámetros del *MobileNode* para el entorno de simulación *wireless*, la nueva API de NS-2 otorga más flexibilidad para configurar las opciones del nodo. Parámetros importantes como el canal, la topología, el modelo de propagación, opción de traza, etc. Se manipulan sencillamente mediante la opción correcta. Posteriormente habría que crear los *MobileNode*. El *MobileNode* tiene una estructura como se muestra en la figura 5.18.

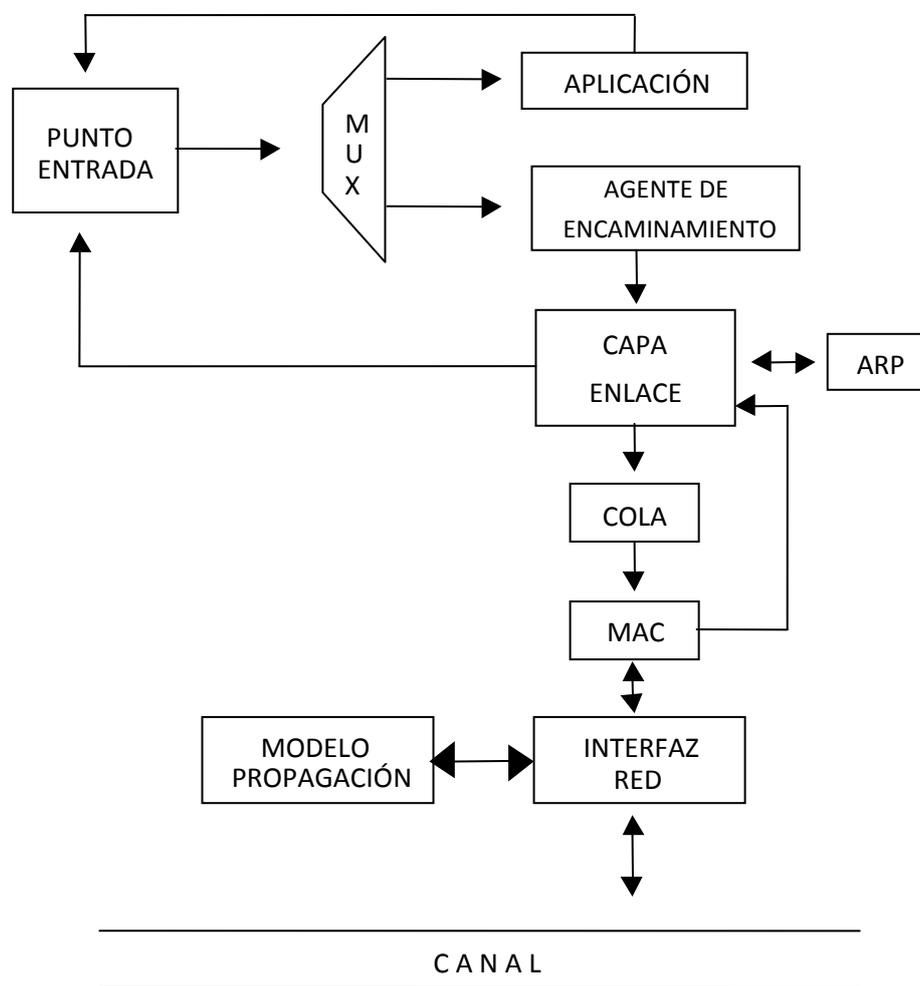


Figura 5.18. Diagrama de bloques *MobileNode* bajo la extensión *wireless*

Los componentes pertenecientes a la pila de red *wireless* son los siguientes:

- **Capa enlace (Link Layer):** La *LL* usada por el *MobileNode* es la misma que se describió en párrafos anteriores. La única diferencia que existe es que cada *MobileNode* posee un modulo *ARP* conectado a él mismo que resuelve todas las direcciones *IP* realizando la conversión a direcciones *HW* (*MAC*).
- **ARP:** El modulo protocolo de resolución de direcciones resuelve las correspondientes consultas desde la *LL*.
- **Cola:** Esta interfaz implementa una cola de prioridad la cual da prioridad máxima a los paquetes del protocolo, insertándolos en la cabeza de la cola.
- **MAC:** Históricamente, el *NS-2* ha empleado la implementación de *802.11 DCF* como función de coordinación distribuida. En la versión actual del simulador, *NS-2 2.33*, diversas implementaciones bajo *802.11* y *802.15.4* están disponibles.
- **Interfaz red:** La capa de interfaz de red sirve como interfaz *HW* la cual es usada por el *MobileNode* para acceder al canal. La interfaz *wireless* para medios compartidos está implementada como clase *Phy/WirelessPhy*. Esta interfaz está sujeta a colisiones y el modelo de propagación de radio recibe los paquetes transmitidos por otras interfaces en el canal. La interfaz plasma cada paquete transmitido con meta-datos relacionados con la interfaz de transmisión tales como longitud de onda, energía de transmisión, etc.
- **Modelo propagación radio:** Se usa la atenuación *Friss-Space* ( $1/r^2$ ) para distancias cercanas y una aproximación del modelo de *Two ray Ground* ( $1/r^4$ ) para distancias largas. La aproximación asume reflexión especular en un plano horizontal.
- **Antenna:** Los *MobileNodes* emplean una antena omnidireccional con unidades de ganancia.

En general, una vez analizada la estructura de bloques de las estaciones inalámbricas utilizadas por el simulador, destacar que de los diferentes niveles, el nivel de aplicación esta implementado en OTCL mientras que el resto de niveles están implementados en C++. El nivel de aplicación utiliza el patrón de tráfico para generar paquetes de datos que son enviados al agente de encaminamiento, el cual implementa las funciones de red y transporte. Posteriormente el agente de encaminamiento envía el paquete al nivel de acceso al medio siendo procesado previamente por el nivel de enlace de datos. Por último el paquete es enviado al interfaz de red donde se calcula la energía de transmisión necesaria para enviar el paquete al canal, el cual envía el paquete a las estaciones, las cuales utilizarán modelos de propagación de la señal para evaluar la correcta o incorrecta recepción del paquete y evaluar si debe ser enviado al nivel superior o descartado.

Como ya se ha comentado previamente, *NS-2* posee implementados dos protocolos de la capa *MAC* para redes *wireless* que son la basada en contención *802.11*, *802.15.4* y la basada en diferentes slots de transmisión mediante *TDMA*. Debido a que en nuestro caso un protocolo

MAC orientado a slots de transmisión es más adecuado y se emplea activamente para el desarrollo de **EDETA**, vamos a proceder a detallar algunos aspectos:

- *Preamble based TDMA MAC protocol*: Actualmente, NS-2 soporta un protocolo MAC basado en TDMA con preámbulo. Con este protocolo, un *frame* de TDMA contiene un preámbulo además de los slots pertenecientes a la transmisión de datos. Dentro de la zona de preámbulo, cada nodo tiene un *subslot* dedicado y lo emplea para realizar un *broadcast* del *id* del nodo destino del paquete saliente. El resto de nodos escuchan durante el preámbulo y registran los slots de tiempo para recibir paquetes. De la misma forma que otros protocolos TDMA como GSM, cada nodo posee un slot de transmisión de datos dedicado para enviar paquetes.

Al principio del *frame*, cada nodo escribe el *id* del destinatario en su *subslot* del preámbulo si tiene algún paquete que enviar. A continuación de la fase de preámbulo, cada nodo envía el paquete en su slot de transmisión de datos y comprueba el preámbulo para determinar si existe algún paquete a recibir en otros slots.

Para evitar un consumo innecesario, cada nodo enciende y apaga su radio explícitamente invocando a la API del nodo *set\_node\_sleep()*. La radio únicamente necesita estar encendida en dos casos: en la fase de preámbulo (ocupa un slot de tiempo) y cuando existe un paquete para recibir o enviar.

En cuanto a los modelos de propagación, estos modelos se emplean para predecir la potencia de la señal de recepción de cada paquete. En la capa física de cada nodo *wireless*, existe un *threshold* de recepción editable mediante *Phy/WirelessPhy set RXThresh\_ <value>*. Este umbral se relaciona con el radio de comunicación ya que el umbral configura la sensibilidad de recepción de la señal de comunicación. Cuando se recibe un paquete, si la potencia de su señal está por debajo de este umbral, se marca como erróneo y es eliminado por la capa MAC.

Hay tres modelos de propagación en NS-2, que son los siguientes:

- *Free Space Model*: Este modelo asume la condición ideal de propagación de que solo existe una única ruta en la línea de visión entre el transmisor y el receptor. Se emplea la ecuación (5.1) para calcular la potencia de la señal recibida en espacio abierto a una distancia *d* desde el transmisor:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (5.1)$$

Donde  $P_r$  es la potencia de la señal transmitida,  $G_t$  y  $G_r$  es la ganancia de la antena del transmisor y el receptor respectivamente,  $L$  ( $L \geq 1$ ) es la pérdida del sistema y  $\lambda$  es la longitud de onda. Es común seleccionar  $G_t = G_r = 1$  y  $L = 1$  en las simulaciones con NS-2.

El *Free Space Model* básicamente representa el radio de comunicación como un círculo alrededor del transmisor como se puede observar en la figura 5.19. Si el receptor se encuentra dentro de dicho círculo, recibe todos los paquetes. En caso contrario los pierde.

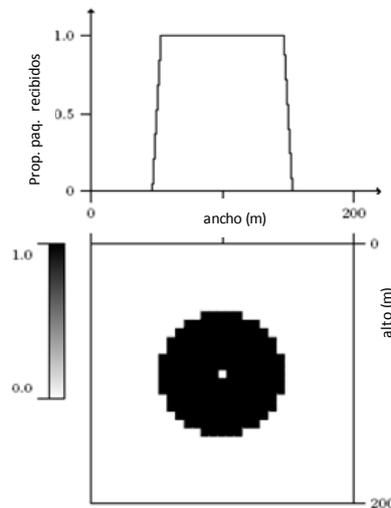


Figura 5.19. Representación gráfica *Free Space Model*

- *Two-ray Ground Reflection Model*: Debido a que rara vez se da una línea directa de visión entre dos *MobileNode*, este modelo asume, además de la propagación por ruta directa, la ruta mediante reflexión con el terreno. La potencia recibida a una distancia  $d$  se predice mediante la ecuación (5.2):

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (5.2)$$

Donde  $h_t$  y  $h_r$  son las alturas de las antenas de transmisión y recepción respectivamente. Se deja constancia de que para mantener la consistencia con el *Free Space Model*,  $L$  se añade en la ecuación para este modelo.

En varios trabajos se demuestra que este modelo proporciona una predicción más exacta en largas distancias que el *Free Space Model*. En cuanto a la representación gráfica del modelo, sería igual que la representada en el *Free Space Model*.

- *Shadowing Model*: El *Free Space Model* y el *Two-ray Ground Reflection Model* predicen la potencia de recepción como una función determinista de la distancia. Ambos representan el radio de comunicación como un círculo ideal cuando en realidad, la potencia de recepción a cierta distancia es una variable aleatoria debido a efectos de propagación *multipath*; lo que se conoce como efecto de *fading*.

El *Shadowing Model* es un modelo más general y ampliamente usado que consta de dos partes: una parte que modela las pérdidas en la ruta de comunicación, que también predice la potencia media recibida a distancia  $d$ , denotada por  $P_r(d)$ . Emplea una cercanía de la distancia

$d_0$  como referencia.  $P_r(d)$  se computa en relación con  $P_r(d_0)$  como se indica en la expresión (5.3):

$$\frac{P_r(d_0)}{P_r(d)} = \left(\frac{d}{d_0}\right)^\beta \quad (5.3)$$

$\beta$  es el denominado exponente de pérdida de ruta y normalmente se determina mediante medidas del propio campo de actuación. Valores más grandes corresponden a mayores obstrucciones y, por lo tanto, mayor rapidez en la disminución de la potencia media recibida conforme la distancia se vuelve mayor.

La pérdida de ruta se mide normalmente en  $dB$ , según (5.4):

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) \quad (5.4)$$

La otra parte que modela la variación de la potencia de recepción a distancia  $d$ , reflejada mediante una variable aleatoria log-normal la cual si se modela como una distribución *Gaussiana* se debe medir en  $dB$ . El *Shadowing Model* general se representa mediante (5.5):

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \quad (5.5)$$

Donde  $X_{dB}$  es una variable aleatoria *Gaussiana* con media cero y desviación estándar  $\sigma_{dB}$ .  $\sigma_{dB}$  es la llamada desviación de encubrimiento o *shadowing* y también se obtiene mediante media de campo. El *Shadowing Model* extiende la idea del modelo circular de comunicación a un modelo estadístico más rico; los nodos pueden comunicarse únicamente con los bordes de su radio de comunicación de forma probabilista. La representación gráfica de este modelo puede observarse en la figura 5.20.

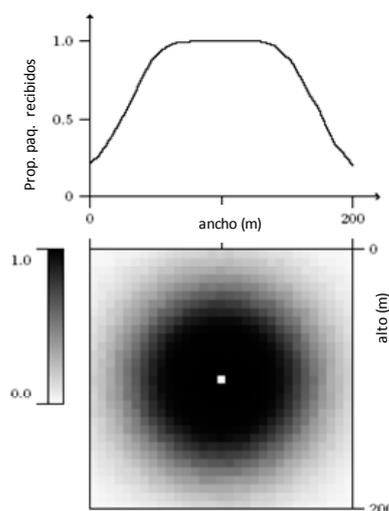


Figura 5.20. Representación gráfica *Shadowing Model*

### 5.1.3 Modelo de energía

El autor de la presente tesis ha extendido las funciones del simulador NS-2 para incorporar un modelo de consumo energético de los interfaces de red inalámbricos utilizados en las redes inalámbricas de sensores. Este apartado describe el modelo energético empleado para evaluar los mecanismos propuestos en la presente tesis.

Hay muchas propuestas teóricas para redes inalámbricas de sensores donde se asume un conjunto homogéneo de nodos [HEINZEL00, YIN08, FAPOJUW09, BOUKER09]. Algunas asunciones de estos protocolos son:

- todos los nodos son homogéneos
- cada nodo tiene la misma cantidad de energía disponible
- El modelo de radio asume un canal simétrico donde el coste energético para una transmisión desde A a B es el mismo que para una transmisión de B a A.

El objetivo de esta nueva arquitectura ha sido maximizar los beneficios de los protocolos multi-hop sin sacrificar la vida de los nodos y manteniendo un coste bajo de la red. Para conseguir nuestra aproximación, se proponen estas asunciones realistas:

- Posibilidad de nodos heterogéneos, dado que especializar una fracción de los nodos puede aumentar de forma muy significativa la vida de la red en conjunto.
- Aplicar un modelo de radio asimétrico donde el coste energético para alcanzar un nodo A desde B pueda ser diferente que el de alcanzar B desde A.
- Posibilidad de usar fuentes de energía suplementarias en algunos nodos.

Para modelar estas consideraciones, la ecuación de Friis (5.6) es el modelo matemático básico para predecir las comunicaciones punto a punto en entornos abiertos.

$$P_R = P_T \frac{G_T G_R \lambda^2}{(4\pi)^2 d^n} \quad (5.6)$$

Donde  $P_R$  es la potencia disponible en la antena receptora,  $P_T$  la potencia proporcionada a la antena transmisora,  $G_R$  la ganancia en la antena receptora,  $G_T$  la ganancia en la antena transmisora,  $\lambda$  la longitud de onda, donde  $\lambda=c/f$ , siendo 'c' la velocidad de la luz y 'f' la frecuencia, 'd' la distancia y  $n=2$ .

#### 5.1.3.1 Modelo de energía en términos del modo de funcionamiento

Este modelo de energía detalla el consumo energético del dispositivo en función del modo de funcionamiento del nodo, es decir, de si transmite, recibe o se encuentra inactivo. Esto es de gran importancia ya que si una de estas dos operaciones incurre en un mayor consumo, el

realizar durante un largo periodo de tiempo y sucesivas veces dichas operaciones, provocará que se dispare el gasto energético.

Además de transmitir y recibir, el nodo o dispositivo estará un determinado tiempo inactivo o en *idle*. Este modo de funcionamiento es de menor consumo que el de transmisión o recepción pero no es el de menor gasto energético. El modo de funcionamiento de bajo consumo, en adelante *Low-Energy (LE)*, es el modo que queremos maximizar en términos de tiempo, en aras de minimizar el consumo de energía del dispositivo. Como ya se detalló anteriormente, este es el objetivo de EDETA, aprovechar el máximo tiempo posible entre envíos de datos para que el nodo realice la transición a *LE* y de esta forma minimizar el consumo.

La expresión (5.7) constituye una primera aproximación al modelo de consumo de energía genérico o disipación en Julios a la hora de transmitir un paquete  $p$ :

$$E(p) = I \times V \times t_p \quad (5.7)$$

siendo  $I$  la corriente en Amperios,  $V$  el voltaje utilizado en Voltios,  $t_p$  el tiempo necesario para transmitir el paquete en segundos.

Dependiendo de si se está transmitiendo o recibiendo el paquete, el modo de funcionamiento cambia y por lo tanto la energía también, que actúa en función del consumo del dispositivo en el modo que se esté empleando. Esto se puede apreciar en las expresiones 5.8, 5.9, 5.10 y 5.11.

$$E_{tx}(p) = 17.04mA * v * t_p \quad (5.8)$$

$$E_{rx}(p) = 19.6mA * v * t_p \quad (5.9)$$

$$E_{idle}(t) = 396\mu A * v * t_p \quad (5.10)$$

$$E_{shut}(t) = (80\rho A * v * t_p) + (396\mu A * v * 970\mu s * n) \quad (5.11)$$

donde  $E_{tx}$ ,  $E_{rx}$ ,  $E_{idle}$  y  $E_{shut}$  denotan la energía disipada en el modo de transmisión, recepción, *idle* y *shutdown*, respectivamente. Nótese que el consumo en los modos de transmisión y recepción son sustancialmente mayores que en *idle* o apagado (*shutdown*), luego maximizar el tiempo en que se está en estos modos producirá un menor consumo del dispositivo. Esto queda patente en la ecuación (5.11) que representa la energía en *shutdown*,  $E_{shut}$ , puesto que la segunda parte de la misma es proporcional al número de veces que se realiza la transición *shutdown-idle*, denotada por  $n$ .

Dichos valores de consumo se han establecido en función del chip transmisor de Radio Frecuencia (RF) Chipcon, CC2420 [RAMACHA07]. Posteriormente realizaremos un análisis más detallado de su Diagrama de Transición de Estados (DTE) y el papel que desempeña cada uno de ellos en el consumo energético total.

Para evaluar las prestaciones del protocolo EDETA, el cual intenta maximizar el tiempo en estados de bajo consumo durante periodos de transmisión duraderos, con el objetivo de disminuir lo máximo posible el consumo de energía de los nodos, se va a realizar una comparación entre el modelo de energía genérico aproximado anteriormente y el aplicado en EDETA.

El modelo de energía para el modo *LE* se ha establecido de forma que dependiendo del número de transiciones *shutdown-Idle* que se realicen y su consumo asociado durante el tiempo de transición indicado, se disipará una u otra cantidad de energía. Se ve claramente como la energía consumida mediante EDETA será teóricamente inferior a un protocolo que no realice ningún tipo de optimización de su tiempo en estado *LE*. Pasar en el estado *shutdown* el tiempo que pasaríamos en *idle* durante largos períodos de inactividad, es más eficiente energéticamente hablando.

### 5.1.3.2 Modelo de energía en términos de estados y transiciones realizadas

A efectos ilustrativos vamos a considerar el chip transmisor de Radio Frecuencia (RF) Chipcon CC2420 [RAMACHA07]. Dicho chip soporta los siguientes cuatro estados [EBERLE05]:

1. *Shutdown* o *Sleep*: Se apaga la radio y se deja el  $\mu$ C en modo de bajo consumo.
2. *Idle*: La radio está preparada para recibir la orden de cambio al estado de Transmisión o Recepción.
3. *Transmisión (tx)*: el radio-modem (RF) está transmitiendo.
4. *Recepción (rx)*: el radio-modem (RF) está recibiendo.

En la figura 5.21 se muestra el diagrama de estados correspondiente.

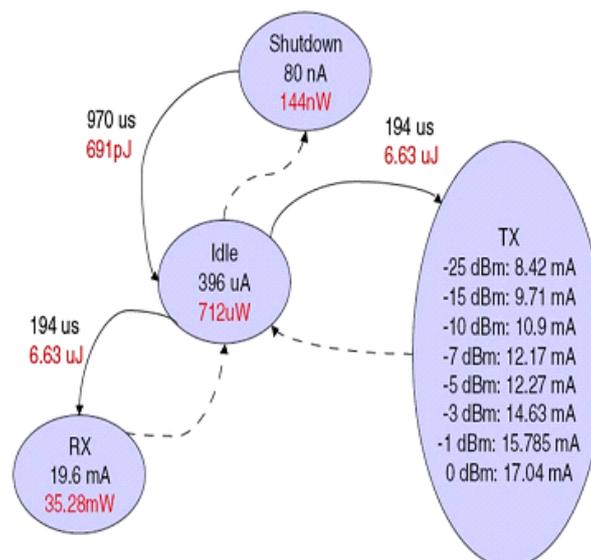


Figura 5.21. DTE incluyendo consumos entre transiciones y tiempo de transición (2v). Chip radio CC2420

De esta forma, la energía consumida asociada a la transición entre los estados  $S_1$  y  $S_2$  sería la proporcionada por (5.12):

$$\text{Energía de Transición} \cong t_{\text{transición}} \times I(S_2) \times V \quad (5.12)$$

donde  $t_{\text{transición}}$ ,  $I(S_x)$  y  $V$  denotan, el tiempo que se tarda en realizar dicha transición, la intensidad a la que se trabaja en el estado  $x$  y el voltaje de 2V, respectivamente.

Se observa principalmente dos cosas: primero, temporalmente hablando, puede costar mucho el realizar según qué tipo de transición. Esto es prioritario, por ejemplo, cuando se trata de pasar del estado *shutdown* al *idle*, donde el tiempo que conlleva realizar dicha transición es de casi 1ms. Y segundo, un mayor tiempo de transición y una mayor Intensidad asociada al estado final, conllevará una mayor energía disipada debido a la transición y, realizando mayoritariamente este tipo de transición, se obtendrá un mayor consumo medio. Debido a ello, permanecer o realizar transiciones moderadamente sucesivas a estados de poco consumo, se reflejará en un ahorro considerable de energía.

El objetivo, por tanto, será permanecer el mayor tiempo posible en estado de bajo consumo, esto es, con la radio en modo *sleep* para disminuir el gasto energético asociado y, a su vez, realizar el menor número de transiciones relacionadas con estados de alto consumo.

Una forma de optimizar dicho ahorro es apagando el mayor tiempo posible la radio entre transmisiones sucesivas y bien separadas, ya que de esta forma, eliminamos el caso en el que el nodo permanece en el estado *idle* para estar en modo *sleep*, disminuyendo su consumo al mínimo. Este apagado forzado ha demostrado ser muy efectivo en términos energéticos [FEENEY01].

De este modo, si no hay ningún paquete esperando a ser transmitido, el nodo, en vez de permanecer en el estado de *Idle*, lo hará en el de *sleep*. Cuando llegue el momento de transmitir un nuevo paquete, la radio se activará, pasando al consecuente estado y realizando la transmisión. Actuar de esta manera tampoco repercute en términos de productividad [RAMACHA07].

La expresión (5.13) modela el consumo medio energético en este caso:

$$C_{av} = t_{shut}E_{shut} + t_{rx}E_{rx} + t_{tx}E_{tx} + n_1(t_{shut-rx}E_{shut-rx}) + n_2(t_{shut-tx}E_{shut-tx}) + n_3(t_{rx-tx}E_{rx-tx}) \quad (5.13)$$

donde  $E_{state}$  es la energía consumida en el estado *state*,  $t_{state}$  representa el tiempo que se permanece en el estado *state* y  $t_{state-state}$  y  $E_{state-state}$  representan el tiempo necesario para realizar la transición *state-state* y la energía consumida por la misma, respectivamente.

Nótese que, la energía media disipada es proporcional al número de veces  $n_i$  que se realiza el producto  $t_{state} E_{state}$ , que no tiene por qué ser igual, relacionado con los estados directos de comunicación, luego a mayor número de transiciones mayor gasto. Por otra parte, la otra

proporcionalidad de la ecuación viene dada por los productos de las transiciones de los estados involucrados.

Para estimar la energía requerida por el radio-modem  $E_{rf}$  utilizaremos la expresión (5.14):

$$E_{rf} = V \times (I_{sleep} \times T_{sleep} + I_{idle} \times t_{idle} + I_{rx} \times t_{rx} + I_{tx} \times t_{tx}) \quad (5.14)$$

Siendo por tanto el objetivo para minimizar el consumo maximizar el tiempo que el radio-modem está en estado *sleep* o *shutdown*.

Como se indica en resultados experimentales en la bibliografía [RAMACHA07], a bajas tasas de envío, el nodo permanecerá la mayor parte de su tiempo esperando a recibir o transmitir paquetes. Por lo que apagar la radio durante estos tiempos de espera, como ya se comentó anteriormente, reducirá considerablemente el consumo. Esto se puede apreciar mejor en las gráficas presentadas en la figura 5.22.

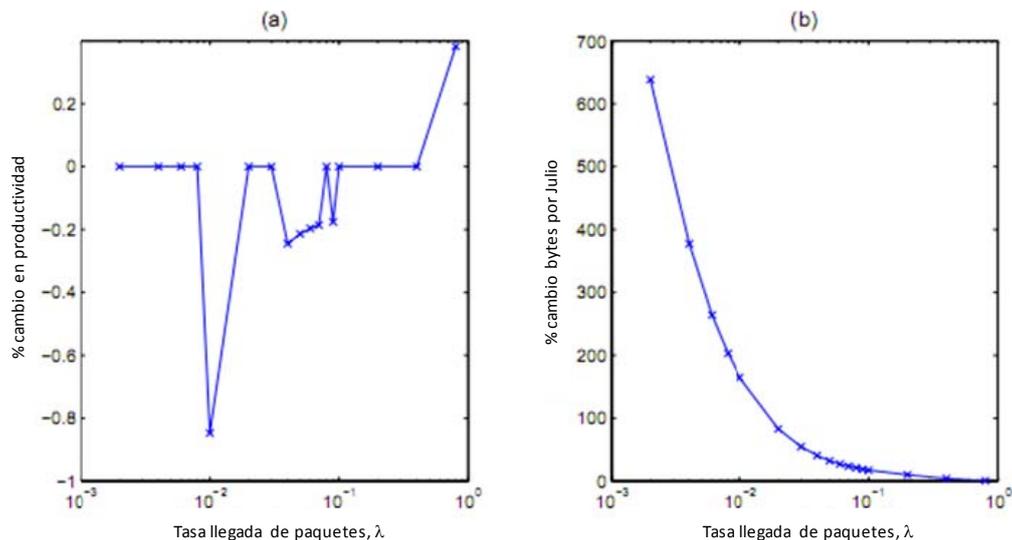


Figura 5.22. Porcentaje de cambio en la productividad (a) y bytes-por-Julio disipado (b) cuando se permanece en *Shutdown* en contrapartida de no hacerlo.

Se observa como los cambios realizados que influyen notablemente en el ahorro energético, lo hacen en menos de  $\pm 1\%$  en la productividad, dependiendo de la tasa de envío  $\lambda$ . Esta mejora es mucho más palpable a valores bajos de  $\lambda$ , ya que apagando la radio en los tiempos de inactividad, que van a ser mayoritariamente largos, disminuye el consumo.

### 5.1.3.3 Modelo de energía en términos de potencia consumida

Desde un punto de vista dedicado a analizar el tiempo en que el dispositivo de radio está en funcionamiento, se ha formulado el siguiente modelo de consumo de potencia [SHIH01] mediante la expresión (5.15).

$$C_{av} = N_{tx} [P_{tx} (T_{on} + T_{st}) + P_{out} T_{on}] + N_{rx} [P_{rx} (R_{on} + R_{st})] \quad (5.15)$$

donde  $P_{tx/rx}$  es el consumo de potencia del transmisor-receptor,  $P_{out}$  la potencia de salida del transmisor,  $T/R_{st}$  el tiempo de *start-time* del transmisor-receptor y  $N_{tx/rx}$ , el número de veces que conmuta el transmisor-receptor por unidad de tiempo, que depende de la tarea y del método de acceso al medio (MAC) que se emplee.  $T_{on}$  puede ser reescrito como  $L/R$ , donde  $L$  es el tamaño del paquete y  $R$  la tasa de envío de datos. Según valores de laboratorio y utilizados en la bibliografía, transmisores de bajo consumo poseen valores típicos para  $P_{tx}$  y  $P_{rx}$  de alrededor de 20dbm y de 0dbm para  $P_{out}$  [NATSEMCOO].

Assumiendo el modelo de propagación de la señal en entornos abiertos (*Free Space Model*), se emplea de forma general la ecuación (5.16) para calcular la potencia con la que un sensor receptor captará la señal enviada por otro sensor:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (5.16)$$

Debido a los diferentes roles que adquieren los sensores, esta potencia cambiará dependiendo de si se está ejerciendo el rol de NL o el de nodo normal (hoja), puesto que el radio de transmisión que deben abarcar es mayor o menor dependiendo de si se están comunicando dentro del *clúster* (*intra-clúster routing*) o fuera de él (*inter-clúster routing*).

Para un nodo normal, asumiendo que la zona de despliegue de la red es de  $100m \times 100m$ , el radio suficiente para comunicarse con el NL de su *clúster* podría ser de  $1/3$  de la distancia del lado del área de simulación, en este caso de  $\sim 30m$ , asumiendo un RF modem de frecuencia 779-928MHz, amperaje medio 10.44295417mA y voltaje 3.6V podría calcularse la potencia según (5.17):

$$P_r(30) = \frac{0.0376 \cdot 0.351^2}{(4\pi)^2 \cdot 30^2} = 3.259 \times 10^{-8} W^{(1)} \quad (5.17)$$

Para un nodo CH, la distancia cambia debido al proceso más disperso de *inter-cluster routing*, calculando una media en el radio de  $\sim 60m$ , se calcularía la potencia según (5.18):

$$P_r(60) = \frac{0.0376 \cdot 0.351^2}{(4\pi)^2 \cdot 60^2} = 8.148 \times 10^{-9} W^{(1)} \quad (5.18)$$

Las diferencias de consumo son palpables pero para ser totalmente coherentes, se va a proceder a realizar los mismos cálculos utilizando el modelo *Two-ray ground reflection model* (expresiones 5.19 y 5.20). Debido a que la señal tiene pérdidas conforme la distancia aumenta, este modelo refleja dicho efecto.

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} = \frac{0.0376 \cdot 0.2^2 \cdot 0.2^2}{30^4} = 7.427 \cdot 10^{-11} W \quad (5.19)$$

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} = \frac{0.0376 \cdot 0.2^2 \cdot 0.2^2}{60^4} = 4.642 \cdot 10^{-12} \text{ W} \quad (5.20)$$

Queda reflejado de este modo que, el consumo por parte del componente de radio de un nodo es proporcional al número de veces que conmuta el transmisor-receptor, esto es, el número de veces que se realiza una transición según el modelo de consumo presentado en el primer apartado. Dependiendo del tiempo que esté dicho transmisor encendido y del tiempo de encendido que posea, sumado a un valor constante de potencia de salida del propio transmisor, se consumirá más o menos. En la medida en la que nuestro módulo de radio esté el menor tiempo posible actuando, se conseguirá un ahorro de consumo considerable. Si se permanece un menor tiempo,  $T_{on}$ , en un estado en el que la radio realiza menos consumo del habitual,  $P_{tx}$ , se estará minimizando el gasto energético y por lo tanto realizando un gran ahorro, con el consecuente aumento en el tiempo de vida de nuestros dispositivos.

Esta estrategia es la que persigue EDETA. Actuando localmente en el tiempo en que un nodo, y su dispositivo de radio, permanecen encendidos y cambiando este comportamiento para disminuir dicho tiempo, apagando la radio durante los periodos de datos inactivos, se consigue un ahorro global en el consumo energético de nuestros dispositivos, sin descuidar la productividad de la red y alargando de forma notable el tiempo de vida de la misma.

#### 5.1.3.4 Modelo de energía del nodo

De una forma global es necesario considerar todos los requisitos de energía de un nodo, lo cual incluye los requisitos de energía de los sensores, microcontrolador y radio-modem.

Dependiendo del estado del nodo las necesidades de energía serán diferentes. Los requisitos totales de energía pueden expresarse según la expresión (5.21):

$$E_{total} = E_{rf} + E_{protocolo} + E_{sensorización} \quad (5.21)$$

Donde:

- $E_{rf}$  representa los requisitos de energía del módulo RF (radio-modem del nodo)
- $E_{protocolo}$  que representa los requisitos de energía necesarios para procesar la transferencia de datos, gestión de las señales del radio-modem y sobrecarga del protocolo.
- $E_{sensorización}$  se trata de los requisitos de energía para llevar a cabo las tareas de sensorización.

### 5.1.3.5 Modelo de energía en adaptación a las rondas operativas y DTE de EDETA

De una forma global, podemos visualizar la línea de tiempo de las operaciones del algoritmo organizada en macro-rondas, cada una de ellas subdividida en fases.

Cada fase posee un número de rondas determinado que alterna períodos de actividad dentro del clúster (*intra-clúster*) y en el árbol (*inter-clúster*), con períodos de inactividad o *sleep*. Estos periodos de inactividad serán distintos dependiendo del rol que esté desarrollando el nodo en ese momento. En la figura 5.23 se detallan las líneas de tiempo operacionales para cada rol.

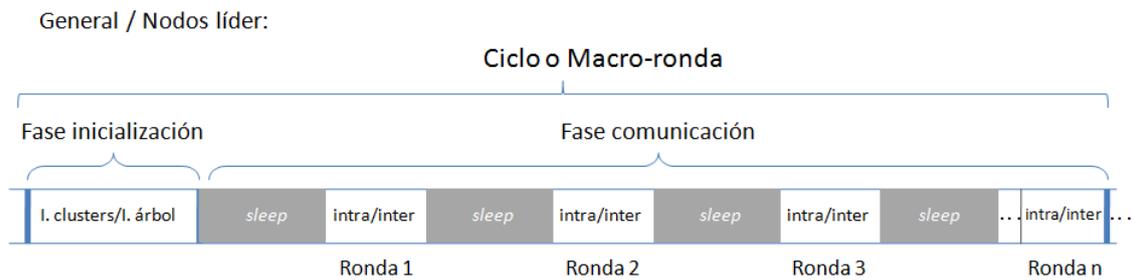


Figura 5.23. Línea de tiempo operacional de EDETA para NL, diferenciando sus operaciones y fases de la comunicación.

Debido a que estos períodos de inactividad establecen cotas bajas de consumo, como se indica en el modelo de energía en términos del modo de funcionamiento, se optimiza el nivel de batería de los dispositivos.

Para la *línea de tiempo* de los nodos líderes, en la fase de comunicación, un nodo líder dispondría de un máximo de  $n$  intervalos de *sleep*. La duración de este período de *sleep* vendrá determinada por la planificación otorgada por su padre y del *polling* a sus hijos, como se observa en su correspondiente *DTE* parcial en la figura 5.24.

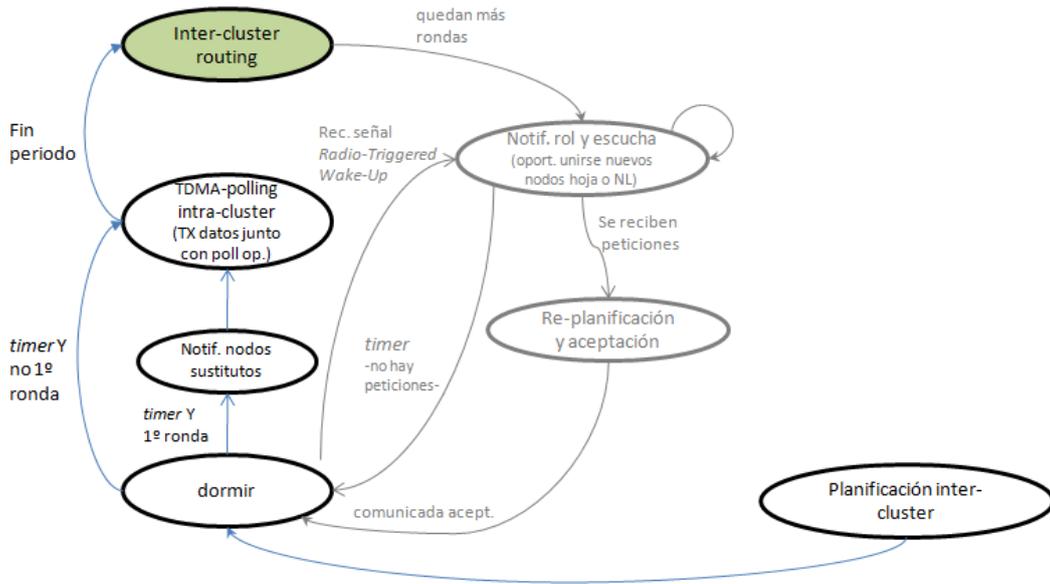


Figura 5.24. DTE parcial nodo líder

De la *línea de tiempo* anterior podemos aislar la fase de comunicación y detallar lo correspondiente a una ronda, quedando de la forma indicada en la figura 5.25.

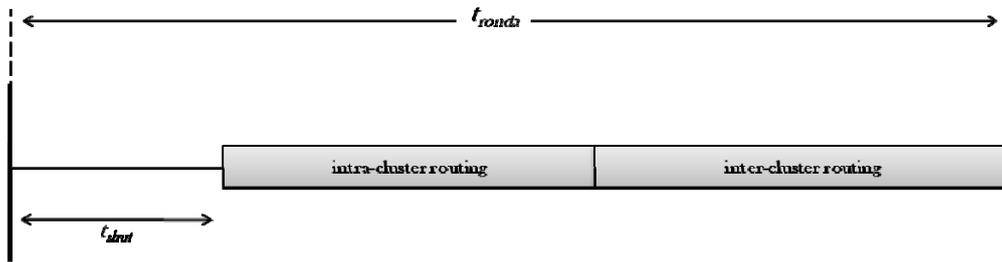


Figura 5.25. Línea de tiempo operacional para el NL

Dónde si desglosamos las fases *intra-clúster* e *inter-clúster*, obtenemos el esquema de la figura 5.26.

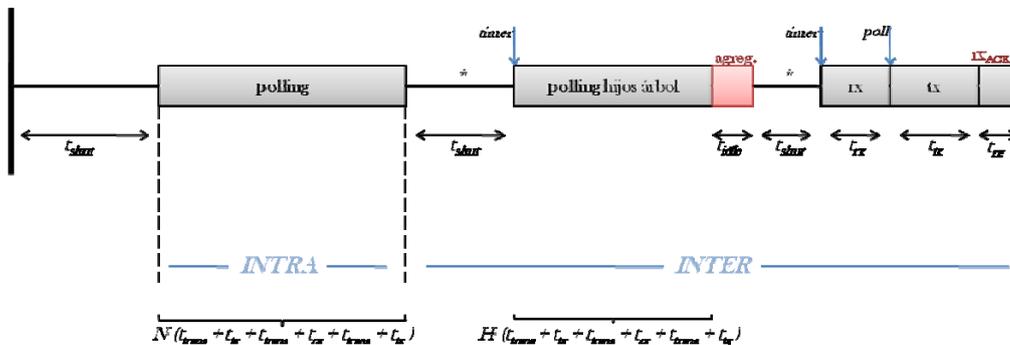


Figura 5.26. Línea de tiempo operacional para el NL desglosado en función de la fase

Teniendo las operaciones y los períodos delimitados de la figura 5.26 el siguiente significado:

- **polling**: tiempos de *polling* por parte del *NL* a los nodos hoja dentro del *clúster* en su ronda *intra-clúster*. En dicho periodo, se realizan transiciones entre los estados de transmisión y recepción, para mandar el *poll* a los sensores y recibir su información.
- **polling hijos árbol**: tiempos de *polling* por parte del *NL* a los posibles hijos dentro del árbol de *CHs*, donde después de los sucesivos *polls*, se realiza la agregación de la totalidad de los datos.
- **agreg**: periodo opcional pero muy recomendado para disminuir la cantidad de información que el *NL* transmitirá a su *NL*-padre.
- **rx**: periodo en el estado de recepción en el que se establece el *NL* una vez el *timer* de su planificación salta. Permanecerá en recepción hasta que su *NL*-padre le haga el *poll*.
- **tx**: periodo en el estado de transmisión en el que el *NL* transmite el agregado al *NL*-padre. Dicho tiempo no está acotado y dependerá del tamaño de la trama.
- **rx<sub>ACK</sub>**: periodo reservado para la recepción del *ACK* correspondiente al envío de los datos.

Los tiempos de *sleep* denotados con asterisco en la Figura 6 indican que, además de ser variables al igual que los demás tiempos de inactividad, si son relativamente pequeños, no conviene realizar el paso al estado de bajo consumo (*LE*), puesto que se disiparía más energía realizando varias transiciones que sin realizar ninguna de éstas.

De lo anterior se deduce, por tanto, la ecuación (5.22):

$$t_{sleep} = t_{ronda} - t_{intra} - t_{inter} \equiv [(t_{ronda} - t_{poll} - t_{agreg} - t_{txpadre})r]M \quad (5.22)$$

siendo  $r$  el número de rondas y el resto de valores los parámetros ya conocidos.

En condiciones óptimas, se garantizarían los  $n$  intervalos de inactividad, de mayor o menor longitud, con el consecuente ahorro de energía por parte del *NL*.

De la misma forma en la que se intenta maximizar el tiempo en que los *NLs* están en modo de bajo consumo para rentabilizar su nivel de batería, los nodos que efectúan una labor normal en las rondas *intra*, también disponen de mecanismos elaborados por EDETA para optimizar su gasto.

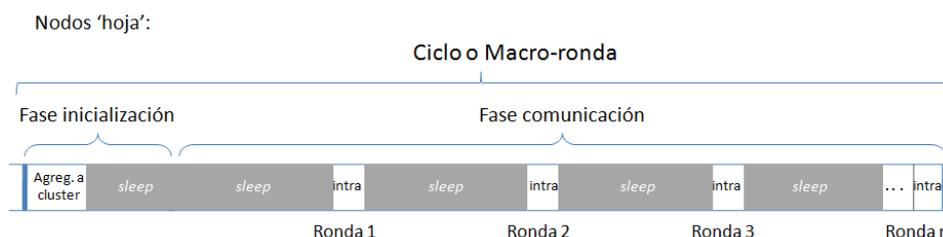


Figura 5.27. Línea de tiempo operacional de EDETA para el nodo normal, diferenciando sus operaciones y fases de la comunicación

Como se puede observar en la *línea de tiempo* correspondiente a los nodos hoja (figura 5.27), éstos también poseen intervalos de *sleep*, correspondientes al tiempo de inactividad que posee el nodo después de realizar las transmisiones oportunas en su ronda *intra-clúster*.

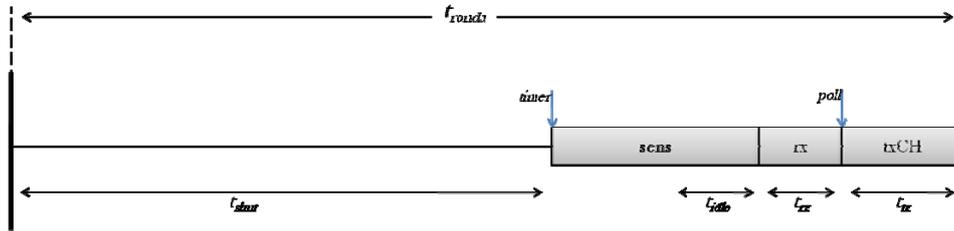


Figura 5.28. Línea de tiempo operacional para el nodo normal

Al igual que se hizo para el caso de rol de NL, aislamos la línea de tiempo del nodo normal dada una ronda en la figura 5.28, dónde *sens* y *txCH* indican los tiempos de sensorización (*sense time*) y transmisión al NL, respectivamente. A su vez, *timer* representa el instante de tiempo en el que el sensor despierta tras su periodo de *sleep*, tal y como se le indicó en su planificación. *Poll* representa el instante en el que el NL le realiza el *polling* para que inicie el envío de los datos recogidos.

Por tanto, se deduce la ecuación (5.23):

$$t_{sleep} = t_{ronda} - t_{sens} - t_{txCH} \quad (5.23)$$

En la figura 5.29 se muestra el diagrama, correspondiente a una sección del DTE para un nodo normal, no elegido como NL.

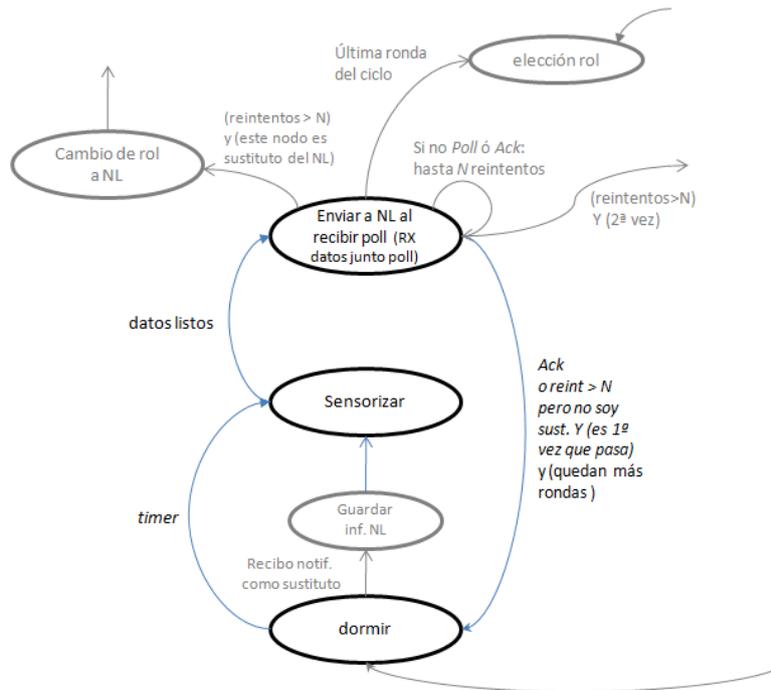


Figura 5.29. DTE parcial para los nodos normales o nodos hoja

Vemos como después de recibir la planificación desde el *NL*, se intercalan estados *sleep* entre los estados típicos de sensorización y transmisión, mejorando de esta forma el consumo energético ya que gracias a la planificación recibida, podemos saber cuánto tiempo podemos estar en modo de bajo consumo y aprovechar esta circunstancia.

Respecto al ámbito *inter-clúster*, en la fase de routing de la información hacia el *sink* pasando por los *CHs* padres, también se establecen mecanismos de ahorro de energía. Éstos se activan haciendo que el *NL* se duerma en el tiempo en el que el total de sus hijos le envían su información en la ronda *TDMA-polling*. Dicho tiempo podrá ser mayor o menor en la medida en la que la planificación se ajuste a la sensorización de los nodos y a la planificación recibida desde *CHs* de capas superiores.

En la figura 5.30 se puede ver el *DTE* correspondiente a esta etapa, donde los estados de *dormir* pueden ser largos o incluso no existir.

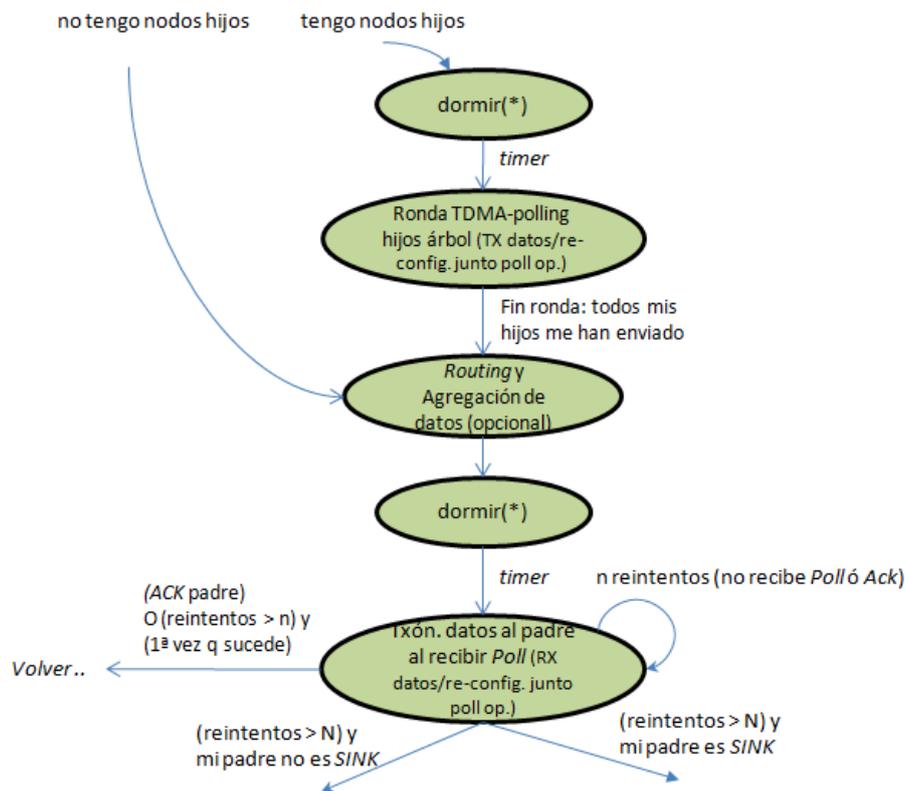


Figura 5.30. DTE parcial del *inter-clúster routing* de los nodos líderes

Además, vemos como también se realiza transición a *sleep* justo después de la agregación de datos opcional, esto es, cuando debo esperar debido a que, teniendo toda la información en mi poder, el *NL* padre ha indicado en mi planificación cuánto debo esperar para “subir” la información a través del árbol hacia el sumidero tras el correspondiente *poll* por su parte.

### 5.1.4 Metodología

El principal objetivo de este estudio es evaluar el consumo de energía, entre otros factores, del protocolo propuesto para redes inalámbricas de sensores, así como otras aproximaciones como son LEACH y estrella, realizando finalmente comparaciones entre los diferentes protocolos.

Para cumplir dicho objetivo, la metodología de evaluación básica ha consistido en definir y evaluar un conjunto de escenarios básicos de simulación. Dichos escenarios se han definido seleccionando los parámetros más representativos de una red inalámbrica de sensores. Partiendo de estos escenarios básicos se ha realizado un estudio de sensibilidad evaluando el impacto de los parámetros característicos de una WSN. Dichos parámetros se han modificado dentro de un rango de valores para obtener un amplio rango de posibles escenarios reales de simulación. Los parámetros seleccionados son los siguientes:

- Número de estaciones
- Dimensiones del área de simulación
- Patrón de tráfico

Como tráfico de red se ha utilizado fuentes de flujo constante, *Constant Bit Rate (CBR)*, definiendo el parámetro de tasa de envío.

El escenario básico utilizado ha consistido en una red inalámbrica de sensores de 100 estaciones distribuidas de forma aleatoria en un área de 100 x 100 metros.

Finalmente indicar que para obtener resultados fiables la técnica de simulación ha consistido en repetir cada experimento un número determinado de veces.

Cada uno de los resultados que se muestran a continuación se ha obtenido a partir de un número 'n' determinado de simulaciones que se han calculado tal y como se detalla a continuación.

Los resultados de cada una de las simulaciones llevadas a cabo se han considerado como variables aleatorias ( $X_1, X_2, \dots, X_n$ ) con media igual a  $\mu$ . Se han repetido un número 'n' de simulaciones hasta obtener una estimación de  $\mu$  con un intervalo de confianza del 90% utilizando la expresión (5.24):

$$\bar{X}(n) \pm t_{n-1,0.95} \sqrt{S^2(n)/n} \quad (5.24)$$

Donde  $t_{n-1,0.95}$  representa el límite superior de una distribución de Student t con  $n - 1$  grados de libertad y  $\bar{X}(n)$  y  $S^2(n)$  representan la media y la varianza de los resultados obtenidos de las diferentes simulaciones.

Por último indicar que los índices de prestaciones evaluados han sido la energía consumida por los protocolos de encaminamiento, tiempo de vida de la red, latencias y % paquetes perdidos.

## 5.2 Evaluación de prestaciones

En este apartado se presentan los resultados de la experimentación llevada a cabo tomando en primer lugar el escenario básico, y repitiendo a continuación las simulaciones empleando un amplio abanico de escenarios para evaluar el impacto de los parámetros indicados anteriormente y realizar las comparaciones comentadas.

Se realiza una evaluación de la bondad del protocolo EDETA analizando su estabilidad y la sobrecarga que introduce. Se evalúa el protocolo EDETA en diversos escenarios especificados a continuación y como índices de prestaciones se analizan la energía total consumida, la tasa de paquetes perdidos y el retardo medio extremo a extremo.

Los experimentos se han realizado con la implementación de EDETA desarrollada por el autor para el simulador ns2.33 [NS09], desarrollándose un modelo completo del protocolo, validado mediante experimentación con nodos reales, usando las extensiones  $\mu$ AMPS proporcionadas por el MIT [MITUAMP]. Estas extensiones fueron desarrolladas para el ns2.1b5, habiéndolas portado el autor a la versión actual ns2.33.

En los siguientes apartados se expondrán los parámetros de simulación, escenarios, resultados y comparaciones de EDETA con otro protocolo de referencia en todos los trabajos publicados en esta línea basado en *clustering* para ahorro energético como es LEACH [DEOSAR09] y con otro protocolo basado en estrella, con transmisiones directas al sumidero pudiendo dormir los nodos el resto del tiempo, considerado como caso ideal.

### 5.2.1 Parámetros y escenarios

Para obtener resultados realistas hemos ajustado los parámetros del modelo a medidas reales procedentes de experimentos realizados en nuestro laboratorio con nodos reales (CC1110 de Texas Instruments). El *transceiver* de radio consume en modo *sleep*  $3.6 \times 10^{-6}$  watts, en recepción 0.072 watts y en transmisión depende de la distancia, realizándose la estimación mediante el modelo *two ray ground model* [TYAN09] proporcionado por el NS2.

La energía usada por el circuito de radio es  $50 \times 10^{-9}$  J/bit, la velocidad de transmisión 250Kbps y la frecuencia utilizada 868MHz, dado que después de varios experimentos [CAPELLA07, MANZANO08] se ha concluido que es mucho más adecuada para redes inalámbricas de sensores que la banda de 2.4Ghz, tanto por consumo como por alcance de los nodos.

Todas las simulaciones se han realizado con 8 Julios de energía inicial. El sumidero se ha ubicado en el centro del escenario, y los nodos se han distribuido aleatoriamente. Se han realizado simulaciones con diferente número de nodos en escenarios de distinto tamaño, obteniéndose para cada escenario la media de los resultados obtenidos en las múltiples simulaciones del escenario.

Para llevar a cabo las simulaciones con LEACH, se han utilizado las extensiones mencionadas anteriormente. Mientras que para simular con el protocolo en estrella, se ha desarrollado un modelo del mismo para ns2.33, validado con resultados empíricos.

En LEACH, los *clúster heads* continuamente recaban datos de sus nodos, enviándolos al sumidero. En estrella los nodos envían sus datos al sumidero cada 20 segundos en el escenario base.

Las simulaciones con EDETA se han realizado con los mismos parámetros, si bien en este caso la distancia máxima de transmisión de los nodos se ha fijado en 100 metros. En cuando al parámetro propio del protocolo como es el tiempo de ronda se ha fijado en 20 segundos para el escenario básico.

### 5.2.2 Impacto del tráfico de red

La figura 5.31 muestra la energía consumida por los protocolos cuando se modifica el patrón de tráfico (tasa de paquetes enviados). Como era de esperar a mayor demanda de envío el consumo aumenta debido a que se deben reducir los tiempos de *sleep*, pero podemos observar como el protocolo propuesto sigue la tendencia del caso ideal. La red simulada dispone de 200 nodos en un área de 500 m.

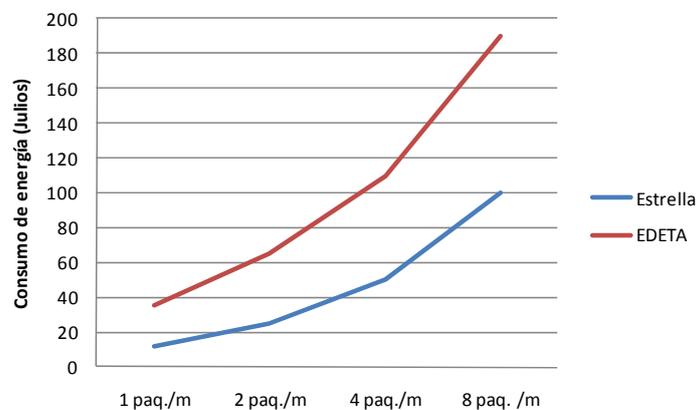


Figura 5.31. Estudio comparativo del consumo de energía medio de los protocolos de encaminamiento en función de la tasa de envío.

### 5.2.3 Impacto del número de nodos

La figura 5.32 muestra los resultados obtenidos cuando se modifica el número de nodos de la red manteniendo el tráfico del escenario básico en un área de 300m. Se han seleccionado 50, 100, 200 y 500 nodos. Las prestaciones de los protocolos estrella y LEACH se degradan con el número de estaciones, dejando de ser una solución escalable a medida que se incrementa el

número de nodos. Así, cuando pasamos del escenario básico a un escenario con 500 nodos el consumo de energía se incrementa en un 600%. Dicho incremento es debido a la poca escalabilidad que supone el que todos los *cluster heads* tengan que transmitir sus datos al sumidero incrementándose al haber más nodos las colisiones y por tanto las retransmisiones lo cual incrementa muy notablemente el consumo. Por otro lado en EDETA los nodos no tienen que transmitir al sumidero sino al nodo más cercano que hayan elegido, siendo necesaria la mínima energía. Además el protocolo distribuye y planifica las transmisiones hacia el sumidero, con todo ello se consigue reducir el consumo de energía.

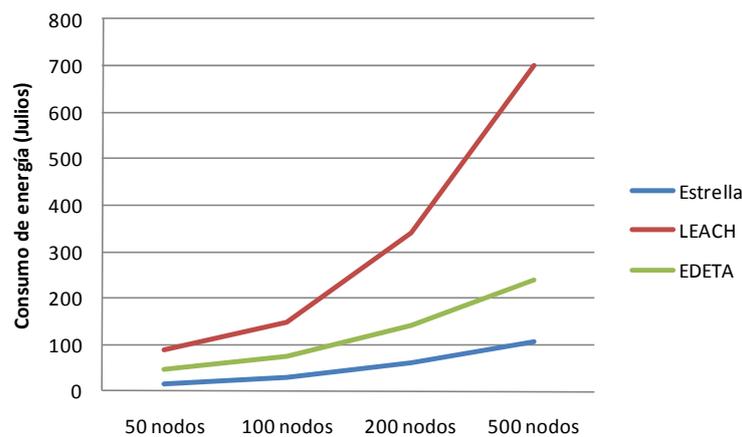


Figura 5.32. Estudio comparativo del consumo de energía medio de los diferentes protocolos de encaminamiento en función del número de nodos.

#### 5.2.4 Impacto del área de la red

En la figura 5.33 podemos observar los resultados obtenidos al variar el área de la red. Se han simulado escenarios de 100x100, 200x200, 500x500 y 1000x1000 metros con 200 nodos. Cuando se incrementa el área de simulación los protocolos cuyos nodos deben transmitir directamente al sumidero incrementan el consumo de energía mucho más rápido que el protocolo EDETA, el cual demuestra que escala muy bien debido a su segundo nivel jerárquico constituido por el árbol dinámico de nodos líderes, con el cual es posible cubrir áreas geográficas muy extensas manteniendo el consumo de energía.

Desde el punto de vista del número de nodos vivos sucede, como cabe esperar, lo mismo, en protocolos como LEACH al incrementarse el área de la red los nodos mueren más rápidamente (la red dura menos) que en EDETA.

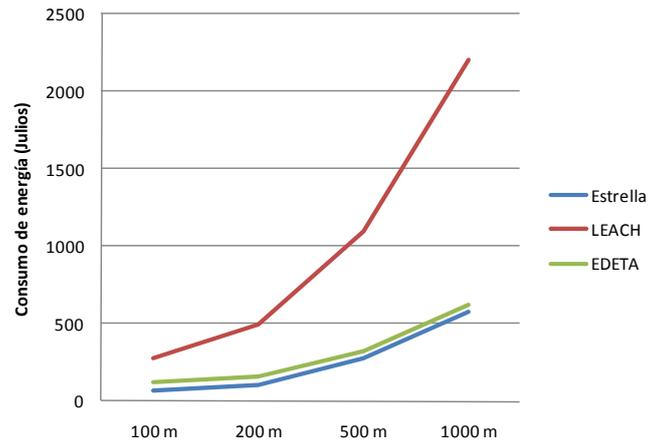


Figura 5.33. Estudio comparativo del consumo de energía medio de los diferentes protocolos de encaminamiento en función del diámetro de la red: 100x100m, 200x200m, 500x500m, 1000x100m.

### 5.2.5 Comparativa de EDETA, LEACH y ESTRELLA

La figura 5.34 representa el número de nodos vivos versus el tiempo, y en la figura Y podemos observar la energía consumida en función del tiempo, en las dos figuras se comparan los protocolos estudiados en los escenarios expuestos anteriormente.

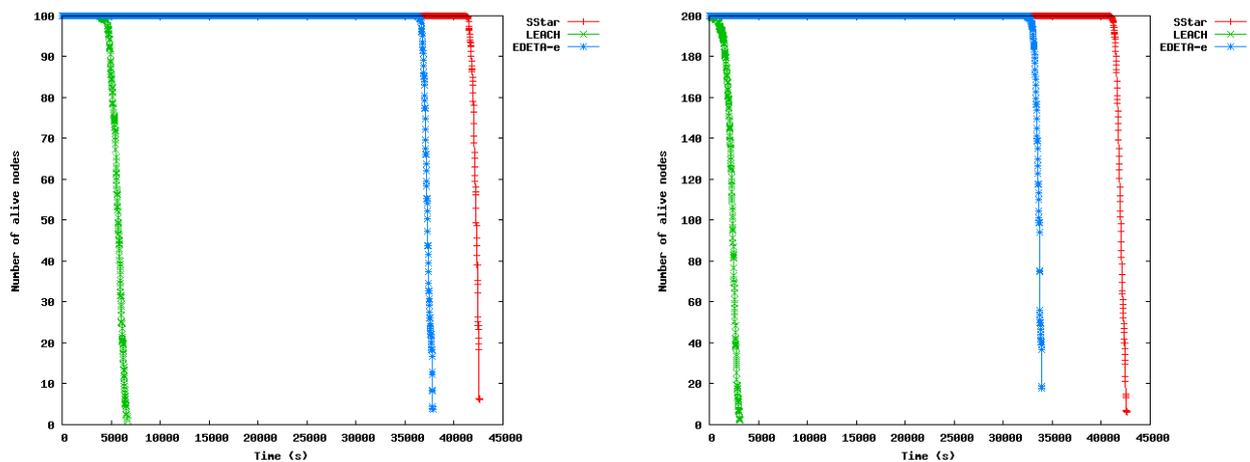


Figura 5.34. Nodos vivos respecto del tiempo. (a) 100 nodos, 200x200 m. (b) 200 nodos, 500x500 m.

Como era de esperar, ni el diámetro de la red ni el número de nodos afecta significativamente a la duración del protocolo en estrella en los escenarios simulados, como se muestra en las gráficas. Si bien, con escenarios más grandes la energía necesaria para que cada nodo alcance al sumidero hace que los nodos agoten las baterías en poco tiempo. Además, en la práctica, con un protocolo basado en estrella es totalmente inviable cubrir un área de 1km<sup>2</sup> debido a su nula escalabilidad.

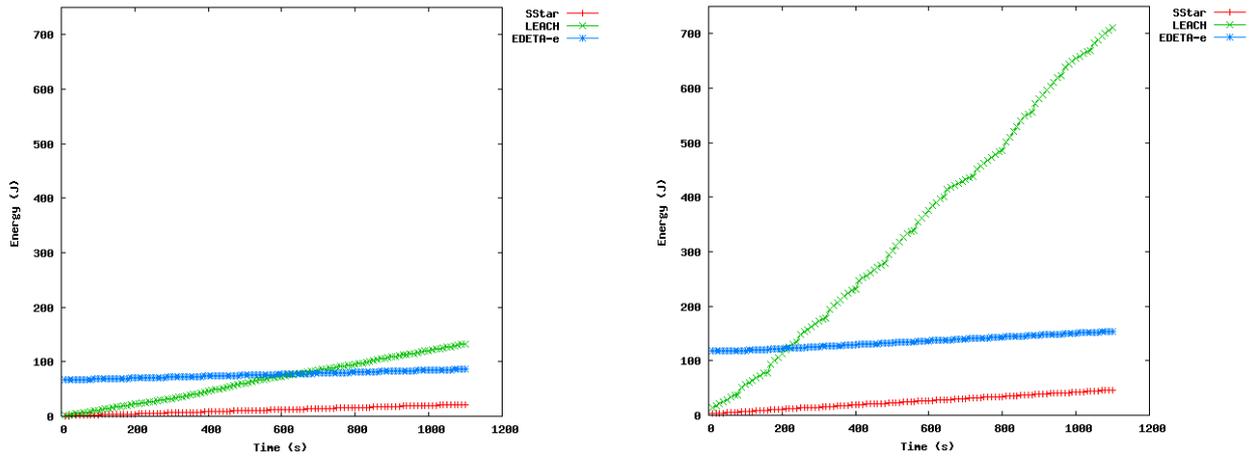


Figura 5.35. Consumo de energía en los primeros 1100 segundos. (a) 100 nodos, 200x200 m. (b) 200 nodos, 500x500 m

EDETA presenta un pequeño *overhead* al principio, en la fase de inicialización, por la construcción de las estructuras clúster y árbol, pero luego se muestra mucho más eficiente en el uso energético que las otras opciones, alargando de manera significativa el tiempo de vida global de la red.

Esto puede observarse en la figura 5.35, donde EDETA empieza con un consumo energético ligeramente mayor que las otras opciones. Pero después de la inicialización su consumo sigue la tendencia del caso ideal. Estos resultados se pueden observar también en la figura 5.36, que representa el consumo de los tres protocolos respecto del tiempo y del tamaño del área cubierta en una gráfica 3D.

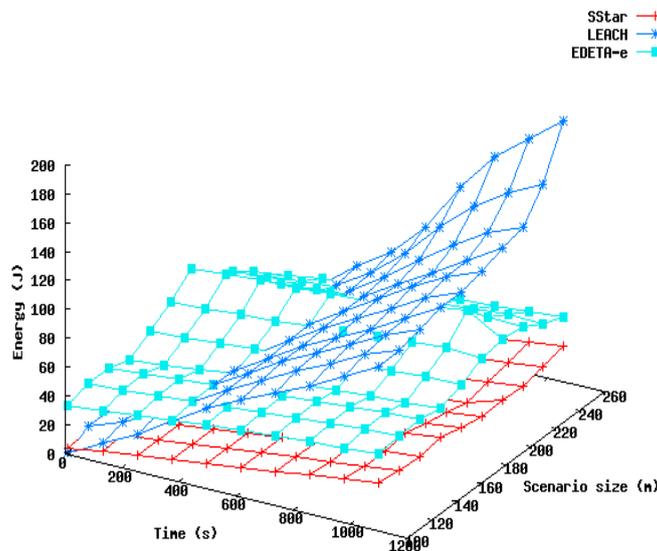


Figura 5.36. Consumo de energía en los primeros 1200 segundos. 100 nodos en áreas desde 100 a 260 metros de diámetro.

Estos resultados se pueden apreciar también en la figura 5.34. Los nodos en EDETA mueren un poco antes que el caso ideal (estrella), debido a la formación de las estructuras de la red. Pero a pesar de ello, como podemos ver en estas gráficas, el primer nodo en morir en EDETA ocurre 32000 segundos después que en LEACH, y el primer nodo que muere en EDETA ocurre tan solo 1700 segundos antes que en estrella. Todo ello presentando en todos los escenarios unas tasas de paquetes perdidos muy por debajo de otras aproximaciones como LEACH y mostrando además un comportamiento mucho mejor aun en superficies extensas. Todo ello se puede apreciar en la figura 5.37, donde se han simulado escenarios desde 200 nodos en un área de 200x200m hasta 400 nodos en un área de 400x400m, mostrándose para cada escenario la media de las repeticiones de las simulaciones de cada uno de ellos.

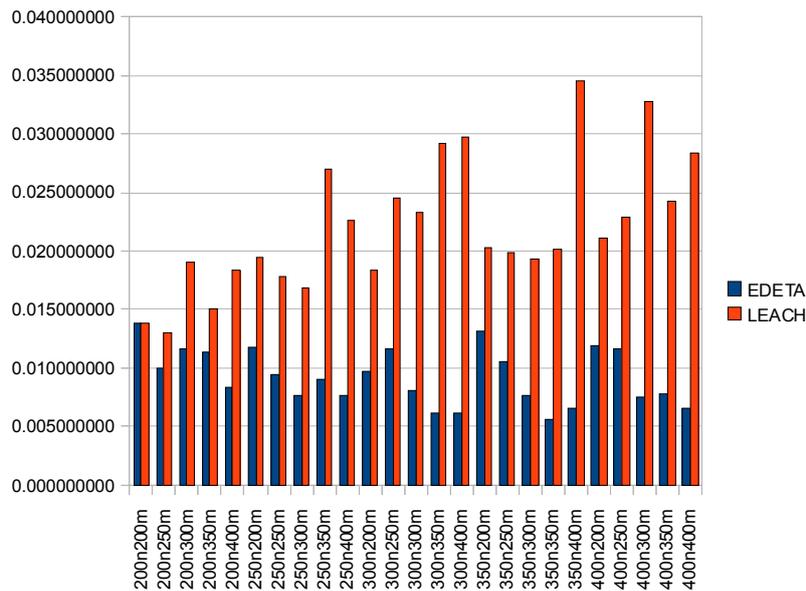


Figura 5.37. % Paquetes perdidos en LEACH y EDETA.

Por tanto, las simulaciones demuestran que las técnicas implementadas en EDETA son efectivas y reducen el consumo de energía desde una perspectiva global, reparten equitativamente (y de forma totalmente automática) el gasto energético, y multiplican la duración de la red. Los experimentos realizados muestran que EDETA reduce en un factor de 8 veces el consumo de un protocolo *power-aware* basada en clústeres como es LEACH, obteniendo que la red dure hasta 10 veces más que con dicho protocolo, proporcionando además características no presentes en las otras opciones, y que tras la experimentación se han demostrado eficaces, como son mecanismos de tolerancia a fallos, seguridad, alta escalabilidad, y tiempos acotados.

### 5.3 Campos aplicación de las técnicas propuestas

Las técnicas propuestas en la presente tesis se han implementado también en nodos reales para probar la viabilidad de las mismas y evaluar su comportamiento real. Se ha aplicado en diversos campos y proyectos como se mostrará a continuación. Además, previo a las implementaciones de los mecanismos de EDETA, se han puesto en marcha y evaluado – realizando diferentes mediciones de consumo, etc.- diversas plataformas hardware (figura 5.38) para redes inalámbricas de sensores (Telos [TELOS04], Micaz [MICAZ04], Jennic JN5121 [JENNIC51], CC2420 [CC2420\_06] y CC2430 [CC2430\_08] junto diversos microcontroladores como el C8051F310, C8051F040 y el C8051F041, etc.) desarrollándose los correspondientes niveles MAC y aplicaciones, y evaluándose también sistemas operativos como el TinyOS [LEVIS05] para redes de sensores basados en la plataforma Telos.

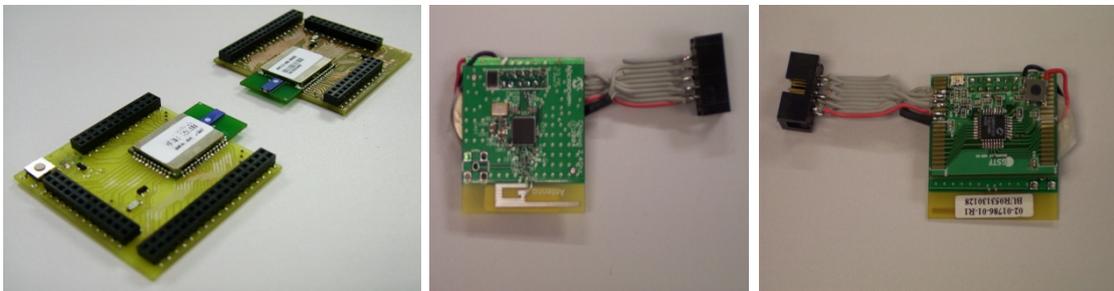


Figura 5.38. Diversas plataformas evaluadas para la implementación.

Las técnicas propuestas en la presente disertación se han aplicado en los siguientes campos y proyectos de investigación:

- Campo de aplicación: Domotica
  - Proyecto de investigación implicado: “Estudio e implementación de un sistema integral para la gestión y control de cargas eléctricas residenciales: Una oportunidad para la participación activa de usuarios finales en la disminución de picos de demanda y el uso eficiente de la energía”. Fundación Séneca – Murcia. Periodo 2006-2007.
  - “Desarrollo de un sistema único de protección perimetral escalable y robusto”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- Campo de aplicación: monitorización medioambiental, detección de incendios y plagas
- Proyectos de investigación implicados:

- “Desarrollo de un sistema de protección forestal mediante redes de sensores inalámbricos (SENIN)”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- “Monitorización medioambiental mediante redes de sensores inalámbricos (MARESIN)”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- “Detección precoz del picudo rojo mediante redes de sensores inalámbricas”. Generalitat Valenciana - Programa GESTA. Periodo 2006-2007.
- “Wireless smart fire alarm system”. Programa Europeo EUREKA (Subprograma AVANZA I+D) TSI-020400-2008-180. 2009-2011
- Tesis de máster: “Diseño e implementación de un sistema de detección de incendios mediante una red de sensores inalámbricos”. Autor: Guillermo Campos Bonet. Director: Juan Vicente Capella Hernández. Fecha de defensa: 10/07/2009 [CAMPOS09].
- Campo de aplicación: aplicación de las redes inalámbricas en entornos submarinos
  - Proyecto de investigación implicado: “Diseño, Evaluación e Implementación de Una Red Subacuática de Sensores”. Ministerio de Educación y Ciencia (CICYT DPI2007-66796-C03-01). Periodo 2007-2010.
- Campo: Mejora y optimización de las tecnologías para la aplicación de redes inalámbricas de sensores
  - Proyecto de investigación implicado: “Investigación básica fundamental sobre tecnologías constitutivas de un sistema de Red Inalámbrica de Sensores y su aplicación para el desarrollo de una Plataforma de Redes Inalámbricas de Sensores”. Ministerio de Educación y Ciencia (PET2008-0011). Periodo 2009-2011.

### 5.3.1 *Plataforma hardware para la implementación de EDETA y mediciones reales realizadas*

A continuación se presenta una de las implementaciones llevadas a cabo utilizando el transceiver Texas Instruments/Chipcon CC1101. En la tabla 5.1 se pueden apreciar los requisitos de corriente para los diferentes estados que soporta (*idle, receiving, transmitting, shut-down*), mostrándose tanto los datos contenidos en las especificaciones como las medidas reales realizadas en laboratorio, mostrándose por tanto parámetros importantes para dispositivos de bajo consumo que no vienen documentados en las especificaciones.

Parámetro	Corriente según hoja especificaciones	Corriente medida	Unidades
Modo recepción para 898 MHz (SRX)	15.7 (reduced mode)	19.6	mA
Modo "SLEEP" (SPWD)	0.2	0.07	uA
Modo transmisión, 868 MHz, FSK, 10 dBm	35.5	-	mA
Leakage current pin CS (chip not selected)	No documentado	< 1uA	N/A
Current leakage pin GD0	No documentado	0.1	mA

Tabla 5.1. Consumo corriente medido y declarado en hoja de especificaciones. VDD=3.3V

### 5.3.1.1 Cálculos estimación vida batería

El propósito de este apartado es analizar los requisitos reales de energía de los nodos implementados ejecutando por un lado EDETA y por otro una configuración en estrella, para compararlos. Aunque los resultados en las simulaciones realizadas son excelentes en este apartado se pretende demostrar que estos resultados son reales.

Se ha analizado un escenario de la aplicación de detección de termitas, considerando el peor caso para predecir la vida de los nodos, tomándose para todos los cálculos el máximo consumo de cada estado. Además se ha medido individualmente cada parte del nodo para dar el resultado más fiable posible, ya que el consumo de cada módulo depende del estado en el que se encuentre.

Para dimensionar los requisitos de batería la energía se ha medido en Amperios-hora (Ah) que es la unidad de capacidad utilizada en las baterías. La configuración en estrella ha sido la siguiente:

- 3 transmisiones por día
- 1 segundo por transmisión (potencia de transmisión: 10dBm)
- Recepción deshabilitada

La energía requerida por el nodo se muestra en la tabla 5.2. En dicha tabla se incluye la energía consumida por cada módulo del nodo y el tiempo en cada modo.

Descripción estado	Corriente (uA)	Tiempo en funcionamiento por día (s)	Requisitos anuales (mAh)
Microcontrolador en modo <i>sleep</i> + RTC	1.0	86313.60	8.75
Microcontrolador activo	3000.0	86.40	26.28
Sensirion SHT1x activo	900.0	5.00	0.45
Sensirion SHT1x desconectado	0.0		
Sensirion SHT1x <i>sleep</i>	0.3		
LED activo	1000.0	43.20	4.38
Sensor luz TAOS activo	780.0	43.20	3.41
Sensor luz TAOS desconectado	0.0		
RF modem CC1101 transmitiendo + <i>idle</i>	35000.0	3.00	10.64
RF modem CC1101 <i>sleep</i>	1.0	86397.00	8.75
RF modem CC1101 recibiendo + <i>idle</i>	19.6	0.00	
		<b>Energía requerida (mAh)</b>	<b>62.68</b>

Tabla 5.2. Requisitos de energía para cada módulo en la configuración en estrella. VDD=3.3V

Para EDETA, la configuración funcional equivalente utilizada en el prototipo ha sido:

- Los nodos realizan 3 transmisiones por día
- 1 segundo por transmisión (potencia de transmisión: 10dBm)
- 2 segundos para el *polling* del nodo líder (Rx), con la máxima sensibilidad en el receptor.

La tabla 5.3 resume los requisitos de energía para EDETA en esta configuración.

Descripción estado	Corriente ( $\mu$ A)	Tiempo en funcionamiento por día (s)	Requisitos anuales (mAh)
Microcontrolador modo <i>sleep</i> + RTC	1.0	86293.60	8.74
Microcontrolador activo	3000.0	106.40	32.36
Sensirion SHT1x activo	900.0	5.00	0.45
Sensirion SHT1x desconectado	0.0		
Sensirion SHT1x <i>sleep</i>	0.3		
LED activo	1000.0	43.20	4.38
Sensor luz TAOS activo	780.0	43.20	3.41
Sensor luz TAOS desconectado	0.0		
RF modem CC1101 transmitiendo + <i>idle</i>	35000.0	3.00	10.64
RF modem CC1101 modo <i>sleep</i>	1.0	86397.00	8.75
RF modem CC1101 recibiendo + <i>idle</i>	19600.0	6.00	11.92
		<b>Energía requerida (mAh)</b>	80.69

Tabla 5.3. Requisitos de energía para cada módulo para EDETA. VDD=3.3V

Dados estos resultados se ha estimado la vida de varios modelos de baterías comerciales empleadas en las implementaciones realizadas. La tabla 5.4 muestra la duración de los diferentes modelos de batería para la configuración en estrella, y la tabla 5.5 para la implementación basada en EDETA.

Modelo de batería	Capacidad (mAh)	Usable (%)	Duración estimada (años)
EMB er14250 3,6 V (seleccionada)	1100	75	13.16
Lithium-tionyl 2/3 AA 3,6 V	1700	75	20.34
Duracell 34mmx16,9diam. 3 voltios. DL12AB1 Ultra M3	1500	75	17.95
Duracell Ultra 3 v. 27x15.6 diam DCLR2	950	75	11.37

Tabla 5.4. Duración estimada baterías para la configuración en estrella

Por otro lado, la tabla 5.5 muestra la duración de los diferentes modelos de batería para la implementación basada en EDETA.

Modelo de batería	Capacidad (mAh)	Usable (%)	Duración estimada (años)
EMB er14250 3,6 V (seleccionada)	1100	75	10.22
Lithium-tionyl 2/3 AA 3,6 V	1700	75	15.80
Duracell 34mmx16,9diam. 3 voltios. DL12AB1 Ultra M3	1500	75	13.94
Duracell Ultra 3 v. 27x15.6 diam DCLR2	950	75	8.83

Tabla 5.5. Duración estimada baterías para la implementación basada en EDETA

Como se muestra en los cálculos realizados la implementación de EDETA basada en los parámetros indicados y utilizando el modelo de batería EMB er14250 3,6 V (batería económica) garantiza una duración de las baterías de más de 10 años, siendo una duración más que aceptable para una red inalámbrica de sensores, y más teniendo en cuenta que el protocolo proporciona muchos más mecanismos y características que una simple configuración en estrella.

### 5.3.2 Aplicación de la plataforma hardware

La implementación realizada consta de una serie de pequeños nodos (véase figura 5.39) dedicados a recolectar la información que deberá llegar al sumidero, que podemos apreciar en la figura 5.40.

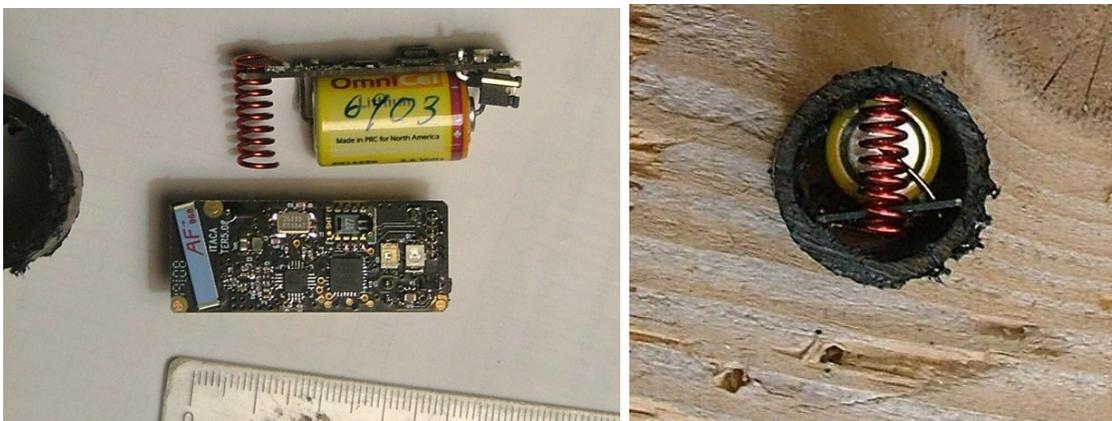


Figura 5.39. (a) Vista componentes PCB del nodo y con la batería instalada. (b) Nodo completo insertado en la madera.

Los nodos disponen de un microcontrolador Silabs C8051F930 de 8 bits basado en 8051. Se trata de un microcontrolador relativamente nuevo, muy flexible y que presenta modos de bajo consumo. El nodo realiza mediciones de temperatura y humedad mediante un sensor sensirion SHTx . Además para la detección de termitas el nodo dispone de un atrayente de insectos que son detectados mediante variaciones de la reflexión de la luz producida por un led de alta eficiencia (AVAGO) y un sensor de alta sensibilidad de TAOS. Este sensor se activa cada dos segundos.

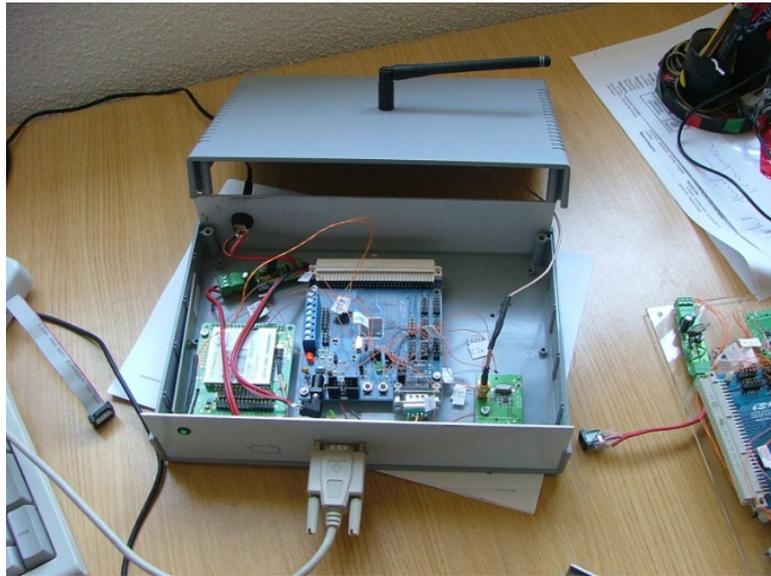


Figura 5.40. Vista interna del nodo sumidero.

El módulo RF del nodo utiliza un transceiver Texas Instruments/Chipcon CC1101 que opera en la banda ISM, habiéndolo configurado para trabajar en la banda ISM europea de 868 Mhz. Esta elección es un consenso entre el tamaño de la antena y la distancia a alcanzar. La banda de 2.4 Ghz utiliza antenas más pequeñas pero su comportamiento es peor ante fenómenos meteorológicos o objetos que se cruzan, alcanzándose menores distancias. La banda ISM europea de 433Mhz también fue considerada pero el tamaño de la antena necesaria es ya demasiado grande.

El sumidero fue construido ensamblando kits de evaluación y módulos comerciales como puede apreciarse en la figura C. El microcontrolador que coordina todo el sistema es el Silabs C8051F120, un microcontrolador de alto rendimiento que puede alcanzar picos de 100 MIPS. El radiomódem RF se basa en un kit de evaluación TI/Chipcon CC1101EMK868 al que montamos una antena  $\lambda/2$  dipolo de Antennafactor.

Los datos recolectados se agregan y se envían utilizando un kit GSM/GPRS Telit GM-862 a la base de datos remota.

## 5.4 Estudio del impacto del número de sumideros y su ubicación

Se ha realizado un estudio utilizando el simulador NS y modelos anteriormente presentados (los parámetros y topología de cada escenario se muestran en cada uno de los siguientes apartados) para determinar el impacto del número de nodos sumidero en una red inalámbrica de sensores así como la ubicación de dichos nodos. Los resultados resultan de interés a la hora de diseñar una red y planificar su despliegue, siendo de especial interés disponer de más de un

sumidero no solo ya por cuestiones de productividad (al ser este un cuello de botella para la WSN) sino también por tolerancia a fallos (punto simple de fallo).

Como ya sabemos, los nodos alrededor del sumidero se ven muy afectados energéticamente durante las simulaciones ya que todo el tráfico del escenario fluye hacia ellos y provocan una sobrecarga de energía en sus alrededores. Cabe destacar que conforme se aumenta el número de nodos sumideros la energía se reparte más adecuadamente entre todos los nodos, disminuyendo la cantidad de energía global que se necesita.

### 5.4.1 Sumideros en escenario *grid* - esférico

#### 5.4.1.1 Un nodo sumidero

Se puede apreciar en la gráfica de la figura 5.42 como para una configuración con 100 nodos y un único sumidero, el posicionarlo centradamente a la distribución total de nodos es preferente a colocarlo lateralmente ya que el consumo total de energía es bastante menor. La elección tomada pues es elegir al nodo azul como indica la figura 5.41. En los estudios realizados se concluye que para un escenario de rejilla o malla, la forma óptima de distribuir un sumidero sigue criterios parecidos al caso de un escenario esférico, donde la mejor posición es centrado.

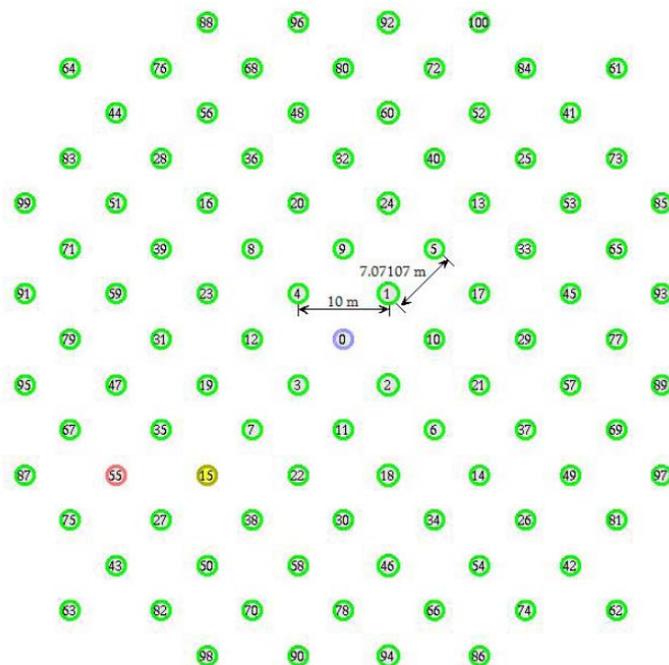


Figura 5.41. Disposición de los sumideros en las pruebas para 1 SINK en escenario *grid*-esférico

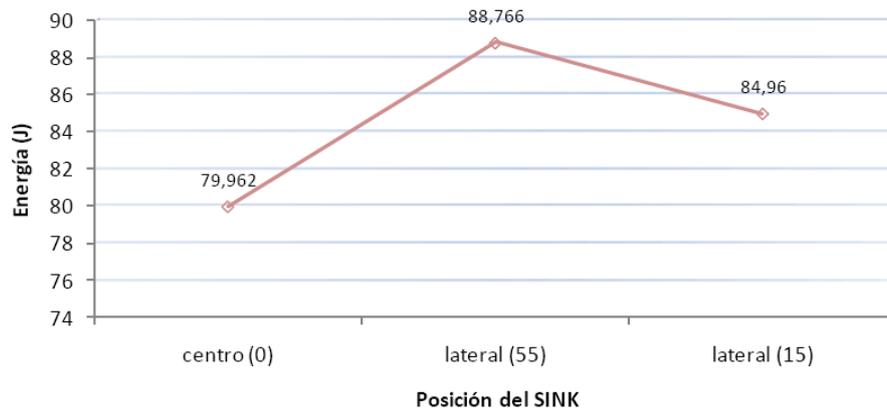


Figura 5.42. Impacto de la posición del SINK. 1 SINK escenario *grid*-esférico

#### 5.4.1.2 Dos nodos sumideros

En el caso de dos sumideros, hay dos configuraciones que obtienen buenos resultados. En concreto, para el caso rojo y gris existe un empate ya que ambas elecciones resultan en un bajo consumo de energía y alto porcentaje de recepciones CBR. Visualmente, como se puede verificar en las figuras 5.43 y 5.44, esto resulta en que la mejor manera de disponer dos sumideros en un escenario en *grid* o esférico es de forma separada centralmente, separándolos vertical u horizontalmente y no posicionándolos muy juntos.

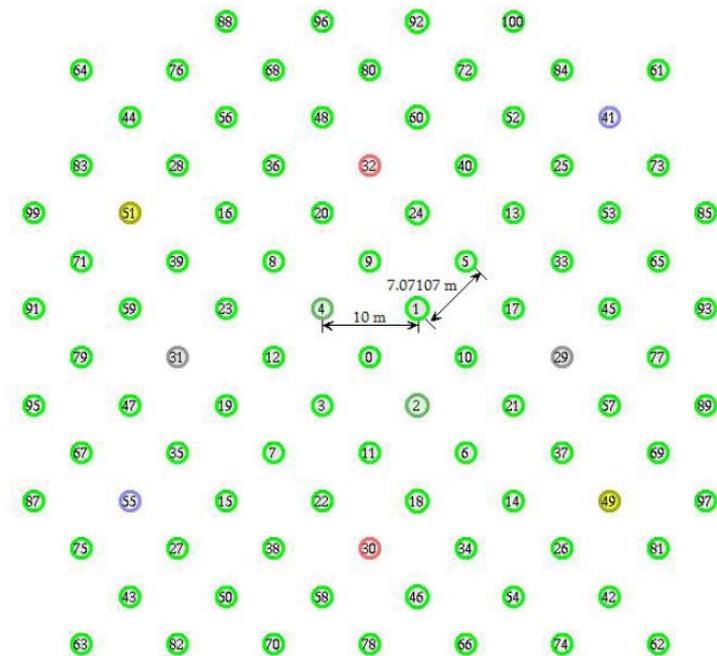


Figura 5.43. Disposición de los sumideros en las pruebas para 2 SINK en escenario *grid*-esférico

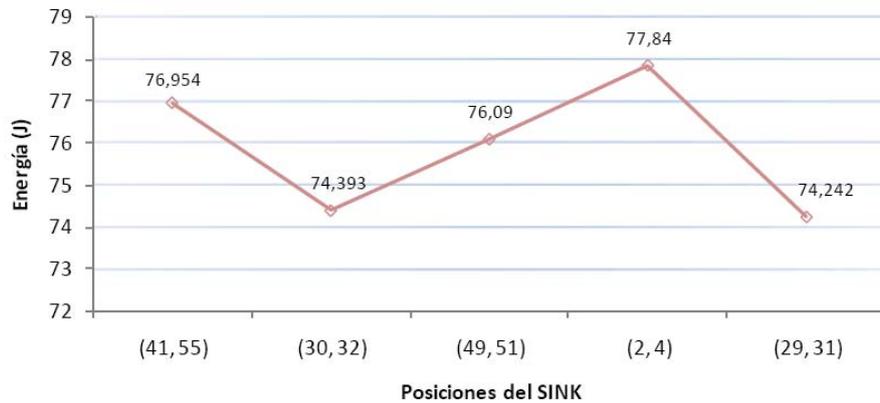


Figura 5.44. Impacto de la posición del SINK. 2 SINKs escenario *grid*-esférico

### 5.4.1.3 Tres nodos sumideros

Con una configuración de tres sumideros, una distribución geoméricamente triangular en primera instancia debería ser la que mejor resultados obtuviese (figura 5.45). Otro tipo de distribución a probar es una lineal separada, como se puede observar también en la figura 5.45. De entre estos dos tipos de distribución, como muestra la gráfica 5.46, la triangular es la que mejores resultados obtiene, y de entre éstas, la de color rojo. Hacer notar que de entre las dos distribuciones triangulares, ambas opciones no muestran diferencias significativas por lo que cualquier variación en las posiciones, mientras se conserven la disposición aproximada, resultará en una buena distribución.

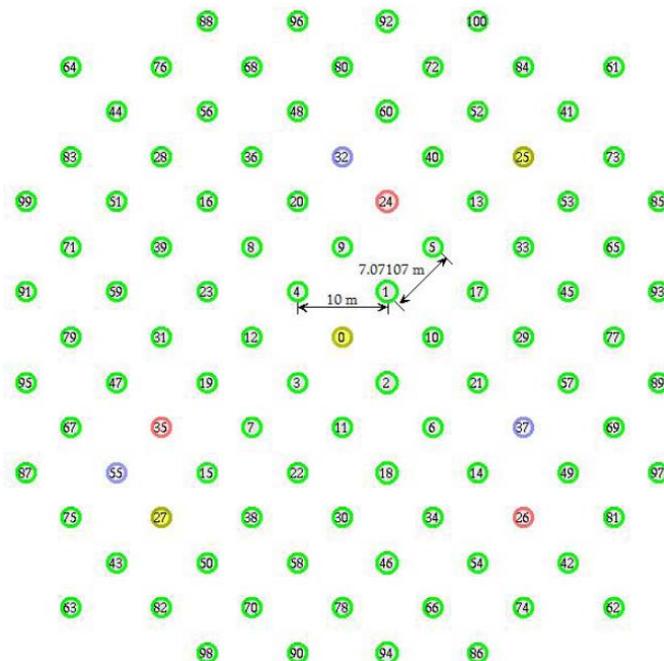


Figura 5.45. Disposición de los sumideros en las pruebas para 3 SINK en escenario *grid*-esférico

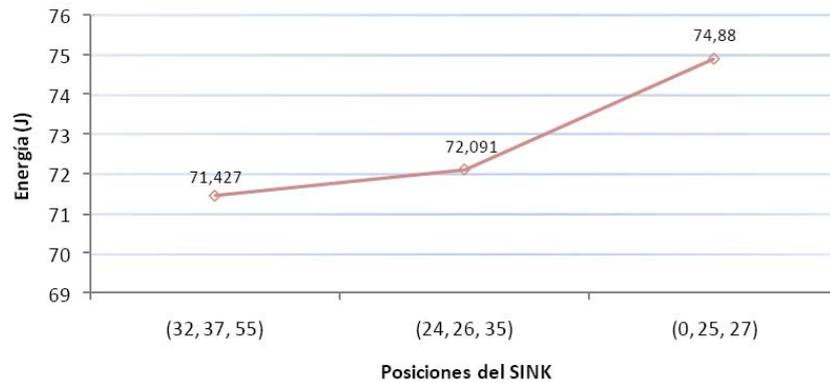


Figura 5.46. Impacto de la posición del SINK. 3 SINKs escenario *grid-esférico*

#### 5.4.1.4 Cuatro nodos sumideros

Con una configuración de cuatro sumideros (figura 5.47), una distribución geoméricamente distribuida en forma de rombo (o cuadrado) espaciado presenta buenos resultados, como puede apreciarse en la figura 5.48, donde para los sumideros resaltados de color rojo se consigue el mejor caso.

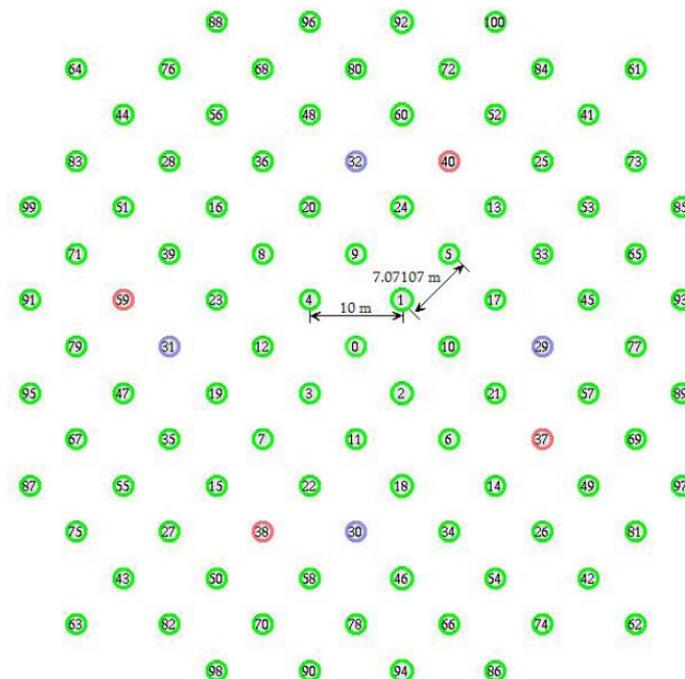


Figura 5.47. Disposición de los sumideros en las pruebas para 4 SINK en escenario *grid-esférico*

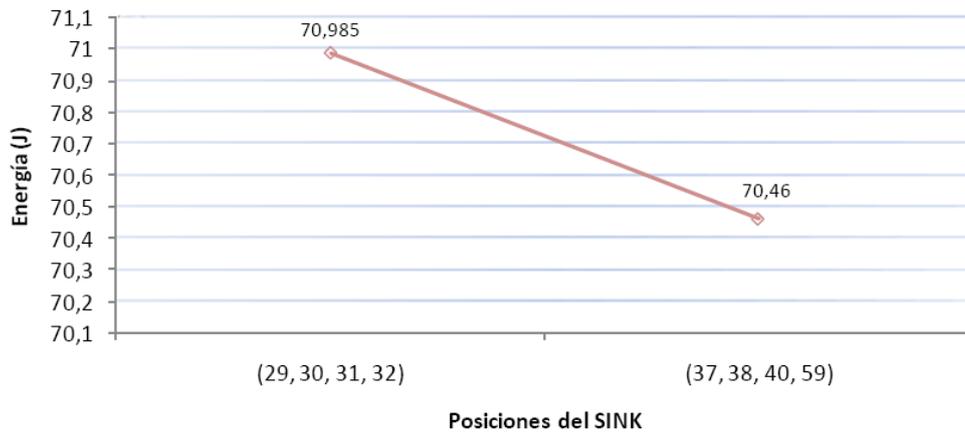


Figura 5.48. Impacto de la posición del SINK. 4 SINKs escenario *grid*-esférico

### 5.4.2 Sumideros en escenario en forma de camino

#### 5.4.2.1 Un nodo sumidero

En forma de camino (figura 5.49) destaca la buena elección del nodo sumidero central (en rojo), mostrando la gráfica de la figura 5.50 los buenos valores que cabía esperar.

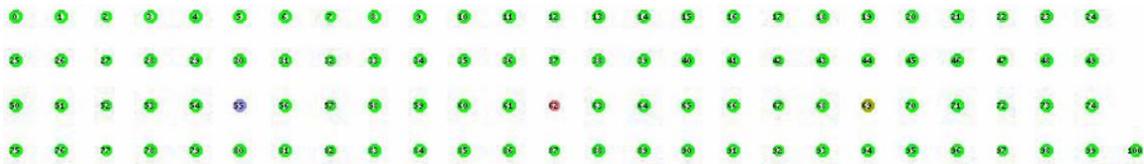


Figura 5.49. Disposición de los sumideros en las pruebas para 1 SINK. Escenario forma de camino

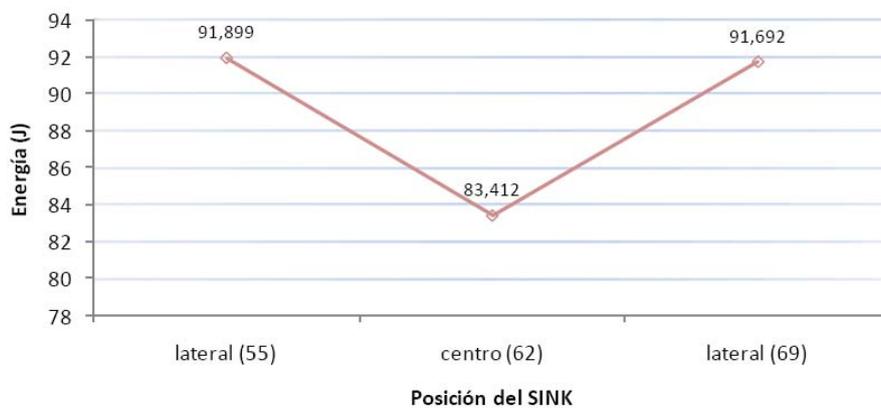


Figura 5.50. Impacto de la posición del SINK. 1 SINK escenario forma de camino

### 5.4.2.2 Dos nodos sumidero

Para un escenario alargado como este, es una buena apuesta constatada mediante los experimentos realizados, el situar linealmente separados los sumideros, obteniendo así como se puede apreciar en la gráfica de la figura 5.52, un notable ahorro energético y un gran porcentaje de recepción de paquetes al cubrir gran parte de la superficie. Esta solución corresponde al color azul en la figura 5.51.

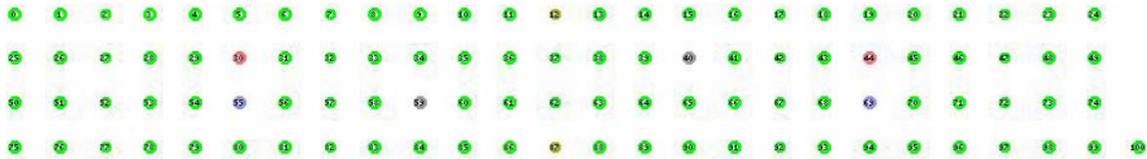


Figura 5.51. Disposición de los sumideros en las pruebas para 2 SINKs. Escenario forma de camino

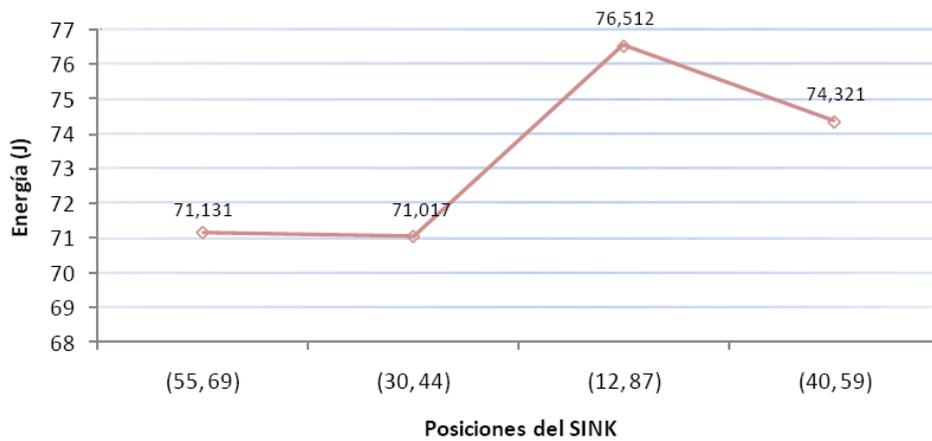


Figura 5.52. Impacto de la posición del SINK. 2 SINKs escenario forma de camino

### 5.4.2.3 Tres nodos sumidero

Los experimentos realizados nos vuelven a indicar la conveniencia de la distribución geométrica triangular a lo largo del camino como podemos observar en la figura 5.53, donde en el caso azul nos otorga muy buenos resultados (figura 5.54).

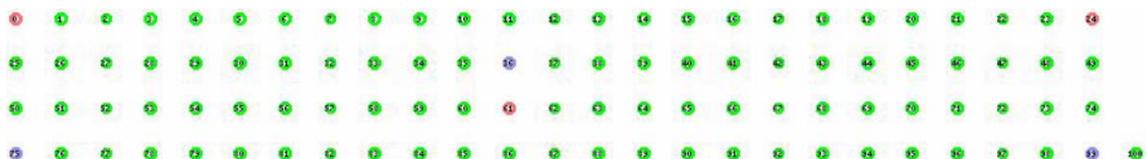


Figura 5.53. Disposición de los sumideros en las pruebas para 3 SINK. Escenario forma de camino

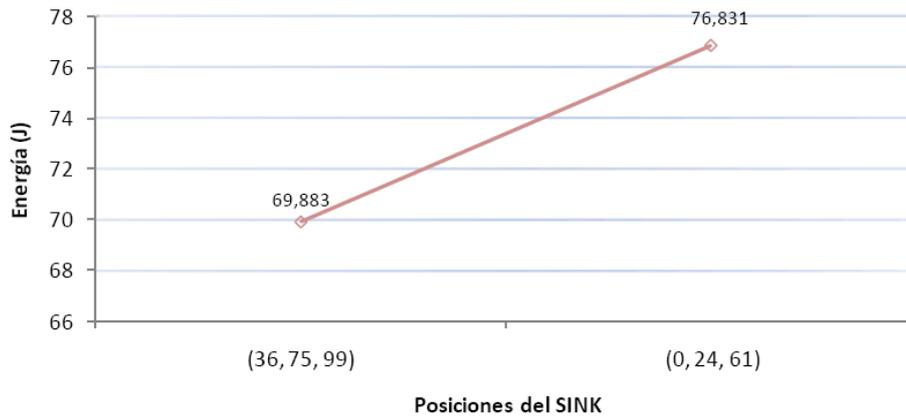


Figura 5.54. Impacto de la posición del SINK. 3 SINK escenario forma de camino

### 5.4.2.4 Cuatro nodos sumidero

Debido a la experiencia adquirida en los anteriores experimentos, se ha valorado la distribución triangular espaciada a lo largo del camino (véase figura 5.55) como la óptima, ya que se ve claramente como da muy buenos resultados en cuanto a consumo de energía como porcentaje de recepción de paquetes.

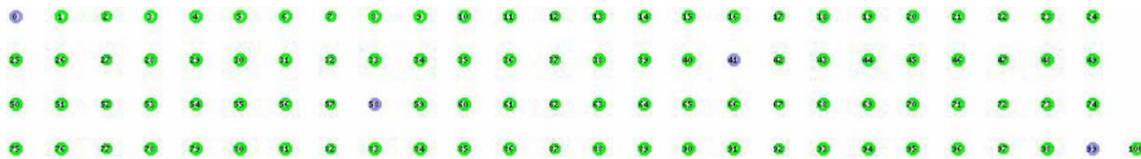


Figura 5.55. Disposición de los sumideros en las pruebas para 4 SINKs. Escenario forma de camino

### 5.4.3 Conclusiones

De un modo global podemos observar en las figuras 5.56 y 5.57 la influencia que tiene el número de sumideros que instalemos. Ante un mayor número de sumideros, el gasto energético de los nodos en total será menor, ya que se reparte el gasto alrededor de los diversos sumideros. De igual manera, puesto que un buen reparto de los sumideros resulta en que los nodos se distribuyan la carga transaccional entre todo los sinks, las latencias globales de envío disminuirán, obteniéndose buenos resultados tanto en términos energéticos como en porcentaje de recepción y latencias.

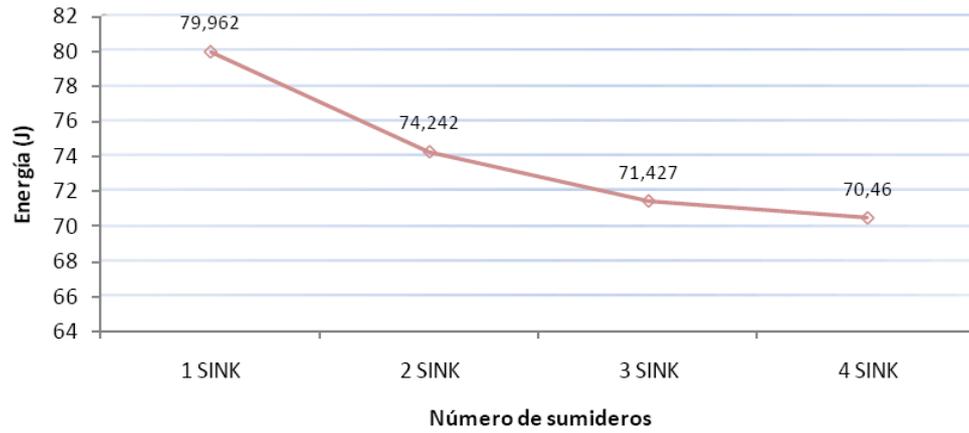


Figura 5.56. Impacto del número de sumideros sobre el consumo. Escenario *grid-esférico*

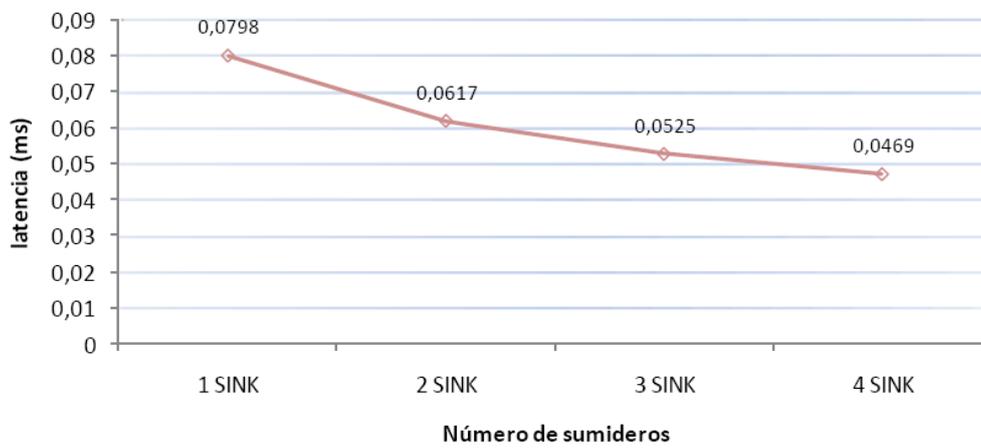


Figura 5.57. Impacto del número de sumideros sobre la latencia. Escenario *grid-esférico*

De igual manera, para el escenario en forma de camino la evolución es clara, de nuevo cuantos más sumideros distribuyamos por el escenario, mejores son los resultados tanto en cuanto a consumo energético como latencias, como podemos apreciar en las figuras 5.58 y 5.59.

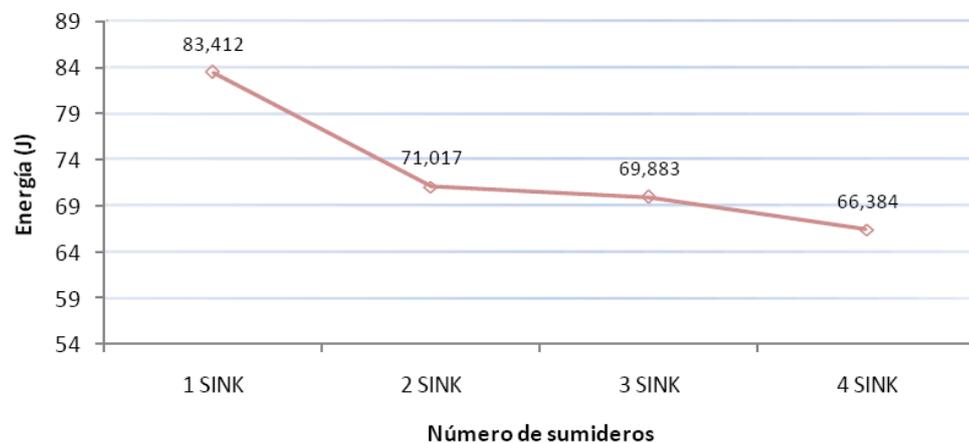


Figura 5.58. Impacto del número de sumideros sobre el consumo. Escenario forma de camino

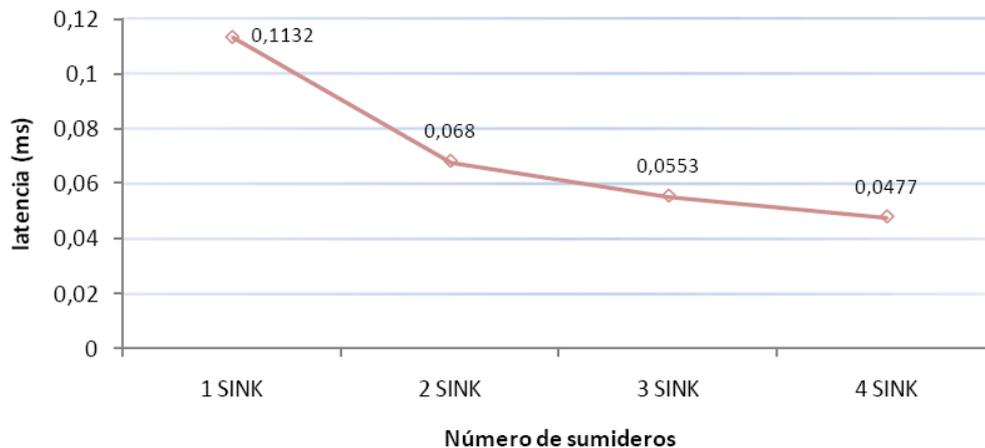


Figura 5.59. Impacto del número de sumideros sobre la latencia. Escenario forma de camino

## 5.5 Conclusiones finales

Se ha realizado un estudio comparativo del consumo de energía, duración de la red, y pérdida de paquetes utilizando diferentes protocolos de encaminamiento para redes inalámbricas de sensores. Se han simulado un amplio rango de escenarios modificando los parámetros más significativos de una WSN.

El protocolo EDETA propuesto en la presente disertación ha obtenido las mejores prestaciones frente a las otras aproximaciones de protocolos WSN jerárquicos [DEOSAR09], a la vez que presenta características innovadoras que no poseen las otras opciones.

La evaluación de prestaciones del protocolo EDETA muestra un funcionamiento estable y óptimo de las estructuras de clústeres y árboles, con un *overhead* muchísimo menor que otras aproximaciones que realizan cálculos de rutas o difusiones para hacer llegar la información al sumidero, todo ello de forma automática. Además presenta una detección temprana de fallos en estaciones gracias al uso de esquemas TDMA.

Los experimentos realizados, basados en medidas reales, demuestran que EDETA reduce en un factor de 8x la energía consumida comparado con los protocolos *power-aware* más populares, multiplicando por tanto la duración de la vida de la red, presentando una tasa de pérdida de paquetes notablemente menor, y ofreciendo además características novedosas en redes de sensores, a la par que necesarias para determinados nuevos campos de aplicación, como son mecanismos de tolerancia a fallos y tiempos acotados.

Además, se ha mostrado un estudio sobre el impacto del número de nodos sumidero y su ubicación, resultando de interés de cara al diseño y despliegue de la red.

Por último, tras los excelentes resultados obtenidos en las simulaciones, se han presentado las diversas pruebas e implementaciones en nodos reales realizadas para probar la viabilidad de las mismas y evaluar su comportamiento real, verificándose los resultados obtenidos mediante simulación. Además se han presentado un conjunto de proyectos de investigación actuales en los que se han aplicado las técnicas propuestas con resultados satisfactorios.

## **Capítulo 6**

# **Conclusiones, publicaciones y trabajo futuro**

### **6.1 Conclusiones**

Como primera conclusión de la presente tesis cabe destacar la consecución de todos y cada uno de los objetivos propuestos al comienzo de la misma.

En primer lugar, se ha realizado un estudio exhaustivo de las Redes Inalámbricas de Sensores (RIS), en el cual se han evaluado en profundidad más de cien propuestas de RIS en diversos estados de evolución, desde sistemas comerciales consolidados – los pocos que existen en la actualidad – hasta las últimas propuestas teóricas aparecidas en los más recientes artículos de investigación sobre el tema. Como fruto de dicho estudio se ha gestado el capítulo 2 de la presente Tesis, que analiza, compara y evalúa las propuestas más interesantes de las actualmente disponibles.

En segundo lugar, dentro del amplio campo de las RIS se centraron los esfuerzos en los algoritmos de MAC y encaminamiento, como aspectos fundamentales en las prestaciones de este tipo de sistemas. De este estudio se extrajeron valiosísimas conclusiones, que se emplearon en la definición del protocolo EDETA tal como se indica en el capítulo 3. Entre las conclusiones más interesantes de este estudio cabe destacar la elección del consumo de energía como parámetro fundamental, orientando los esfuerzos hacia la minimización de este parámetro. Un consumo eficiente prolonga el tiempo de vida de los nodos inalámbricos,

umentando el tiempo de vida de la WSN. Otro de los objetivos ha sido la robustez en las comunicaciones, cuestión también muy importante de cara a mantener la red operativa y minimizar el impacto sobre el consumo y los retardos en las comunicaciones en presencia de fallos, cuestión que resulta decisiva en ámbitos de aplicación industriales cada vez más demandados en la actualidad.

En tercer lugar, fruto de los estudios previos se concluyó que ningún protocolo cumplía totalmente las expectativas deseadas, por lo que se procedió a la propuesta de un nuevo protocolo, denominado EDETA, que mejorase además de forma significativa los parámetros de prestaciones propuestos en los estudios anteriores. El capítulo 4 especifica este nuevo protocolo, siendo esta propuesta una de las aportaciones más importantes de la Tesis. Además, este protocolo aporta otras características como flexibilidad, escalabilidad, seguridad y comunicaciones acotadas en el tiempo.

Como última conclusión, cabe destacar la verificación de las bondades del nuevo protocolo y la cuantificación mediante simulación de la mejora con respecto a los protocolos de referencia dentro del campo de las RIS. Dichos resultados pueden encontrarse en el capítulo 5.

Los hitos y resultados más sobresalientes de la presente Tesis pueden concretarse en:

- 1) Análisis exhaustivo de las diversas tecnologías inalámbricas, realizando además experimentos prácticos, habiéndose realizado estudios de interferencias [CAPELLA07, MANZANO08], evaluándose diversas plataformas hardware y desarrollando para ellas capas MAC y aplicaciones para medición de consumos [CAPELLA09b, CAPELLA09e], para determinar las más adecuadas a las fines perseguidos en redes inalámbricas de sensores. Una de las conclusiones a las que se llegó es que las tecnologías más apropiadas para nuestros fines son aquellas que utilizan bandas de frecuencia *sub-gigahercio*, al obtenerse menor consumo y mayores distancias.
- 2) Estudio general y evaluación de las diferentes aproximaciones al enrutamiento en redes de sensores, de este estudio se concluyó que la línea más apropiada la constituyen los protocolos jerárquicos basados en *clusterización*, determinándose los aspectos a mejorar en protocolos existentes, así como nuevas características necesarias que éstos no incorporan (Capítulo 2 y 3).
- 3) Tras las evaluaciones realizadas se ha llegado a la conclusión de que una de las mejores aproximaciones para obtener la máxima eficiencia energética son las redes heterogéneas, en las que los nodos hoja pueden llegar a ser extremadamente simples tanto en su constitución como funciones.
- 4) Otra conclusión es que la única forma de conseguir una WSN de bajo consumo es que el protocolo permita tener completamente apagados el mayor tiempo posible los nodos, cualquiera que sea el rol que ejecuten. Cuando el objetivo es reducir el consumo se debe considerar en bloque todos los niveles y el sistema completo por tanto [CAPELLA09d].

- 5) Se han realizado diversos experimentos estudiando la ubicación adecuada de los nodos sumideros en una red de sensores, concluyéndose que la mejor alternativa la constituyen los denominados “centroides” de la población de nodos para optimizar el consumo de energía y aumentar la vida de la red. Además, resulta fundamental un adecuado balance entre el número de nodos (tamaño de la red) y el de sumideros tanto desde el punto de vista del consumo como desde el punto de vista de la fiabilidad así como de la productividad y operatividad de la red.
- 6) Se han presentado nuevos mecanismos eficientes para el despliegue automático de redes inalámbricas de sensores sin necesidad de intervención humana que resuelve el problema de nodos huérfanos y mantiene de forma apropiada la sincronización entre nodos.
- 7) Se han propuesto diversos mecanismos de recuperación en caso de fallo de nodos sensores, nodos líderes y nodos sumidero, que se han demostrado eficaces, y que mantienen la red operativa y minimizan el impacto sobre el consumo y los retardos en las comunicaciones aún en presencia de fallos. Estos mecanismos no suponen una sobrecarga excesiva para los nodos, ni en cuanto a energía consumida, mensajes intercambiados ni tiempo de cómputo.
- 8) El ajuste de parámetros de una red inalámbrica de sensores depende directamente de los requisitos de la aplicación. No existen soluciones globales eficientes adecuadas para cualquier aplicación o escenario. Las características de las WSN hacen que el diseño de las mismas represente un compromiso de prestaciones. El mecanismo de elección de nodo líder padre en el árbol que se ha propuesto permite una óptima selección de rutas permitiendo prolongar el tiempo de vida de los nodos y la red.
- 9) Propuesta y evaluación de un nuevo protocolo (EDETA) para redes de sensores inalámbricas que contempla todos los objetivos y aspectos estudiados, analizando su comportamiento mediante diversos experimentos y pruebas prácticas, y comparándolo con otras alternativas existentes, habiendo demostrado una eficiencia energética superior proporcionando además características novedosas en protocolos de enrutamiento en redes inalámbricas de sensores como escalabilidad, flexibilidad (funcionamiento adecuado y transparente, sin necesidad de intervención humana, tanto en redes homogéneas como heterogéneas), baja sobrecarga, robustez, posibilidad de transmisión de datos desde el sumidero hacia los nodos incluso peer-to-peer entre nodos, seguridad, y comunicaciones acotadas en el tiempo, que es capaz de ejecutarse en microcontroladores de poca potencia, y que requiere de poca memoria incluso para ejecutar el papel de nodo líder, ya que no se basa en tablas de enrutamiento, lo cual redundaría en considerables ahorros de coste al implantar grandes redes de sensores.
- 10) Implementación del protocolo EDETA para el simulador de redes NS-2, que queda a disposición de la comunidad científica bajo licencia GPL. La evaluación de prestaciones del protocolo EDETA muestra un funcionamiento estable y óptimo de las estructuras de clústeres y árboles, con un *overhead* muchísimo menor que otras aproximaciones

que realizan cálculos de rutas o difusiones para hacer llegar la información al sumidero, todo ello de forma automática. Además presenta una detección temprana de fallos en estaciones gracias al uso de esquemas TDMA.

Los mecanismos anteriormente citados han sido evaluados utilizando un amplio abanico de escenarios de redes inalámbricas de sensores. Se ha realizado un estudio de sensibilidad de los diferentes mecanismos propuestos, evaluando el impacto de parámetros como el número de nodos, el diámetro de la red y la carga utilizada. Los índices de prestaciones estudiados son el consumo energético, duración de la red, latencias y porcentaje de paquetes perdidos. Además, dichos mecanismos han sido implementados en diversos proyectos reales para evaluar la bondad del protocolo, habiéndose obtenido resultados muy satisfactorios [CAPELLA10].

Los experimentos realizados, basados en medidas reales, demuestran que EDETA reduce en un factor de 8x la energía consumida [CAPELLA09] comparado con los protocolos *power-aware* de referencia [HEINZEL00, DEOSAR09], multiplicando por tanto la duración de la vida de la red, y ofreciendo además características novedosas en redes de sensores, a la par que necesarias para determinados nuevos campos de aplicación, como son mecanismos de tolerancia a fallos y tiempos acotados.

Habiendo cumplido todos los objetivos fijados, consideramos que el propósito final de la presente tesis ha sido alcanzado satisfactoriamente, concluyéndose por tanto la presente memoria.

## 6.2 Publicaciones relacionadas con la tesis

El trabajo de investigación relacionado con la presente tesis ha dado lugar a las siguientes publicaciones en revistas y conferencias internacionales (varias indexadas en *JCR-SCI*, *CORE* o *CS conference ranking*), donde se han publicado los resultados presentados en el presente trabajo:

- J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano, "Distributed and mobile systems based on wireless networks: definition of a generic control architecture", *Proc. IEEE International Conference on Industrial Technology*, Túnez, 2004.
- J.V. Capella, A. Bonastre, R. Ors, "Industrial applications of wireless networks: A bridge crane distributed control system based on Bluetooth", *Proc. IEEE International Conference on Industrial Technology*, Túnez, 2004.
- J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano, "New challenges in wireless sensor networks: fault tolerance and real time", *Proc. IEEE International Conference on Industrial Technology*, Hong Kong, China, 2005.
- J.V. Capella, J.J. Herrero, A. Bonastre, R. Ors, "Integration and application of emerging RF systems formed by wireless sensor networks: exhaustive experiment to their implantation in different environments", *Proc. IEEE International Workshop on Radio Frequency Integration Technology*, Singapore, 2007.

- J.V. Capella, J.J. Serrano, R. Ors, A. Bonastre, "Enabling ubiquitous wireless sensor networks: A new fault tolerant RF architecture", *Proc. IEEE International Workshop on Radio Frequency Integration Technology*, Singapore, 2007.
- A. Bonastre, J.V. Capella, R. Ors, M. Peris, "A mobile virtual interface for the monitoring and control of chemical analysis systems: Application to a wort fermentation process", *International Journal of Process Analytical Chemistry*, Nº 2, Vol. 10, 2008.
- M. Manzano, A. Bonastre, J.V. Capella, "Performance Study of Non-beaconed and Beacon-Enabled Modes in IEEE 802.15.4 Under Bluetooth Interference", *Proc. International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, Valencia, España, 2008.
- J.V. Capella, A. Bonastre, R. Ors, S. Climent, "A new energy-efficient, scalable and robust architecture for Wireless Sensor Networks", *Proc. IEEE/IFIP International Workshop on Wireless Sensor Networks: theory and practice*, El Cairo, Egipto, 2009.
- J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano, "A Wireless Sensor Networks Protocol for Distributed Analysis of Water", *Proc. International Conference on Wireless Networks*, Las Vegas, USA, 2009.
- J.V. Capella, A. Bonastre, J.J. Serrano, R. Ors, "A robust, energy-efficient and scalable wireless sensor networks architecture applied to a wireless fire detection system", *Proc. IEEE International Conference on Wireless Networks and Information Systems*, Shanghai, China, 2009.
- J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano, "A pollution monitoring system based on an energy-efficient Wireless Sensor Networks architecture", *IEEE Telecommunications Forum TELFOR*, 2009.
- J.V. Capella, A. Bonastre, J.J. Serrano, R. Ors, "An energy-efficient, scalable and robust wireless sensor networks architecture: application to habitat monitoring", *IEEE Telecommunications Forum TELFOR*, 2009.
- J.V. Capella, A. Bonastre, R. Ors, M. Peris, "A wireless sensor network approach for distributed in-line chemical analysis of water", *International Journal of Pure and Applied Analytical Chemistry, TALANTA, Elsevier*, Nº 5, Vol. 80, 2010.
- J.V. Capella, A. Bonastre, R. Ors, M. Peris, "In-line monitoring of ethanol along a fermentation process by means of a wireless sensor network", Enviado a: *International Chemical Congress of Pacific Basin Societies*, que tendrá lugar en diciembre de 2010 en Hawaii, USA.
- J.V. Capella, A. Perles, A. Bonastre, J.J. Serrano, "An energy-efficient scalable Wireless Sensor Network architecture: Application to historical building monitoring", Enviado a: *IEEE Int. Symposium on a World of Wireless, Mobile and Multimedia Networks*, que tendrá lugar en 2010 en Montreal, Canada.

### 6.3 Trabajo futuro

El trabajo desarrollado en esta tesis es innovador en muchos aspectos, por lo que abre muchas posibles líneas de trabajo, en algunas de las cuales ya se está trabajando en la actualidad.

Las principales líneas son:

- Estudiar la adaptación del protocolo EDETA a redes inalámbricas de sensores con movilidad: La versión actual del protocolo resulta extremadamente eficiente cuando los nodos son estáticos. Sin embargo, cuando un conjunto de nodos varía su posición relativa con respecto al sumidero con una velocidad apreciable las técnicas empleadas no resultan adecuadas, puesto que la reconstrucción del clúster debe realizarse de forma habitual. En este entorno, se pretende utilizar mecanismos propuestos en redes móviles para mejorar la eficiencia energética y aumentar la robustez en sistemas que requieran esta movilidad. En esta línea se ha solicitado un nuevo proyecto de investigación titulado “IRISAMI – Integración de Redes Inalámbricas de Sensores Autoconfigurables y Móviles en Internet” en la convocatoria de 2010 del Vicerrectorado de Investigación de la UPV. PAID-06-10
- Estudiar la adición de nuevos mecanismos de seguridad que no perjudiquen las características fundamentales del protocolo y lo hagan más seguro frente a nuevas amenazas.
- Estudiar y adaptar los parámetros y mecanismos del protocolo EDETA para su aplicación en redes subacuáticas de sensores (*Underwater Sensor Networks – USN*): Las redes subacuáticas de sensores constituyen un caso especial de las RIS que, por su elevado interés científico y comercial, han captado la atención de un buen número de investigadores. En este entorno el medio de transmisión no es adecuado para la transmisión vía radio, por lo que deben emplearse otros tipos de señales – en muchos casos, sónicas – para la transmisión. Este cambio de comportamiento hace que los protocolos habituales en RIS terrestres no sean aplicables, mientras que EDETA, gracias a su extrema flexibilidad, puede ser susceptible de adaptación únicamente modificando algunos parámetros de funcionamiento. Esta línea se encuentra actualmente en estudio, habiéndose solicitado para ello, entre otras cosas, un proyecto coordinado a la CICYT titulado “RISTA: Investigación Fundamental en Redes Inalámbricas de Sensores Terrestres y SubAcuáticas con nodos fijos y móviles. Aplicación a la monitorización y visualización de entornos marinos” en colaboración con la Universidad de Sevilla, la Universidad Miguel Hernández y la Universidad de Girona.
- Implementación de sistemas comerciales basados en EDETA: En la actualidad se encuentran en proceso de producción diversos dispositivos para la implementación de redes inalámbricas de sensores basados en el protocolo propuesto. A través de la empresa “*Wireless Sensor Networks Valencia*”, *spin off* de la Universidad Politécnica de Valencia, se pretende comenzar la comercialización de dichos dispositivos en los próximos meses.

- Patentes: Parte de los avances y resultados de esta Tesis están en proceso de patentabilidad por parte de la Universidad Politécnica de Valencia.

## Bibliografía

- [ADAIST] *ADA Project* (UE) URL: <http://www.adaist.org>
- [AKYILDI02] *"Wireless sensor networks: A survey"*  
F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci  
Computer Networks, Vol. 38, N° 4, pp.: 393–422, 2002
- [ALKARAK00] *"Routing Techniques in Wireless Sensor Networks: A Survey"*  
J.N Al-Karaki, A. E. Kamal  
IEEE Wireless Communications. Vol. 11, N° 6, 2000
- [ALKARAK04] *"Data Aggregation in Wireless Sensor Networks - Exact and Approximate Algorithms"*  
J.N. Al-Karaki, R. Ul-Mustafa, A. E. Kamal  
Proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR), Phoenix, Arizona, USA, 2004
- [AN040TI08] *"Folded dipole antenna for CC2400, CC2420, CC2430, CC2431, and CC2480"*.  
Application Note AN040. Texas Instruments. 2008
- [AWAKE] *AWAKE Project* (UE) URL: <http://www.awake-eu.org>
- [BALAGU09] *"Estudio Y Evaluación De Nuevas Tecnologías De Redes De Área Local"*  
Balaguer Díez, Enrique  
*Proyecto Final de Carrera*. Dirigido por Juan Vicente Capella Hernández  
E.T.S. Ingeniería Informática. Septiembre 2009

- [BELLA09] *“Estudio Y Comparación Mediante Simulación De Algoritmos De Enrutamiento Para Redes De Sensores Inalámbricas”*  
Bella Ripoll, Francisco Javier  
*Proyecto Final de Carrera*. Dirigido por Juan Vicente Capella Hernández  
E.T.S. Ingeniería Informática. Octubre 2009
- [BERMUD08] *“Evaluación y Optimización de Criterios para la Selección de Ruta en Redes Inalámbricas de Sensores”*  
Bermúdez Garzón, Diego Fernando  
*Tesis de máster*. Dirigida por Juan Vicente Capella Hernández  
Dpto. Informática de Sistemas y Computadores (DISCA). Diciembre 2008
- [BERTOC07] *“Performance Measurements of CSMA/CA-Based Wireless Sensor Networks for Industrial Applications”*  
M. Bertocco, G. Gamba, A. Sona, S. Vitturi  
IEEE Instrumentation and measurement technology conference (IMTC07)  
Warsaw, Poland, 2007
- [BJORNEM07] *“Two hops is one too many in an energy limited wireless sensor network”*  
E. Bjornemo, M. Johansson, A. Ahlen  
Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP '07. pp. 181-184, Hawaii, USA, 2007
- [BLUET09] IEEE 802.15.1™ IEEE Standard for Information technology.  
Telecommunications and information exchange between systems. Local and metropolitan area networks Specific requirements Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs™), 2009
- [BONASTR08] *“A mobile virtual interface for the monitoring and control of chemical analysis systems: Application to a wort fermentation process”*  
A. Bonastre, J.V. Capella, R. Ors, M. Peris  
International Journal of Process Analytical Chemistry, Nº 2, Vol. 10, 2008
- [BOUKER09] *“Algorithms and Protocols for Wireless Sensor Networks”*  
A. Boukerche (ed.)  
Wiley & Sons, New Jersey, USA, 2009
- [BRAGINS02] *“Rumor routing algorithm for sensor networks”*  
D. Braginsky, D. Estrin  
Proceeding of the 1st workshop on sensor networks and applications  
ACM Press, Atlanta, USA, 2002

- [CAMPOS09] *“Diseño e Implementación de Un Sistema de Detección de Incendios Mediante Una Red de Sensores Inalámbricos”*  
Campos Bonet, Guillermo  
Tesis de máster. Dirigida por Juan Vicente Capella Hernández  
Dpto. Informática de Sistemas y Computadores (DISCA). Julio 2009
- [CAPELLA04] *“Industrial applications of wireless networks: A bridge crane distributed control system based on Bluetooth”*  
J.V. Capella, A. Bonastre, R. Ors  
Proc. of IEEE Int. Conf. on Industrial Technology, Hammamet, Túnez, 2004
- [CAPELLA04b] *“Distributed and mobile systems based on wireless networks: definition of a generic control architecture”*  
J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano  
Proc. of IEEE International Conference on Industrial Technology, Túnez, Diciembre 2004
- [CAPELLA05] *“New challenges in wireless sensor networks: fault tolerance and real time”*  
J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano  
Proc. of IEEE International Conference on Industrial Technology, Hong Kong, China, Diciembre 2005
- [CAPELLA07] *“Integration and application of emerging RF systems formed by wireless sensor networks: exhaustive experiment to their implantation in different environments”*  
J.V. Capella, J.J. Herrero, A. Bonastre, R. Ors  
Proc. of IEEE International Workshop on Radio Frequency Integration Technology, Singapore, Diciembre 2007
- [CAPELLA07b] *“Enabling ubiquitous wireless sensor networks: A new fault tolerant RF architecture”*  
J.V. Capella, J.J. Serrano, R. Ors, A. Bonastre  
Proc. of IEEE International Workshop on Radio Frequency Integration Technology, Singapore, Diciembre 2007
- [CAPELLA09] *“A new energy-efficient, scalable and robust architecture for Wireless Sensor Networks”*  
J.V. Capella, A. Bonastre, R. Ors, S. Climent  
Proc. of IEEE/IFIP International Workshop on Wireless Sensor Networks: theory and practice, El Cairo, Egypt, Diciembre 2009

- [CAPELLA09b] *"A Wireless Sensor Networks Protocol for Distributed Analysis of Water"*  
J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano  
Proc. of International Conference on Wireless Networks, Las Vegas, USA,  
Julio 2009.
- [CAPELLA09c] *"An energy-efficient, scalable and robust wireless sensor networks architecture: application to habitat monitoring"*  
J.V. Capella, A. Bonastre, J.J. Serrano, R. Ors  
IEEE Telecommunications Forum TELFOR, Noviembre 2009
- [CAPELLA09d] *"A pollution monitoring system based on an energy-efficient Wireless Sensor Networks architecture"*  
J.V. Capella, A. Bonastre, R. Ors, J.J. Serrano  
IEEE Telecommunications Forum TELFOR, Noviembre 2009
- [CAPELLA09e] *"A robust, energy-efficient and scalable wireless sensor networks architecture applied to a wireless fire detection system"*  
J.V. Capella, A. Bonastre, J.J. Serrano, R. Ors  
Proc. of IEEE International Conference on Wireless Networks and Information Systems, Shanghai, China, Diciembre 2009
- [CAPELLA10] *"A wireless sensor network approach for distributed in-line chemical analysis of water"*  
J.V. Capella, A. Bonastre, R. Ors, M. Peris  
International Journal of Pure and Applied Analytical Chemistry, TALANTA,  
Elsevier, Vol. 80, Nº 5, 2010
- [CC2420\_06] *"CC2420 Data Sheet (rev. 2.1)"*  
Texas Instruments, 2006
- [CC2431D\_07] *"CC2431DK Development Kit User Manual (rev. 1.5)"*  
Texas Instruments, 2007
- [CC2430\_08] *"CC2430 Data Sheet (rev. 2.1)"*  
Texas Instruments, 2008
- [CERVERA09] *"Estudio Y Evaluación De Las Nuevas Tecnologías En Redes De Sensores Inalámbricas"*  
Cervera Andes, Adrian  
*Proyecto Final de Carrera*. Dirigido por Juan Vicente Capella Hernández  
E.T.S. Ingeniería Informática. Septiembre 2009

- [CHAVEZ07] *"Routing in wireless networks with position trees"*  
E. Chavez, N. Mitton, H. Tejada  
Conference on ad hoc Networks & Wireless (Ad Hoc Now'07), Morelia, Mexico, pp 32-45, 2007
- [CHEN07] *"A Uniformly Distributed Adaptive Clustering Hierarchy Routing Protocol"*  
J. Chen, F. Yu  
Proceedings of the IEEE international conference on integration technology, pp. 628-632, Shenzhen, China, 2007
- [CHOI04] *"A Framework for Energy-Saving Data Gathering Using Two-Phase Clustering in Wireless Sensor Networks"*  
W. Choi, P. Shah, S.K. Das  
Proceeding of the International Conference on Mobile and Ubiquitous Systems: Networking and Service (MOBIQUITOUS)  
IEEE Computer Society, Los Alamitos, CA, USA, 2004
- [CHU02] *"Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks"*  
M. Chu, H. Haussecker, F. Zhao  
The International Journal of High Performance Computing Applications, Vol. 16, Nº 3, 2002
- [CLIMENT09] *"Estudio E Implementación De Un Modelo De Encaminamiento En Redes De Sensores Inalámbricos"*  
Climent Bayarri, José Salvador  
*Proyecto Final de Carrera*. Dirigido por Juan Vicente Capella Hernández  
E.T.S. Ingeniería Informática. Marzo 2009
- [CSOINET] *CSONet Project* (Francia)  
URL: <http://www.nd.edu/~lemmon/projects/21CTF-260911/>
- [DEOSAR08] *"Clusterhead Selection in Clustering Algorithms for Wireless Sensor Networks: A Survey"*  
B. Deosarkar, N. Yadav and R.P. Yadav  
Proc. Int. Conf. Computing Communication and Networking (ICCCN 2008), 18-20 de Diciembre 2008, Karur, Tamilnadu, India.
- [DEOSAR09] *"The Effects of Spreading Factor on the Data Gathering Performance of Low Energy Adaptive Clustering Hierarchy (LEACH)"*  
B. P. Deosarkar, N. S. Yadav, R.P. Yadav  
International Journal of Recent Trends in Engineering, Vol. 2, No. 6, 2009
- [DINERO04] *"Sin cables, sin normas y a bajo precio"*  
Revista "dinero", observatorio tecnológico  
Julio 2004
- [DN020TI07] *"Programming Output Power on CC243x"* Design Note DN020  
Texas Instruments, 2007

- [DN112TI08] *"Using UART in CC111xFx, CC2431x, and CC251xFx"* Design Note DN112  
Texas Instruments, 2008
- [DTSSN] *Delay Tolerant Networks / Sensor Networks Project*  
URL: <http://www.sics.se/node/104>
- [EBERLE05] *"From myth to methodology: cross-layer design for energy-efficient wireless communication"*  
W. Eberle, B. Bougard, S. Pollin, F. Catthoor  
Proceedings of the Annual ACM IEEE Design Automation Conference, Anaheim, California, USA, 2005
- [ELHAFSI07] *"Cost over progress based energy efficient routing over virtual coordinates in wireless sensor networks"*  
E.H. Elhafsi, N. Mitton, D. Simplot-Ryl  
Proc. of International Workshop: From Theory to Practice in Wireless Sensor Networks (T2PWSN'07), Helsinki, Finland, 2007
- [EYES] *EYES Project (UE)* URL: <http://www.eyes.eu.org>
- [FANG03] *"Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation"*  
Q. Fang, F. Zhao, L. Guibas  
Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing (MOBIHOC), 2003
- [FAPOJUW09] *"Base-station Controlled Dynamic Clustering Protocol"*  
A. O. Fapojuwo, A. Cano-Tinoco  
IEEE Transactions on Wireless Communications, Vol. 8, Nº 10, 2009
- [FEENEY01] *"Investigating the energy consumption of a wireless network interface in an ad hoc networking environment"*  
L.M. Feeney, M. Nilsson  
Proceedings of IEEE INFOCOM 2001, pp. 1548-1557, Alaska, USA, 2001
- [GALISTE07] *"Redes De Sensores: Simulación De Redes Zigbee 802.15.4 Estudio De Flujo Continuo De La Red"*  
Galisteo Ruiz, Noelia  
*Proyecto Final de Carrera*. Dirigido por Juan V. Capella y Alberto Bonastre  
E.T.S. Ingeniería Informática. Junio 2007
- [GIGAANT] *GigaAnt, Titanis Data sheet*. URL: [www.antenova.com](http://www.antenova.com)
- [GU05] *"Radio-Triggered Wake-Up for Wireless Sensor Networks"*  
L. Gu, J. A. Stankovic  
ACM Real Time Systems, Vol. 29, No. 2, 2005
- [HOGTHRO] *Hogthrob project (Dinamarca)* URL: <http://www.hogthrob.dk>

- [HEINZEL00] *“Energy-Efficient Communication Protocol for Wireless Microsensor Networks”*  
Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan  
Proceedings of the Hawaii International Conference on System Sciences, Maui,  
Hawaii, USA, 2000
- [HEINZEL02] *“An Application-Specific Protocol Architecture for Wireless Microsensor  
Networks”*  
W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan  
IEEE Transactions on Wireless Communications, Vol. 1, No. 4, 2002
- [HUANG06] *“Dynamic Minimal Spanning Tree Routing Protocol for Large Wireless Sensor  
Networks”*  
G. Huang, X. Li, J. He  
Proceedings of the 1st IEEE Conference on Industrial Electronics and  
Applications (ICIEA'2006), Singapore, pp. 1531-1535. 2006
- [HUANG07] *“Tree-Clustered Data Gathering Protocol (TCDGP) for Wireless Sensor  
Networks”*  
K.-C. Huang, Y.-S. Yen, H.-C. Chao  
Proc. of the Conf. Future Generation Communication and Networking, FGCN  
Jeju-Island, Korea, 2007
- [IEEE15.4-06] *IEEE 802.15.4™ Standard for Information Technology - Telecommunications  
and Information Exchange Between Systems - Local and Metropolitan Area  
Networks - Specific Requirements – Part 15.4: Wireless Medium Access Control  
(MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal  
Area Networks (LR-WPANs), IEEE Std. 802.15.4-2006 (Revision of IEEE Std  
802.15.4-2003), September, 2006*
- [INTANAG03] *“Directed diffusion for wireless sensor networking”*  
C. Intanagonwivat, R. Govindan, D. Estrin, J. Heidemann  
IEEE/ACM Transactions on Networking, 2003
- [JENNIC51] *“Jennic Reference Manuals JN5121”*  
URL: <http://www.jennic.com>
- [KOHVAK06] *“Performance Analysis of IEEE 802.15.4 and ZigBee for Large-Scale Wireless  
Sensor Network Applications”*  
M. Kohvakka, M. Kuorilehto, M. Hännikäinen, T. D. Hämäläinen.  
PE-WASUN'06, Torremolinos, Málaga, España, Octubre 2006
- [KULIK99] *“Adaptive Protocols for Information Dissemination in Wireless Sensor  
Networks”*  
J. Kulik, W. Rabiner, H. Balakrishnan  
Proceedings of the 5th annual ACM/IEEE international conference on Mobile  
computing and networking, pp. 174 - 185  
Seattle, USA, 1999

- [KULIK02] *"Negotiation based protocols for disseminating information in wireless sensor networks"*  
J. Kulik, W.R. Heinzelman, H. Balakrishnan,  
Wireless Networks, Springer, Netherlands , 2002
- [KOUBAA06] *"Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks"*  
A. Koubaa, M. Alves, B. Nefzi, Y. Q. Song  
5th International Workshop on Real-Time Networks, Dresden, Alemania 2006
- [KOUBAA06b] *"GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks"*  
A. Koubaa, M. Alves , E. Tovar  
IEEE 20th International Parallel and Distributed Processing Symposium, 2006
- [LI01] *"Minimum-Energy Mobile Wireless Networks Revisited"*  
L. Li, J. Y. Halpern  
Proceedings of the IEEE International Conference on Communications (ICC),  
2001
- [LI05] *"An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks"*  
C. F. Li, M. Ye, G. Chen, J. Wu  
Proc. of IEEE International Conf. Mobile Ad-hoc and Sensor Systems,  
Washington, USA, 2005
- [LI06] *"An energy efficient clustering routing algorithm for wireless sensor networks"*  
L. LI, S.-S. DONG, X.-M. WEN  
The Journal of China Universities of Posts and Telecommunications. Vol. 13,  
Nº 3, pp. 71-75, 2006
- [LINDSEY02] *"PEGASIS: Power-Efficient Gathering in Sensor Information Systems"*  
S. Lindsey, C. S. Raghavendra  
IEEE Aerospace Conference, 2002
- [LIQ01] *"Hierarchical Power-aware Routing in Sensor Networks"*  
Q. Li, J. Aslam, D. Rus  
Proceedings of the DIMACS Workshop on Pervasive Networking  
Piscataway, USA, 2001
- [LEE06] *"Performance Evaluation of IEEE 802.15.4 for Low-Rate Wireless Personal Area Networks"*  
J. S Lee  
IEEE Transaction on Consumer Electronics, Vol. 52, No. 3, 2006

- [LEE09] *"An Energy-Efficient Distributed Unequal Clustering for Wireless Sensor Networks"*  
S. Lee, J. Lee, H. Sin, S. Yoo, S. Lee, J. Lee, Y. Lee, S. Kim  
Proc. of Int. Conf. on Communication Software and Networks, Macau, China, 2009
- [LEVIS05] *"TinyOS: An operating system for wireless sensor networks"*  
P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler  
In *Ambient Intelligence*, Springer-Verlag, 2005.
- [LOPEZ06] *"Redes De Sensores: Simulación De Redes Zigbee 802.15.4 Estudio De Flujo Degradado De La Red"*  
López Sierra, Milagros  
*Proyecto Final de Carrera*. Dirigido por Juan V. Capella y Alberto Bonastre  
E.T.S. Ingeniería Informática. Junio 2006
- [LU04] *"Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks"*  
G. Lu, B. Krishnamachari, C. S. Raghavendra  
IEEE International Conference on Performance, Computing, and Communications, pp. 701-706, 2004
- [MA08] *"A power controlled interference aware routing protocol for dense multi-hop wireless networks"*  
L. Ma, Q. Zhang, X. Cheng  
Journal of Wireless Networks, Vol. 14, No. 2, pp.: 247-257. 2008
- [MAINWA02] *"Wireless Sensor Networks for Habitat Monitoring"*  
A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson  
Proc. of Int. Workshop on Wireless Sensor Networks and Applications, Atlanta, USA, 2002
- [MANJESH01] *"TEEN: a routing protocol for enhanced efficiency in wireless sensor networks"*  
A. Manjeshwar, D. P. Agarwal  
Proc. International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, IEEE Computer Society San Francisco, USA, 2001
- [MANJESH02] *"APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks"*  
A. Manjeshwar, D. P. Agarwal  
Proceedings International Parallel and Distributed Processing Symposium, IPDPS, 2002

- [MANZANO08] *“Performance Study of Non-beaconed and Beacon-Enabled Modes in IEEE 802.15.4 Under Bluetooth Interference”*  
M. Manzano, A. Bonastre, J.V. Capella  
Proc. of International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Valencia, Spain, Octubre 2008
- [MICAZ04] *“MICAz wireless measurement system”*  
Crossbow Technology Inc.  
URL: <http://www.xbow.com>, Junio 2004
- [MIT03] *“10 Emerging Technologies That Will Change the World”*  
Technology Review (MIT), Febrero 2003. URL:  
[www.technologyreview.com/read\\_article.aspx?id=13060&ch=infotech](http://www.technologyreview.com/read_article.aspx?id=13060&ch=infotech)
- [MITTON08] *“Hector is an Energy efficient Tree-based Optimized Routing protocol for wireless networks”*  
N. Mitton, T. Razafindralambo, D. Simplot-Ryl, I. Stojmenovic  
Proceedings of International Conference on Mobile Ad-hoc and Sensor Networks (MSN'08), Wuhan, China, 2008
- [MITUAMP] *“MIT  $\mu$ AMPS project ns2 code extensions”*  
URL: <http://mtlweb.mit.edu/researchgroups/icsystems/uamps/>
- [MOHAJER09] *“Tree based energy efficient and congestion aware routing protocol for wireless sensor networks”*  
A. H. Mohajerzadeh, M. H. Yaghmaee, Z. Eskandari  
IEEE ICCS, China, pp. 1707–1711, 2009
- [MURATA94] *“Performance evaluation of genetic algorithms for flowshop scheduling problems”*  
T. Murata, H. Ishibuchi  
Proceedings of the 1st IEEE Conf. Evolutionary Computation, 1994
- [MURUGA05] *“A centralized energy-efficient routing protocol for wireless sensor networks”*  
S. D. Muruganathan, D.C.F. Ma, P.I. Bhasin *et al.*  
IEEE Communications Magazine, Vol. 43, Nº 3, pp. 8-13, 2005
- [NS09] *UCB/LBNL/VINT Network Simulator - ns (Version 2.34)*  
Released Jun 17 2009  
URL: <http://www.isi.edu/nsnam/ns/ns-2 ns-2.34>
- [NATSEMC00] *National Semiconductor Corporation, LMX3162 Single Chip Radio Transceiver, Evaluation Notes and Datasheet, March 2000.*

- [NAVARRO09] *"Evaluación Y Comparación De Mecanismos Para Enrutamiento Eficiente En Redes De Sensores Inalámbricos"*  
Navarro Sebastián, Jorge Joaquín  
*Proyecto Final de Carrera*. Dirigido por Juan Vicente Capella Hernández  
E.T.S. Ingeniería Informática. Julio 2009
- [PETROVA06] *"Performance Study of IEEE 802.15.4 Using Measurements and Simulations"*  
M. Petrova, J. Riihijärvi, P. Mähönen, S. LaBella  
Wireless Communications and Networking Conference, Vol. 1, pp. 487-492,  
Las Vegas, NV, USA, 2006
- [PUCCIN06] *"Multipath Fading in Wireless Sensor Networks: Measurements and Interpretation"*  
D. Puccinelli, M. Haenggi  
IWCMC'06, Vancouver, Canadá, Julio 2006
- [RAICU04] *"e3D: An Energy-Efficient Routing Algorithm for Wireless Sensor Networks"*  
I. Raicu, L. Schwiebert, S. Fowler, S. K.S. Gupta  
Proceedings of the Intelligent Sensors, Sensor Networks and Information  
Processing Conference, Melbourne, Australia, 2004
- [RAMACHA07] *"Analysis of the Contention Access Period of IEEE 802.15.4 MAC"*  
I. Ramachandran, A. K. Das, S. Roy  
ACM Transactions on Sensor Networks (TOSN), Vol. 3, No. 1, 2007
- [RODOPLU99] *"Minimum Energy Mobile Wireless Networks"*  
V. Rodoplu, T. H. Meng  
IEEE JSAC, 1999
- [RUNES] *RUNES Project (UE)* URL: <http://www.ist-runes.org>
- [SADAGOP03] *"The ACQUIRE Mechanism for Efficient Querying in Sensor Networks"*  
N. Sadagopan, B. Krishnamachari, A. Helmy  
Elsevier journal on Ad Hoc Networks, 2003
- [SAEZ07] *"Quick And Dirty Routing Protocol (Diseño e Implementación de Un Algoritmo de Enrutamiento para Redes de Sensores Inalámbricos)"*  
Sáez Incertis, Abel  
*Tesis de máster*. Dirigida por Juan V. Capella y Alberto Bonastre  
Dpto. Informática de Sistemas y Computadores (DISCA). Diciembre 2007
- [SAJJANH07] *"Distributive Energy Efficient Adaptive Clustering Protocol for Wireless Sensor Networks"*  
U. Sajjanhar, P. Mitra  
Proceedings of the 2007 International Conference on Mobile Data  
Management, pp.: 326-330, Mannheim, Germany, 2007

- [SANCHEZ09] *“Estudio De Los Problemas Y Mecanismos De Seguridad En Redes Inalámbricas De Sensores Y Su Aplicación A Protocolos De Nivel De Enrutamiento”*  
Sánchez Martínez, Alberto  
*Proyecto Final de Carrera*. Dirigido por Juan Vicente Capella Hernández  
E.T.S. Ingeniería Informática. Diciembre 2009
- [SAVVIDE01] *“Dynamic fine-grained localization in Ad-Hoc networks of sensors”*  
A. Savvides, C.-C. Han, M. Srivastava  
Proceedings of the Seventh ACM Annual International Conference on Mobile Computing and Networking, 2001
- [SCHEIB07] *“Unplugged but connected - Design and Implementation of a Truly Wireless Real-Time Sensor/Actuator Interface”*.  
G. Scheible, D. Dzung, J. Endresen, J.-E. Frey  
IEEE Industrial Electronics Magazine, Vol. 1, Num. 2, pp. 25-34, 2007
- [SCHURG01] *“Energy Efficient Routing In Wireless Sensor Networks”*  
C. Schurgers, M. B. Srivastava  
Military Communications Conference, MILCOM 2001. IEEE Communications for Network-Centric Operations: Creating the Information Force, 2001
- [SCOTT96] *“Routing and Channel Assignment for Low Power Transmission in PCS”*  
K. Scott and N. Bambos  
Proceedings of the 5th IEEE Int. Conf. on Universal Personal Communications, Vol. 2, pp.: 498–502, 1996
- [SECURET] *Secure Time Synchronization In Sensor Networks Project* (Dinamarca)  
URL: <http://imm.dtu.dk/sens/sectymesync.html>
- [SERVETTO2] *“Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale WSNs”*  
S. D. Servetto, G. Barrenechea  
Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pp. 12-21, Atlanta, Georgia, USA, 2002
- [SHAH02] *“Energy Aware Routing for Low Energy Ad Hoc Sensor Networks”*  
R. C. Shah, J. M. Rabaey  
Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Orlando, USA, 2002.
- [SHIH01] *“Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks”*  
E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan  
Proc. of ACM MobiCom’01, Rome, Italy. pp. 272–286, 2001

- [SHIN07] *"Packet Error Rate Analysis of ZigBee Under WLAN and Bluetooth Interferences"*  
S. Y. Shin, H. S. Park, S. Choi, W. H. Kwon  
IEEE Transactions on Wireless Communications, Vol. 6, No. 8, 2007
- [SIKORA05] *"Coexistence of IEEE802.15.4 with other Systems in the 2.4 GHz-ISM-Band"*  
A. Sikora, V. F. Groza  
IMTC 2005, Ottawa, Canadá, Mayo 2005
- [SIMPLTI08] *"Introduction to SimplicTI"*  
Texas Instruments, 2008
- [SOBEIH05] *"J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks"*  
A. Sobeih, W.-P. Chen, J. C. Hou, L.-C. Kung, N. Li, H. Lim, H.-Y. Tyan, H. Zhang  
Proceedings of the 38th annual Symposium on Simulation, p.175-187  
San Diego, California, USA, 2005
- [SOHRABIO0] *"Protocols for Self-Organization of a Wireless sensor Network"*  
K. Sahrabi, J. Gao, V. Ailawadhi, G. Pottie  
IEEE Pers. Commun., vol. 7, no. 5, pp. 16-27, Octubre 2000
- [SPOT] *SPOT Project* (Dinamarca) URL: <http://imm.dtu.dk/sens/secloc.html>
- [SUBRAMA01] *"An Architecture for Building Self-Configurable Systems"*  
L. Subramanian, R. H. Katz  
Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, pp. 63-73. Boston, Massachusetts, 2001
- [TANG07] *"Channel Characterization and Link Quality Assessment of IEEE 802.15.4-Compliant Radio for Factory Environments"*  
L. Tang, K-C Wang, Y. Huang, F. Gu  
IEEE Transactions on Industrial Informatics, Vol. 3, No. 2, Mayo 2007
- [TELOS04] *"Telos (Rev B) Datasheet"*  
Moteiv Corporation  
URL: <http://www.moteiv.com>, Diciembre 2004
- [TYAN09] *"Design, Realization and Evaluation of a Component-based, Compositional Network Simulation Environment"*  
H.-Y. Tyan, A. Sobeih, J. C. Hou  
Simulation Journal, Vol. 85, No. 3, pp.: 159-181, 2009
- [WIFI09] IEEE 802.11n™ IEEE Standard for Information technology. Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 2009

- [WISENTS] *WiSeNts project* (UE) URL: <http://www.embedded-wisents.org>
- [WOO01] *"A Transmission Control Scheme for Media Access in Sensor Networks"*  
A. Woo, D. Culler  
Proceedings of the 7th annual international conference on Mobile computing and networking , pp. 221-235, Rome, Italy, 2001
- [XU01] *"Geography-informed Energy Conservation for Ad-hoc Routing"*  
Y. Xu, J. Heidemann, D. Estrin  
Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking, pp. 70-84. Roma, Italia, 2001
- [YANG09] *"An energy-balancing unequal clustering protocol for wireless sensor networks"*  
J. Yang, D. Zhang  
Journal of Inform. Technol., Vol. 8, No. 1: pp. 57-63, 2009
- [YAO02] *"The cougar approach to in-network query processing in sensor networks"*  
Y. Yao, J. Gehrke  
ACM SIGMOD Record, Vol. 31, N° 3, 2002
- [YE01] *"A scalable solution to minimum cost forwarding in large sensor networks"*  
F. Ye, A. Chen, S. Liu, L. Zhang  
Proceedings of the tenth International Conference on Computer Communications and Networks (ICCCN), 2001
- [YE02] *"A Two-Tier Data Dissemination Model for Largescale Wireless Sensor Networks"*  
F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang  
Proceedings of the 8th annual international conference on Mobile computing and networking , pp. 148-159. Atlanta, Georgia, USA, 2002
- [YIN08] *"An Energy-Efficient Routing Algorithm for Wireless Sensor Networks"*  
G. Yin, G. Yang, W. Yang, B. Zhang, W. Jin  
International Conference on Internet Computing in Science and Engineering  
pp. 181-186. ICICSE '08, Harbin, China, 2008
- [YOO09] *"A scalable publish/subscribe system for large mobile ad hoc networks"*  
S. Yoo, J.H. Son, M.H. Kim  
Journal of Systems and Software. Vol. 82 , N° 7, 2009.
- [YOUNIS06] *"Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges"*  
O. Younis, M. Krunz, S. Ramasubramanian,  
IEEE Network, pp. 20-25, Mayo-Junio 2006

- [YU01] *“Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks”*  
Y. Yu, D. Estrin, R. Govindan  
UCLA Computer Science Department Technical Report, 2001
- [YU06] *“GROUP: a Grid-clustering Routing Protocol for Wireless Sensor Networks”*  
L. Yu, N. Wang, W. Zhang, C. Zheng  
Proc. International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Wuhan, China, 2006
- [ZIGBEE07] Zigbee Alliance - Zigbee specification Q4/2007  
URL: [www.zigbee.org/Products/DownloadZigBeeTechnicalDocuments.aspx](http://www.zigbee.org/Products/DownloadZigBeeTechnicalDocuments.aspx)

## Agradecimientos

Quisiera aprovechar esta ocasión para agradecer el apoyo, comprensión y amistad de todos los que me han acompañado en esta travesía, no obstante resulta imposible citar de forma exhaustiva a todos los que ha aportado algo a esta tesis.

A mi familia, Papá, Isa y Emi, porque son como un mástil al que aferrarse en la tormenta. Porque tanto en los buenos momentos como en los que no lo son tanto siempre están ahí mostrándome su cariño, comprensión y entrega incondicional.

A mi madre, porque su luz me guía como un faro a los marineros para que arriben a buen puerto.

A Susana, por su paciencia, constancia y amor, por el tiempo que me ha concedido. Sin su apoyo este trabajo hubiera resultado mucho más arduo.

A mis amigos, que siempre me han prestado un gran apoyo moral y humano, necesarios en los momentos difíciles.

Agradecer a mi director de tesis, Alberto, la dedicación y apoyo que me ha brindado en este trabajo, por el respeto a mis ideas y por la dirección y el rigor que ha facilitado a las mismas. Gracias por la confianza ofrecida desde siempre.

Agradecer también a Rafa su inestimable ayuda y consejo. Gracias por transmitirme su entusiasmo y palabras de aliento desde mi llegada al grupo.

A Juanjo, fuente de conocimientos e ideas, agradecerle su confianza así como todo el apoyo prestado.

A mis compañeros de departamento e instituto, por su ayuda y por los buenos momentos compartiendo risas en los pasillos y cafés.

Y en general, a todos los que de una forma u otra han colaborado en el desarrollo de esta tesis, a todos ellos, muchas gracias.