

Clustering dinámico para tiempo de encendido mínimo en redes de sensores inalámbricas (CLUDITEM)

Autora: Ing. Rosa Marina Corti

Director: Dr. Joan Oliver Malagelada

Director: Dr. Gustavo Rossi

*Tesis presentada para obtener el grado de Magister en:
Ingeniería del Software*

Facultad de Informática
Universidad Nacional de La Plata

2011

Corti , Rosa Marina

Clustering dinámico para tiempo de encendido mínimo en redes de sensores inalámbricas CLUDITEM. - 1a ed. - La Plata : Universidad Nacional de La Plata, 2012. E-Book.

ISBN 978-950-34-0851-3

1. Sensores. 2. Diseño de Redes. 3. Ingeniería de Software. I. Título
CDD 005.3

Fecha de catalogación: 31/05/2012

Para José, Cecilia y Ariel

Agradecimientos

A mi familia, por su paciencia y amor, ya que en todos estos años me sostuvo y me alentó a continuar con el trabajo pese a las muchas dificultades encontradas en el camino.

A todos los integrantes del grupo de Redes Inalámbricas de Sensores Inteligentes, por su apoyo, sugerencias y colaboración, cada uno desde su área de conocimiento; aprendí mucho y disfruté del quehacer conjunto.

A Enrique y Estela, ya que sin su ayuda y sostén esto no hubiera sido posible.

A Estela otra vez, que además de ser una excelente compañera de tareas es una amiga como pocas veces se encuentra en la vida.

Al Dr. Joan Oliver, por sus invaluable consejos, la cuidadosa revisión de mi trabajo, y el apoyo brindado para seguir adelante.

Finalmente a los profesores de la Maestría en Ingeniería del Software, por todos los conocimientos recibidos que enriquecieron mi labor cotidiana, y a la Secretaría de Posgrado de la UNLP por su amabilidad y diligencia.

Índice General

ÍNDICE GENERAL	7
ÍNDICE DE ILUSTRACIONES.....	10
ÍNDICE DE TABLAS.....	12
CAPÍTULO 1. INTRODUCCIÓN	13
1.1 REDES INALÁMBRICAS DE SENSORES	13
1.2 OBJETIVOS Y ALCANCE DE LA TESIS	17
1.3 ORGANIZACIÓN DEL TRABAJO	18
CAPÍTULO 2. FUNDAMENTOS TEÓRICOS	21
2.1 ESPECIFICACIONES PARA EL DISEÑO DE REDES DE SENSORES	21
2.2 ALGORITMOS DE ENCAMINAMIENTO	23
2.2.1 <i>Control de topología</i>	24
2.3 CLUSTERING.	26
2.3.1 <i>Atributos de las técnicas de clustering</i>	27
2.3.2 <i>Técnicas de clustering y objetivos de diseño de la red.</i>	29
2.4 AGREGACIÓN DE DATOS	32
2.4.1 <i>Encaminamiento y agregación</i>	33
2.4.2 <i>Funciones de agregación</i>	34
2.4.3 <i>Técnicas de agregación periódicas</i>	36
CAPÍTULO 3. REQUERIMIENTOS PARA REDES DE SENSORES	40
3.1 LAS APLICACIONES DE SUPERVISIÓN AMBIENTAL	42
3.2 LINEAMIENTOS DE DISEÑO PARA UN ALGORITMO DE ENCAMINAMIENTO	44

CAPÍTULO 4. CLUDITEM.....	47
4.1 CARACTERÍSTICAS GENERALES	47
4.2 TRABAJOS RELACIONADOS	48
4.3 ETAPAS DE FUNCIONAMIENTO.....	50
4.4 ETAPA DE DEFINICIÓN DEL ÁRBOL DE ENCAMINAMIENTO	51
4.4.1 <i>Definición de clusters</i>	52
4.4.2 <i>Rotación de cabeceras</i>	56
4.4.3 <i>Análisis de tiempos para la definición de clusters</i>	57
4.4.4 <i>Definición del árbol de cabeceras</i>	59
4.4.5 <i>Análisis de tiempos para la definición del árbol de CH</i>	61
4.5 ETAPA DE ENVÍO DE DATOS	62
4.5.1 <i>Envío de datos intracluster</i>	63
4.5.2 <i>Análisis de tiempos para el envío de datos intracluster</i>	64
4.5.3 <i>Envío de datos intercluster</i>	65
4.5.4 <i>Análisis de tiempos para el envío de datos intercluster</i>	67
4.5.5 <i>Características de la agregación de datos</i>	67
4.6 MODELO DE ENERGÍA	69
4.6.1 <i>Potencia de transmisión intracluster</i>	70
4.6.2 <i>Potencia de transmisión intercluster</i>	71
4.6.3 <i>Cálculo del momento de rotación de CH</i>	72
CAPÍTULO 5. SIMULACIÓN DEL FUNCIONAMIENTO DE CLUDITEM	74
5.1 EL SIMULADOR NS2.....	74
5.2 IMPLEMENTACIÓN DE CLUDITEM EN NS2	74
5.2.1 <i>Definición de parámetros del modelo de energía</i>	78
5.2.2 <i>Temporización del funcionamiento</i>	79
5.3 ESCENARIO DE SIMULACIÓN	81
5.4 EVALUACIÓN DE CLUDITEM.....	82

5.4.1	<i>Métricas para evaluar la gestión de la estructura de clusters.....</i>	83
5.4.2	<i>Comparación de CLUDITEM con otro algoritmo de encaminamiento.....</i>	83
5.4.3	<i>Métricas relacionadas con la QoS.....</i>	85
5.4.4	<i>Métricas referidas al consumo de energía.....</i>	85
5.5	ANÁLISIS DE LA DEFINICIÓN DE LA ESTRUCTURA DE CLUSTERS	86
5.6	COMPARACIÓN DE CLUDITEM CON EL ALGORITMO SIN GRILLA	90
5.7	EVALUACIÓN DE ESQUEMAS TDMA	93
5.8	RESULTADOS REFERIDOS AL MODELO DE ENERGÍA: TIEMPO DE VIDA DE LA RED.....	97
CAPÍTULO 6. PROTOTIPO DE NODO CLUDITEM.		104
6.1	PLATAFORMAS HARDWARE PARA REDES DE SENSORES	104
6.1.1	<i>Plataformas reconfigurables.....</i>	105
6.2	ARQUITECTURA DEL PROTOTIPO DE NODO	108
6.3	EL MÓDULO SOFTWARE	110
6.3.1	<i>Implementación con un procesador de 8 bits</i>	111
6.3.2	<i>Implementación con un procesador de 32 bits</i>	115
6.4	ENSAYOS CON EL PROTOTIPO INALÁMBRICO BASADO EN MICROBLAZE	116
6.4.1	<i>Detalle de los módulos implementados para las pruebas.....</i>	116
6.4.2	<i>Pruebas realizadas.....</i>	119
6.4.3	<i>Resultados obtenidos</i>	120
CONCLUSIONES.....		123
TRABAJO FUTURO		125
PUBLICACIONES RELACIONADAS CON LA TESIS		127
NOMENCLATURA.....		129
BIBLIOGRAFÍA.....		131

Índice de ilustraciones

FIGURA 1. ESQUEMA DE UNA RED DE SENSORES CON UNA ESTACIÓN BASE	14
FIGURA 2. ESQUEMA DE UN NODO SENSOR.....	15
FIGURA 3. RELACIÓN ENTRE LOS REQUERIMIENTOS DE LAS APLICACIONES ELEGIDAS Y LAS CARACTERÍSTICAS DEL ALGORITMO.	45
FIGURA 4. ETAPAS DE FUNCIONAMIENTO DE CLUDITEM DURANTE LA VIDA ÚTIL DE LA RED.	51
FIGURA 5. USO DE UNA CUADRÍCULA VIRTUAL PARA LA DEFINICIÓN DE CLUSTERS	52
FIGURA 6. MENSAJE PARA LA DEFINICIÓN DE LOS <i>CLUSTERS</i> (EC).....	53
FIGURA 8. CONDICIONES DE PROCESAMIENTO DE MENSAJES EC.....	55
FIGURA 9. TIEMPOS LOCALES A LOS NODOS EN LA FASE DE DEFINICIÓN DE <i>CLUSTERS</i>	58
FIGURA 10. FORMATO DEL MENSAJE ACH	59
FIGURA 11. COMPORTAMIENTO DE UN NODO DURANTE EL ARMADO DEL ÁRBOL DE CH	60
FIGURA 12. FUNCIONAMIENTO DE CLUDITEM. DETALLE DE LAS FASES DE ENVÍO DE DATOS.....	62
FIGURA 13. FORMATO DEL MENSAJE DE DATOS.	63
FIGURA 14. COMPORTAMIENTO DE UN NODO EN LA FASE DE ENVÍO DE DATOS INTRACLUSTER	63
FIGURA 15. FORMATO DEL MENSAJE DE DATOS AGREGADOS.....	65
FIGURA 16. COMPORTAMIENTO DE UN NODO EN LA FASE DE ENVÍO DE DATOS INTERCLUSTER.....	66
FIGURA 17. RADIO DE TRANSMISIÓN INTRACLUSTER.....	71
FIGURA 18. RADIO DE TRANSMISIÓN INTERCLUSTER	71
FIGURA 19. ESQUEMA DE LA IMPLEMENTACIÓN DEL PROTOCOLO DE COMUNICACIÓN	75
FIGURA 20. CONFIGURACIÓN DE LAS CAPAS DEL PROTOCOLO DE COMUNICACIONES EN UN NODO CLUDITEM.....	76
FIGURA 21. PARÁMETROS DEL AGENTE CLUDITEM	77
FIGURA 22. CONFIGURACIÓN DE LA ESTACIÓN BASE	77
FIGURA 23. DEFINICIÓN DEL ESCENARIO DE SIMULACIÓN.....	81
FIGURA 24. EJEMPLO DE DEFINICIÓN DE LA ESTRUCTURA DE <i>CLUSTERS</i>	87
FIGURA 25. ANÁLISIS DE LA MÁXIMA PROFUNDIDAD DE LOS <i>CLUSTERS</i>	88
FIGURA 26. ANÁLISIS DEL NÚMERO DE POSTULANTES POR CELDA.	89

FIGURA 27. EJEMPLO DE ESTRUCTURA DE CLUSTERS DEL ALGORITMO SIN GRILLA.....	91
FIGURA 28. ESQUEMA INTRACLUSTER DE 32 SLOTS PARA TRANSMISIÓN DE DATOS	93
FIGURA 29. MÓDULOS DE ALTO NIVEL DE LAS UNIDADES DE PROCESAMIENTO Y COMUNICACIONES.....	110
FIGURA 30. CONEXIÓN ENTRE LOS NODOS NTX Y RTX EN EL PROTOTIPO CABLEADO.....	112
FIGURA 31. DIAGRAMA SIMPLIFICADO DE LOS NODOS UTILIZANDO PICOBLAZE: NODO TRANSMISOR NTX.....	112
FIGURA 32. DIAGRAMA SIMPLIFICADO DE LOS NODOS UTILIZANDO PICOBLAZE: NODO RECEPTOR NRX.....	113
FIGURA 33. IMAGEN DEL PROTOTIPO DE NODO SENSOR IMPLEMENTADO.....	116
FIGURA 34. ESQUEMA DE LA ARQUITECTURA DEL NODO UTILIZANDO EL PROCESADOR MICROBLAZE	117
FIGURA 35. VISUALIZACIÓN DE SEÑALES: RECEPCIÓN DE LA POSTULACIÓN DEL NODO 1 POR PARTE DEL NODO 3.....	120
FIGURA 36. VISUALIZACIÓN DE SEÑALES: POSTULACIÓN DEL NODO 3	121

Índice de tablas

TABLA 1. VALORES DE LOS PARÁMETROS DEL MODELO DE RADIO Y CONSUMO EN LOS DISTINTOS ESTADOS DEL TRANSECTOR	79
TABLA 2. COMPARACIÓN DEL FUNCIONAMIENTO DE CLUDITEM VS. SIN GRILLA	92
TABLA 3. RESUMEN RESULTADOS DE SIMULACIÓN PARA LOS ESQUEMAS TDMA INTRACLUSTER	95
TABLA 4. SALIDA DE SERVICIO DE LA RED CON ESQUEMA DE 9 SLOTS DE ENVÍO DE AGREGADOS.....	96
TABLA 5. PROMEDIO DE MENSAJES POR RONDA CON 32 SLOTS DE ENVÍO DE DATOS Y 18 SLOTS DE ENVÍO DE AGREGADOS.....	97
TABLA 6. VALORES PARA LA TEMPORIZACIÓN DEL FUNCIONAMIENTO DEL ALGORITMO EN SIMULACIÓN.....	98
TABLA 7. CONSUMOS DE ACUERDO AL ROL DE LOS NODOS Y EL TIPO DE RONDA.....	99
TABLA 8. RONDA DE SALIDA DE SERVICIO DE LA RED PARA DISTINTOS VALORES DE X.....	99
TABLA 9. ENERGÍA RESIDUAL EN LA RED CUANDO SALE DE SERVICIO POR INCUMPLIMIENTO DE QoS.....	100
TABLA 10. GRUPOS DE NODOS CLASIFICADOS POR ENERGÍA RESIDUAL CUANDO SALE DE SERVICIO LA RED.....	101

Capítulo 1. Introducción

1.1 *Redes inalámbricas de sensores*

La medición de variables del medio es fundamental para la supervisión y control de ambientes y actividades de la más diversa índole. Resulta de gran utilidad en aplicaciones industriales, médicas, agrícolas, de preservación del medio natural o creación de ambientes inteligentes, entre otras. En muchas de las aplicaciones mencionadas, el seguimiento de las variables de interés debe realizarse en locaciones distantes u hostiles que hacen muy difícil el cableado y la atención periódica de los dispositivos de medición. Por estos motivos se utilizan las redes inalámbricas de sensores inteligentes (RISI) para obtener los datos necesarios [2] [65] [84].

Las RISI son un tipo particular de redes Ad-Hoc, constituidas por nodos que se distribuyen en el área a estudiar, cuyo objetivo es realizar mediciones sobre el fenómeno de interés y transmitirlos hacia una o varias estaciones base también conocidas como nodos *sink*. Los nodos de las redes de sensores son dispositivos capaces de adquirir y procesar los datos necesarios, para luego colaborar con sus vecinos con el fin de hacerlos llegar hasta el nodo *sink*. En la figura 1 puede verse el esquema de una red de sensores con una estación base ubicada fuera del área bajo estudio.

Los ambientes en que se despliegan las RISI y el número de dispositivos que se distribuyen para constituir la red, obligan a diseñar nodos pequeños y económicos, restringidos en recursos. Estos nodos pueden fallar por agotamiento de sus reservas de energía o daño físico, y la comunicación inalámbrica puede verse perturbada por interferencias en el medio. Además, si bien en la mayoría de las aplicaciones las RISI son estáticas, algunas de ellas estudian fenómenos en movimiento o pueden requerir cierta movilidad de los dispositivos sensores o de la estación base. Por lo tanto, estas redes necesitan auto-organizarse para adaptarse a topologías cambiantes, y deben trabajar bajo fuertes restricciones de energía, tratando de maximizar su tiempo de vida útil [2] [4] [15] [65].

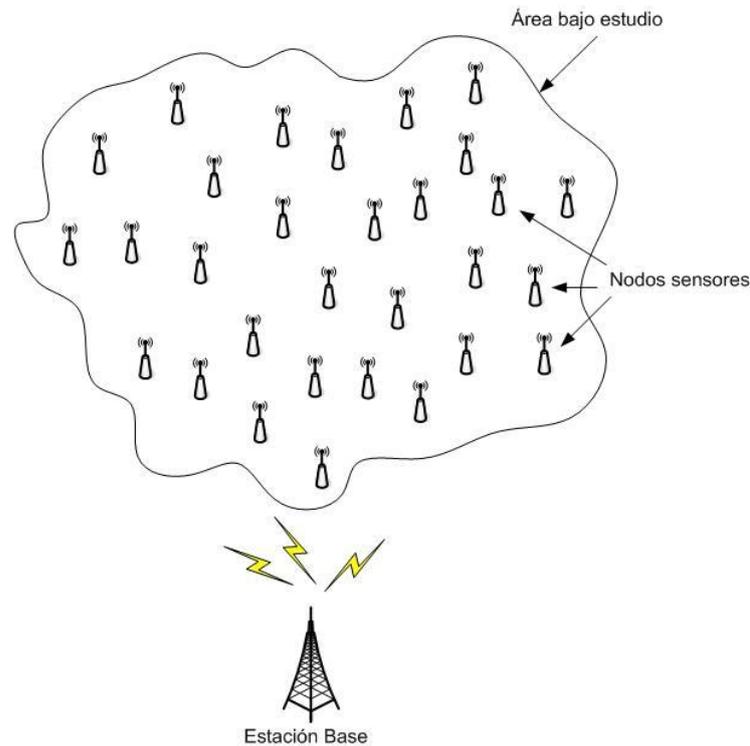


Figura 1. Esquema de una red de sensores con una estación base

Una RISI realiza dos actividades fundamentales para hacer llegar hasta la estación base información relevante para la aplicación:

- Adquisición y procesamiento de la información.
- Transmisión de la misma hacia la/las estación/es base.

El nodo sensor o mote, es la unidad básica de una RISI, y está constituido por cinco módulos diferenciados, como se muestra en la figura 2 [4] [86].

- *Unidad de adquisición de datos:* Incluye los elementos capaces de obtener las muestras de las variables de interés. Existe una gran variedad de estos dispositivos que pueden ser integrados al nodo de acuerdo con las necesidades de la aplicación.
- *Interfaz de adaptación de señales:* Las muestras adquiridas deben ser adaptadas para su posterior procesamiento. Esta interfaz debe entregar los datos al módulo procesador, en un formato compatible con sus características.

- *Unidad de proceso:* Es la encargada de realizar el tratamiento de la información que llega al nodo, pudiendo incluir por este motivo unidades de almacenamiento de datos. Además, suelen estar a su cargo otras tareas como la elaboración y control de los mensajes ligados a la definición del encaminamiento y al envío de mediciones.
- *Unidad de comunicaciones:* Se ocupa de transmitir la información procesada con el objetivo de alcanzar la estación base.
- *Unidad de potencia:* Provee energía para el funcionamiento del dispositivo.

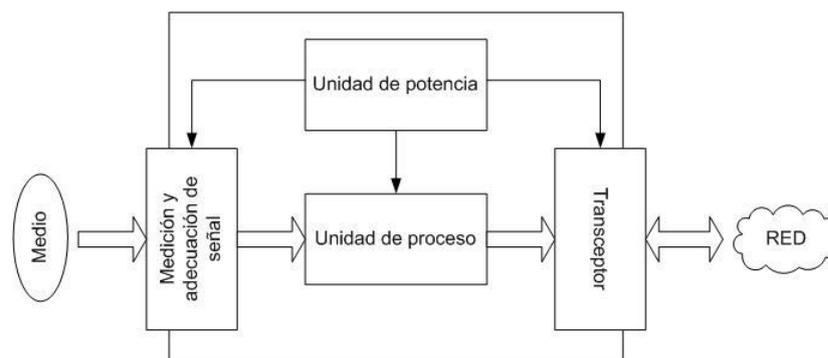


Figura 2. Esquema de un nodo sensor

Estas unidades colaboran entre sí e intercambian información de forma que el nodo cumpla con sus tareas en la red. Además, de acuerdo a la actividad que realice cada dispositivo, puede estar en uno de los siguientes estados: transmisión, recepción, en escucha y apagado. En transmisión, el nodo ha ganado acceso al medio y envía información a la red. En recepción, detecta la llegada de un mensaje enviado por un vecino, en estado de escucha está analizando el medio y cuando se apaga, no realiza actividad alguna consumiendo el mínimo de energía posible. Por lo tanto, cuanto más tiempo permanezca un nodo en este último estado, consumirá menos y podrá participar de las actividades de la red por más tiempo [8]. En este sentido, el tiempo de vida de la red es una de las métricas más importantes para evaluar el desempeño de las RISI. Para utilizar esta métrica, es necesario fijar cuando se considera que la red ya no es útil y, al respecto, se pueden establecer diversos criterios entre los que se pueden citar: cuando muere el primer nodo, cuando la pérdida de mensajes es superior a un porcentaje establecido antes de poner en funcionamiento la red, o cuando alguna zona del área a supervisar queda desconectada [4].

El tráfico de comunicación en las RISI puede ser muy variado dependiendo de la aplicación, y puede cambiar de una transmisión escasa o casi nula de datos a un flujo importante de información por distintas causas. Por lo tanto, el tráfico puede estar dirigido por: tiempo, eventos, requerimientos o un método híbrido que englobe parte de cada uno [33]. Respecto al tipo de comunicación inalámbrica entre los nodos, existen varias alternativas, pero la comunicación por radiofrecuencia es la más utilizada.

Finalmente es importante destacar que las RISI tienen una gama de aplicaciones muy amplia, ya que son capaces de recolectar y transmitir información relativa a variables físicas muy diversas. Se pueden destacar entre otras [26] [84]:

- *Aplicaciones Ambientales:* Son variadas ya que la recolección de información sobre el medioambiente y los fenómenos que alberga permiten realizar tareas de prevención, control y toma de decisiones para su gestión y preservación. Se puede mencionar la detección de fenómenos destructivos como incendios o inundaciones, tareas relacionadas con agricultura de precisión, la supervisión de niveles de polución, el seguimiento de poblaciones de animales o el registro de variables de distinta naturaleza para controlar su evolución espacio/temporal.
- *Aplicaciones de supervivencia y asistencia en desastres:* Los sensores se distribuyen en la zona de la catástrofe (muchas veces inaccesible) y resultan de gran utilidad para estimar daños y detectar sobrevivientes apoyando las tareas de rescate y asistencia.
- *Aplicaciones a la Salud:* Pueden mencionarse el monitoreo de pacientes, actividades ligadas al diagnóstico o seguimiento de personal en grandes instituciones.
- *Aplicaciones Domésticas:* Permiten a los dueños de las fincas realizar un control remoto sobre las condiciones de sus viviendas.
- *Aplicaciones Militares:* Tareas relacionadas con C4ISRT (Comando, Control, Comunicaciones, Cómputo, Inteligencia, Supervivencia y Reconocimiento).
- *Aplicaciones Comerciales / Industriales:* Se pueden mencionar el control ambiental en grandes edificios, gestión de inventario, detección y seguimiento de vehículos, control de robots o supervisión de productos.

En resumen, se puede decir que las RISI tienen características muy específicas respecto a bajo consumo de energía, ancho de banda estrecho y nodos de bajo costo [15]. Además, dependiendo del problema a resolver, es habitual que incorporen requerimientos referidos a topologías cambiantes, extensión del área de cobertura y calidad de servicio. Estas características de las RISI han llevado a encarar soluciones específicas para el área, ya que los desarrollos existentes para redes convencionales no se adecúan a los requerimientos planteados [9] [65] [84]. El diseño de la red debe centrarse en las necesidades de la aplicación, estableciendo prioridades y realizando un cuidadoso balance para darle el peso adecuado a cada una. El éxito de la solución dependerá de un minucioso análisis de las condiciones asociadas para lograr una especificación de requerimientos realista y factible de encarar.

1.2 *Objetivos y alcance de la tesis*

La autora de la tesis integra el grupo de investigación y desarrollo GRISI (Grupo de Redes Inalámbricas de Sensores Inteligentes), del Departamento de Sistemas e Informática de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura de la Universidad Nacional de Rosario (DSI/FCEIA/UNR). La tesis aquí presentada se enmarca dentro del trabajo que lleva adelante el grupo antes mencionado y se desarrolló a partir de los siguientes objetivos:

1. *Definir un conjunto de requerimientos comunes para redes de sensores orientadas hacia aplicaciones de supervisión ambiental.*
2. *Definir métricas basadas en la especificación de requerimientos utilizando heurísticas del dominio, para analizar la viabilidad de establecer parámetros en el diseño de un algoritmo de encaminamiento. Dichos parámetros están orientados a adaptar el funcionamiento a las características de la gama de aplicaciones seleccionadas en el dominio elegido.*
3. *Diseñar un algoritmo de encaminamiento jerárquico distribuido para minimizar el consumo de energía en los nodos de una red de sensores, respetando las métricas definidas. Los aspectos más destacados del mismo son:*
 - a. *Balancear la carga de trabajo entre los nodos de la red rotando las cabeceras de cluster (CH) para evitar su agotamiento prematuro.*

- b. Elegir los CH en la zona de trabajo utilizando una grilla virtual para obtener aproximadamente una cabecera en cada cuadro de la misma.*
 - c. Apagar los transceptores de los nodos el mayor tiempo posible.*
 - d. Incorporar parámetros de diseño que permitan adaptar su funcionamiento de acuerdo con las métricas definidas.*
4. *Diseñar un prototipo de nodo sensor que soporte el algoritmo desarrollado, utilizando dispositivos de lógica reconfigurable (FPGA) y módulos de comunicación inalámbrica para implementar una red constituida por un número reducido de dispositivos.*

El análisis de requerimientos se orientó hacia aplicaciones de supervisión ambiental con medición periódica de las variables de interés, cuyos valores generalmente guardan un alto grado de correlación espacio/temporal. Se trabajó sobre los requerimientos correspondientes a las redes de sensores en general y al dominio elegido en particular.

El aporte de la tesis consiste por un lado en el análisis de requerimientos para el dominio elegido, orientado a establecer los lineamientos de diseño para un algoritmo de encaminamiento y, por otro, en el desarrollo de un algoritmo capaz de adaptar su funcionamiento para un conjunto acotado de aplicaciones, cumpliendo con las especificaciones asociadas. La viabilidad de implementación de dicho algoritmo se verifica utilizando un prototipo de nodo sensor.

1.3 Organización del trabajo

El trabajo aquí detallado se estructura de la siguiente manera:

- *Capítulo 1:* Incluye una introducción general, los objetivos y el alcance de la tesis. También se detalla la organización del trabajo.
- *Capítulo 2:* Se exponen los tópicos teóricos fundamentales que sustentan el trabajo realizado. Los mismos versan sobre especificación de requerimientos en RISI, algoritmos de encaminamiento para redes de sensores, técnicas de *clustering* y métodos de agregación de datos.

- *Capítulo 3:* Se presenta una clasificación de requerimientos para redes de sensores orientadas hacia aplicaciones de supervisión ambiental, basada en el estudio del espacio de diseño de las aplicaciones y un conjunto de especificaciones concretas ligadas al dominio elegido. Se utilizan los resultados obtenidos para definir los lineamientos de diseño de un algoritmo de encaminamiento en dicho dominio de trabajo.
- *Capítulo 4:* Incluye una descripción detallada del algoritmo desarrollado. Se enumeran sus características generales, definidas en base a la especificación de requerimientos realizada en el capítulo 3, y se mencionan trabajos relacionados. Además, se analizan las etapas de trabajo de CLUDITEM, poniendo énfasis en las responsabilidades de los nodos, la estructura de los mensajes intercambiados y la temporización que dirige el funcionamiento del algoritmo. También se describe la función de agregación adoptada para las pruebas y se detallan sus características. Finalmente se estudia el modelo de energía y se obtiene analíticamente el momento óptimo para realizar la rotación de las cabeceras de *cluster*.
- *Capítulo 5:* Se mencionan las características del simulador NS2 y se dan detalles referidos a la implementación de CLUDITEM en dicho ambiente. Se define el escenario de simulación y se establecen las métricas para evaluación del desempeño del algoritmo, utilizando como base el análisis de los requerimientos de las aplicaciones de interés descrito en el capítulo 3. Se presentan los resultados obtenidos y se los analiza utilizando las métricas antes mencionadas para definir aspectos del funcionamiento y realizar los ajustes necesarios.
- *Capítulo 6:* Se resumen las características de las plataformas de *hardware* tradicionales para redes de sensores, y se enumeran algunas aplicaciones relacionadas con la utilización de dispositivos reconfigurables. Se describen las arquitecturas evaluadas para la implementación del nodo y se justifica la elección de la más adecuada. Por último se reportan las pruebas realizadas sobre la arquitectura seleccionada, informando los resultados obtenidos para verificar la viabilidad de implementación del algoritmo.

Finalmente se obtienen conclusiones respecto del trabajo realizado y se establecen líneas de desarrollo futuro para mejorar y ampliar los resultados logrados. La presentación culmina con un detalle de las publicaciones derivadas de distintas etapas de avance de la tesis y la bibliografía utilizada.

Capítulo 2. Fundamentos teóricos

2.1 Especificaciones para el diseño de redes de sensores

Las particularidades de las RISI obligan a evaluar cuidadosamente una serie de tópicos de diseño al encarar el desarrollo de un algoritmo de encaminamiento. Se trata de características propias de estas redes altamente influenciadas por el dominio de la aplicación, que resultan determinantes para el funcionamiento y el nivel de desempeño del algoritmo [9]. Muchos autores, como los que reportan sus resultados en [4], [33] y [69], han trabajado sobre las especificaciones ligadas a distintas aplicaciones, enumerando un conjunto de características a tener en cuenta cuando se aborda el desarrollo de un algoritmo de encaminamiento. Estos aspectos son fundamentales, ya que presentan una importante variabilidad cuando se analizan los diversos escenarios en los que puede desplegarse una red de sensores, a saber:

- *Distribución de los nodos:* Puede realizarse en forma manual (determinística), o aleatoria cuando los dispositivos se arrojan sobre el área a estudiar desde algún vehículo. En el primer caso, existen lazos de comunicación que se pueden establecer cuando se inicia el funcionamiento de la red, pero que luego deberán ser mantenidos frente a fallos o interferencias. En el segundo, desde el principio se hará necesario definir rutas que permitan llevar la información útil hasta la estación base.
- *Entrega de datos:* La naturaleza de las aplicaciones define la forma en que las mediciones realizadas se enviarán hacia la estación base. La modalidad de entrega de los valores adquiridos será periódica cuando la red supervisa continuamente el ambiente, y los nodos reportan sus mediciones cada vez que se cumple un período establecido. En cambio, si se necesita detectar la presencia de algún fenómeno, el funcionamiento será por eventos o por pedidos específicos diseminados por la estación base o algún nodo particular (*queries*), en cuyo caso la red deberá reaccionar frente a la presencia del evento o valor solicitado. Por último, existen casos en los que se necesitará un modelo híbrido que incluya ambos tipos de funcionamiento

- *Tipos de nodos:* Los nodos de la red pueden ser homogéneos o heterogéneos en cuanto a recursos disponibles o energía inicial. La existencia de nodos con características diferenciadas puede surgir de necesidades del esquema de control de topología adoptado, o de la aplicación abordada. En el primer caso, algunas técnicas requieren que los nodos cumplan diferentes roles asociados con distintos niveles de consumo y/o procesamiento, y puede resultar conveniente utilizar nodos heterogéneos. Desde el punto de vista de las aplicaciones, en ciertos dominios se hace necesario medir variables en distintos rangos y tiempos, con sensores muy específicos o costosos, lo que lleva a incluirlos sólo en algunos nodos de la red.
- *Escalabilidad:* El problema a resolver determinará si el algoritmo de encaminamiento debe ser capaz de trabajar con un gran número de nodos, y/o adaptarse en caso de necesitar ampliar el área de cobertura de la red.
- *Dinámica de la red:* La mayoría de las aplicaciones definen redes estáticas, pero algunas requieren movilidad de los nodos, de la estación base, o incluyen mediciones de un fenómeno que se desplaza en el área bajo estudio.
- *Calidad de servicio (QoS):* Define valores admisibles para los atributos o características de los datos entregados por la red. Las mediciones pueden o no ser críticas en el tiempo, y esto definirá la latencia aceptable. Por otro lado, se pueden exigir distintos niveles de confiabilidad de las mediciones, o umbrales aceptables de pérdida de información. Estos requerimientos pueden ser muy diversos y surgirán de las aplicaciones encaradas.
- *Balance entre consumo de energía y QoS:* Los nodos están restringidos en recursos, lo que obliga a realizar una cuidadosa gestión de los mismos. En el diseño del algoritmo se deberán considerar las necesidades de la aplicación, para obtener un adecuado balance entre la calidad de servicio solicitada y el consumo de energía asociado.
- *Tolerancia a fallas:* Los nodos pueden agotarse, sufrir daños físicos o interferencias del ambiente, que degraden su comunicación con otros dispositivos. Esto puede ocasionar desconexiones en la red y pérdida de información útil. El algoritmo deberá ser capaz de reconfigurar rutas para cumplir con los requerimientos de la aplicación.

Las consideraciones realizadas por estos autores, han sido recogidas por los desarrolladores de algoritmos para RISI, para fundamentar sus decisiones de diseño. Sin embargo, es habitual que se definan especificaciones demasiado generales y más enfocadas hacia los protocolos de comunicación utilizados que al espacio de diseño específico sobre el que se trabaja [48]. La implementación de una red de sensores, implica el desarrollo, adaptación y/o utilización de componentes de *software*, y también de la plataforma *hardware* asociada. Por estos motivos, como ya se ha demostrado extensamente, los lineamientos de diseño que se adopten deben basarse en un análisis temprano de requerimientos, para prevenir fracasos o incrementos desmedidos en los tiempos y costos del desarrollo [44]. De hecho, en los últimos años, varios autores han comenzado a trabajar sobre la aplicación de los principios de la Ingeniería del Software en el campo de las RISI, teniendo en cuenta las particularidades de estas redes [7] [48] [63]. Para ello han propuesto taxonomías de requerimientos y servicios de las RISI y han focalizado la definición de los lineamientos de diseño de cada algoritmo de encaminamiento, sobre las especificaciones de las aplicaciones apuntando a la definición de un lenguaje común para los requerimientos. Para el desarrollo de la tesis se ha adoptado el punto de vista de estos autores, y las decisiones tomadas para la definición de CLUDITEM, se basan en sus aportes, como se detalla en el capítulo 3.

2.2 Algoritmos de encaminamiento

Las RISI tienen un conjunto de características propias que las diferencian de otras redes inalámbricas, como las redes móviles *ad-hoc* o las redes de telefonía móvil. Están constituidas en general por un número elevado de nodos, restringidos en capacidad de procesamiento y energía disponible [9] [65]. Construir un esquema de direccionamiento global para la distribución de estos dispositivos no es aconsejable si la cantidad de nodos es muy grande, ya que la sobrecarga de la red puede resultar importante. Los nodos de una RISI necesitan auto-organizarse para llevar los datos hacia la estación base, con un flujo de información de “muchos a uno”, pero esto no excluye otros tipos de comunicación entre los dispositivos como “uno a uno” o “uno a muchos”. En la mayoría de las aplicaciones, los nodos son estacionarios, aunque hay casos donde ellos y/o la estación base tienen cierta movilidad. Estas redes están centradas en los datos, pero en algunos dominios es importante conocer de dónde provienen las mediciones. Son redes con altos niveles de

redundancia, debido al número de dispositivos distribuidos, y al hecho de que los datos colectados provienen de un mismo fenómeno. Por último, las redes de sensores son específicas para las aplicaciones, y por lo tanto, los requerimientos de la aplicación abordada definen las características y el tipo de funcionamiento de la RISI a diseñar. Por estos motivos, si bien algunos de los algoritmos de encaminamiento utilizados pueden aplicarse a otras redes ad-hoc, en general se diseñan especialmente para trabajar en redes inalámbricas de sensores [2] [4] [65] [84].

2.2.1 *Control de topología*

Las redes de sensores suelen incluir una importante cantidad de dispositivos, por lo que cada nodo puede comunicarse con muchos vecinos dentro de su radio de alcance. Esta característica es beneficiosa pues permite lograr una buena cobertura del área monitoreada, y provee redundancia para prevenir posibles fallos en algunos de los dispositivos de la red [2] [26] [84]. Sin embargo, también trae aparejados serios inconvenientes: se eleva la probabilidad de interferencias, existen muchas rutas potenciales que deben ser recalculadas aún frente a pequeños cambios, y los nodos pueden utilizar más energía de la necesaria para tratar de comunicarse con vecinos alejados. Por lo tanto, se han desarrollado técnicas de control de topología para tratar de obtener un balance que permita mantener parte de los beneficios, reduciendo las desventajas. Estas técnicas se basan en disminuir deliberadamente la cantidad de vecinos potenciales de cada nodo, conservando la red conectada. La forma en la que se realiza este control de topología, permite clasificar a las redes en planas o jerárquicas [42].

Las redes planas están constituidas por nodos idénticos, o sea, todos cumplen las mismas funciones referidas a la adquisición y envío de datos hacia la estación base [57]. El control de topología en esta situación, se basa en reducir la potencia de transmisión de cada nodo para limitar su alcance, y por ende, el número de potenciales vecinos. Se debe lograr un balance entre la probabilidad de encontrar vecinos útiles, que crece al aumentar la potencia de transmisión, y reducir la posibilidad de colisiones, que disminuye si la potencia de transmisión es baja [42].

Las redes jerárquicas por su parte, poseen nodos que asumen roles diferenciados. Es importante destacar, que esta diferencia se basa en la funcionalidad y no necesariamente apunta a la disponibilidad de recursos. Sin embargo, existen redes donde la heterogeneidad de roles está directamente asociada a diferentes arquitecturas de los nodos respecto a los recursos de *hardware* y disponibilidad de energía [51].

Las redes jerárquicas, se dividen en dos tipos, basadas en conjuntos dominantes y basadas en *clusters*.

- *Redes jerárquicas basadas en conjuntos dominantes:* Sea V el conjunto de todos los nodos de la red y D un subconjunto de ellos. D constituirá un conjunto dominante de nodos, o espina dorsal de la red (*backbone*), si para todos los nodos de V , se cumple que: o pertenecen a D , o son vecinos de un salto de algún nodo de D . Los nodos de la espina dorsal de la red, serán entonces los encargados de llevar la información adquirida hasta la estación base. Para que el conjunto dominante de la red resulte efectivo en términos de encaminamiento, deberá ser pequeño o incluso mínimo. Obtener este conjunto mínimo constituye un problema NP complejo. Por lo tanto, en la práctica, se recurre a soluciones aproximadas [42].
- *Redes jerárquicas basadas en clusters:* Los *clusters* son grupos de nodos con un organizador o cabecera (CH), al que reportan los miembros del grupo, conocidos como nodos comunes (NC) [42]. Formalmente, dado un grafo $G = (V, E)$, donde V son los nodos y E los enlaces entre ellos, definir *clusters* en G , consiste en establecer un conjunto de subconjuntos de nodos V_i , con $i = 1, 2, \dots, n$, de forma que

$$V = \bigcup_1^n V_i.$$

En una red basada en *clusters*, la información fluye desde los nodos comunes hacia el coordinador, y luego desde las cabeceras hasta la estación base. Cada algoritmo de encaminamiento define la modalidad y las características de la comunicación intracluster e intercluster [51].

2.3 *Clustering.*

Los algoritmos de encaminamiento para redes de sensores que aplican técnicas de *clustering*, se basan en la heurística de divide y vencerás, que acota los problemas de gestión a subconjuntos de nodos e incrementa la escalabilidad de la red [41]. Por lo tanto, el uso de *clusters* tiene un gran impacto en el desempeño general del sistema, y ha demostrado ser una forma muy eficiente de organización de la red [18] [85].

Entre los beneficios que brinda este tipo de control de topología se pueden mencionar [1]:

- Conservar el ancho de banda, ya que la comunicación intercluster queda limitada a los CH y previene el intercambio redundante de mensajes entre los nodos sensores.
- Estabilizar la topología de la red al nivel de los sensores y acotar la sobrecarga de mantenimiento, ya que los NC sólo deben enviar información a sus coordinadores y no deben preocuparse de los problemas de comunicación con la estación base.
- Habilitar a los CH para gestionar las tareas de los miembros de su *cluster* con el fin de mejorar la operación de la red. Las cabeceras pueden organizar las transmisiones de los NC reduciendo la posibilidad de pérdida de información, o lograr que los miembros de su *cluster* trabajen en la modalidad de bajo consumo aprovechando la redundancia en la cobertura del área bajo estudio.
- Realizar agregación de datos en las cabeceras de *cluster* para reducir la cantidad de transmisiones.
- Limitar la cantidad de colisiones al disminuir el número de mensajes intercambiados.

Los investigadores y profesionales que trabajan en el área de redes de sensores, han propuesto una gran cantidad de algoritmos de encaminamiento basados en *clusters*. Autores como [1], [18] y [85] los han comparado utilizando distintos criterios, y han definido taxonomías de estos algoritmos con el fin de apoyar a los diseñadores en la elección de la o las técnicas más adecuadas para sus necesidades. Todos estos autores coinciden en que el modelo de red utilizado y los objetivos priorizados al encarar una aplicación particular, son fundamentales para decidir las características de la técnica de *clustering* a incorporar.

2.3.1 Atributos de las técnicas de clustering

Los algoritmos para RISI basados en *clusters* pueden caracterizarse a partir de un conjunto de atributos que resultan de gran utilidad para compararlos y evaluar su aplicabilidad al momento de abordar un problema particular. Siguiendo a los autores de [1] podemos dividir estos atributos en tres categorías: propiedades de los *clusters*, capacidades de las cabeceras y características del proceso de *clustering*. Respecto a las propiedades de los *clusters* es importante tener en cuenta los siguientes atributos:

- *Número y estabilidad de los clusters*: Cuando los CH están predefinidos al iniciar el funcionamiento del sistema la cantidad de *clusters* queda determinada. Si por el contrario, las cabeceras se eligen entre los nodos de la red, el número de *clusters* es variable. Por otro lado, si los NC mantienen la membresía a un mismo *cluster* durante el tiempo de vida útil de la red, los *clusters* son fijos, mientras que si se producen variaciones en la constitución de los grupos de nodos, el *clustering* se denomina adaptativo o dinámico.
- *Topología intracluster*: La comunicación entre los nodos comunes y su coordinador puede ser directa o multisalto. Esta última es aconsejable cuando se necesita acotar el radio de transmisión o la cantidad de cabeceras está limitada, ya que permite que nodos alejados de sus coordinadores utilicen los servicios de vecinos intermedios para hacer llegar sus mediciones hasta los CH. Se debe tener en cuenta que si la cantidad de saltos es elevada, el retardo introducido en la entrega de datos puede resultar inaceptable. Por este motivo, muchos algoritmos limitan la profundidad del árbol intracluster. Otro inconveniente de la comunicación multisalto a tener en cuenta es que los NC cercanos al CH tienen una carga de trabajo mayor, por lo que pueden agotar sus recursos de energía en forma prematura.
- *Topología intercluster*: La comunicación entre las cabeceras y la estación base también puede ser directa o multisalto. En el primer caso, puede ser necesario utilizar una potencia de transmisión demasiado alta para los recursos disponibles, situación posible si no se conoce a priori la ubicación de las cabeceras en el área bajo estudio. Si se necesita supervisar una zona de dimensiones variables y se utiliza

clustering adaptativo, la segunda opción puede ser la única alternativa viable. Los inconvenientes son los mencionados en el ítem anterior, mayor retardo en la entrega de los datos y sobrecarga de las cabeceras cercanas a la estación base.

Los nodos que asumen el rol de coordinadores de su *cluster*, pueden tener distintas capacidades dependiendo del modelo de red y de la aplicación encarada. En primer lugar hay que considerar que los CH pueden ser fijos o móviles. Si bien la movilidad no es una propiedad habitual en la mayoría de las aplicaciones de RISI, algunas de ellas la incluyen. En este último caso, el *clustering* será necesariamente dinámico y exigente respecto del mantenimiento de la conectividad de la red. Por otro lado, las cabeceras pueden ser nodos comunes o tener recursos adicionales, en cuyo caso se habla de una red heterogénea. En la misma, los CH deberán elegirse siempre entre el subconjunto de nodos con mayores capacidades. La ubicación de este subconjunto especial de nodos no puede ser cualquiera, porque si quedan concentrados en alguna zona pueden producirse desconexiones en la red [41].

Finalmente, el rol de coordinador puede reducirse a recibir los datos de sus hijos y reenviar hacia la estación base, o realizar procesamiento de los mismos para acotar el número de transmisiones. Además pueden cumplir funciones adicionales referentes a la toma de decisiones basadas en el análisis de las mediciones recibidas [1].

El proceso del *clustering* por su parte, puede tener características diversas, entre las cuales se pueden mencionar:

- *Metodología:* La ejecución del algoritmo puede ser centralizada o distribuida. En el primer caso, se requiere información sobre la topología general de la red, y su eficacia es reducida en redes con un número elevado de nodos, donde reunir la información requerida por parte del control central resulta caro en tiempo y energía. Para redes de estas características las aproximaciones distribuidas son por lo tanto más adecuadas, ya que en ellas cada nodo decide proponerse como CH o unirse a un *cluster* basándose en información propia o a lo sumo de sus vecinos de un salto [85].
- *Elección de cabeceras:* La misma puede basarse en distintos parámetros. Hay redes donde los nodos están identificados y el algoritmo utiliza los ID para la selección de

CH. También es habitual que se utilicen otros atributos como la energía remanente, la distancia promedio de sus vecinos o el número de vecinos de un salto dentro de un rango de transmisión definido [85].

- *Convergencia en el tiempo:* Permite clasificar a los algoritmos como de convergencia variable o de convergencia constante en el tiempo. En el último caso, se suele introducir algún elemento aleatorio por lo que también se los conoce como algoritmos probabilísticos. Los algoritmos de convergencia variable, son adecuados para redes con un número bajo de nodos, ya que dependen del número total de nodos de la red para definir su convergencia temporal. En cambio, los algoritmos de convergencia constante, utilizan estrategias localizadas mediante las cuales un nodo puede definir su rol basándose exclusivamente en información propia o de vecinos de un salto [1] [85].

2.3.2 *Técnicas de clustering y objetivos de diseño de la red.*

Las técnicas de *clustering* adecuadamente utilizadas han demostrado ser efectivas para organizar la red en una jerarquía conectada y mejorar el desempeño de los algoritmos de encaminamiento para RISI [18]. Uno de los principales objetivos de los diseñadores al incluirlas en dichos algoritmos es incrementar la escalabilidad de la red, ya que estas técnicas son especialmente recomendadas para este fin. Sin embargo existen otros objetivos de diseño, algunos de ellos generales para los algoritmos de encaminamiento en redes de sensores, otros particulares para los algoritmos basados en *clusters*, que constituyen verdaderos desafíos para el diseñador y sobre los cuales se continúa investigando [1].

La información recolectada por una RISI llega hasta la estación base mediante las rutas definidas por el algoritmo de encaminamiento utilizado. El conjunto de estas rutas, constituye el árbol de encaminamiento que debe ser definido y mantenido por el algoritmo implementado. Esto involucra un costo en términos de recursos de los nodos, que si bien es inevitable no es directamente imputable al transporte de datos útiles hasta el *sink*. Por lo tanto, una cuestión básica en el diseño de un algoritmo es reducir el consumo de los recursos abocados a las tareas de encaminamiento frente a los utilizados para el envío de información [18].

Prolongar la vida útil de la red es fundamental para todo algoritmo de RISI, considerando los recursos limitados de los nodos y los escenarios en los que se despliegan las redes que suelen ser alejados, inhóspitos o de acceso muy dificultoso [4]. El *clustering* aporta para cumplir con este objetivo desde varios puntos de vista. Para comenzar, se trata de una topología especialmente adecuada para aplicar agregación de datos. Esta técnica, reduce drásticamente la cantidad de mensajes circulantes en la red, ahorrando potencia de transmisión y disminuyendo la posibilidad de colisiones [42] [57]. Además, una buena gestión de los *clusters* permite que los nodos comunes pasen a un estado de muy bajo consumo luego de finalizada la etapa de envío de mediciones a sus coordinadores. A esto se agrega el hecho de que las comunicaciones intracluster se realizan habitualmente con una potencia de transmisión reducida [8]. Sin embargo, se debe destacar que los CH son nodos más exigentes en consumo por su funcionalidad y mayor participación en el envío de la información a la estación base. Por este motivo, otro objetivo de los algoritmos basados en *clusters* es balancear la carga entre todos los miembros de la red para evitar el agotamiento prematuro de algunos dispositivos. Este objetivo es particularmente importante en redes homogéneas donde todos los nodos poseen la misma energía inicial, y obliga a efectuar cambios en los roles asignados con el fin de que no sean siempre los mismos dispositivos los que asumen las tareas más costosas [85].

Los algoritmos que aprovechan la posibilidad de que los transceptores funcionen en la modalidad de bajo consumo para prolongar el tiempo de vida del sistema, deben implementar alguna forma de sincronización de relojes en los nodos de la red, para que la definición de la estructura de encaminamiento y el posterior envío de información resulten efectivos. Por lo tanto, lograr esta sincronización ligada a la distribución temporal de tareas es un tópico de diseño importante para las redes que apliquen esta filosofía de funcionamiento [18].

Otros objetivos de las técnicas de *clustering* son lograr un buen nivel de conectividad y disminuir el retardo en la entrega de datos. Respecto al primero, los algoritmos deben prestar especial atención a la conexión entre CH, ya que la pérdida de mensajes agregados puede degradar la funcionalidad de la red. Se deben incorporar mecanismos que prevengan la desconexión de cabeceras, o permitan recuperarse a la red de estas situaciones. Si hay

requerimientos de retardo máximo, se deberá analizar la topología de los *clusters*, limitando la máxima cantidad de saltos hasta el coordinador [1].

La tolerancia a fallos es un requerimiento habitual en RISI, teniendo en cuenta que estas redes suelen trabajar con mantenimiento bajo o nulo. Para esta especificación también es fundamental la conectividad entre cabeceras, y la disponibilidad de mecanismos que permitan recuperarse de la caída o desconexión temporal de alguna de ellas. Una técnica muy utilizada es la de rotación de cabeceras, que además de prevenir fallos, balancea la carga entre todos los nodos de la red [1].

Los *clusters* son especialmente adecuados para implementar agregación aprovechando la correlación existente entre las variables estudiadas. En este sentido, los autores de [72] demuestran que si existe un alto grado de agregación en los *clusters*, un algoritmo que implemente esta técnica tendrá siempre un mejor desempeño en términos de consumo que uno que no lo haga. Por lo tanto, otro objetivo de interés es adecuar la definición de los grupos de nodos para favorecer y simplificar el procesamiento distribuido de información en la red [12] [51] [72].

Es importante destacar que los objetivos de diseño mencionados están relacionados entre sí, y la mejora en el logro de uno de ellos puede empeorar el desempeño del algoritmo en otros. El diseñador deberá establecer sus prioridades de acuerdo con los requerimientos de la aplicación que aborde, intentando alcanzar la mayor generalidad posible para que las técnicas aplicadas resulten de utilidad para un conjunto de aplicaciones de similares características [18].

El algoritmo de encaminamiento desarrollado deberá ser evaluado en cuanto a su adecuación y cumplimiento de los objetivos de diseño establecidos. La evaluación se basa en un conjunto de parámetros cualitativos y cuantitativos que permiten analizar el comportamiento del algoritmo y compararlo con otros. Los parámetros cualitativos se refieren a propiedades deseables de los algoritmos que son difíciles de evaluar como la escalabilidad y el grado de distribución en la operación del algoritmo. En cambio los parámetros cuantitativos son métricas que pueden ser evaluadas con modelos analíticos, mediciones o simulación. Entre ellos se puede mencionar consumo de energía, latencia,

tiempo de configuración de la red y cantidad de nodos dormidos. La definición de cada métrica dependerá de los requerimientos de las aplicaciones abordadas y de los modelos adoptados para la red, y permitirá comparar el desempeño de distintos algoritmos en los dominios seleccionados [37].

2.4 *Agregación de datos*

El envío y recepción de mensajes son actividades que consumen una cantidad significativa de energía. Por lo tanto es importante reducir, en la medida de lo posible, el número de mensajes intercambiados por los nodos de la red, y para ello se han desarrollado técnicas de agregación. Según Fasolo et al. [24] la agregación de datos se puede definir como el proceso de colectar y encaminar la información a través de una red multisalto, procesando los datos en nodos intermedios con el objetivo de reducir el consumo de recursos (en particular energía), para incrementar el tiempo de vida del sistema. En una red de este tipo los nodos elegidos como puntos de agregación obtienen un único mensaje, lo más compacto posible, que resulta representativo de los datos recibidos desde sus vecinos. Este agregado se transmite hacia la estación base, evitándose el envío de todos los mensajes que él representa [42] [57].

Los resultados alcanzados por los investigadores en el área permiten afirmar que cuando se propaga la información recolectada hacia el *sink*, las técnicas de agregación resultan muy efectivas para disminuir el consumo de la red [57] [66] [84]. Cabe destacar que este importante beneficio tiene su costo en dos sentidos. En primer lugar la incorporación de agregación de datos introduce demora en la entrega de las mediciones en la estación base, debido a que los nodos encargados de procesar las mediciones deben esperar la llegada de la información desde sus hijos. Además, los puntos de agregación deben soportar una mayor carga de procesamiento y disponer de los recursos necesarios para almacenar los datos recibidos. En este sentido se debe tener en cuenta que los beneficios aportados por el uso de agregación se incrementan cuando las variables del medio estudiadas están altamente correlacionadas en tiempo, espacio, o mediante algún otro criterio semántico que permite utilizar agregados sencillos que implican poca sobrecarga [24].

La implementación de técnicas de agregación de datos en una red de sensores requiere de tres elementos fundamentales, algoritmos de encaminamiento que se puedan adaptar a las necesidades de procesamiento, funciones de agregación que resulten efectivas y, ligadas con ellas, representaciones eficientes de la información recogida [24].

2.4.1 Encaminamiento y agregación

Los algoritmos de encaminamiento que incorporan agregación de datos deben elegir las rutas de envío de mensajes de manera de favorecer el procesamiento de la información. Desde este punto de vista, no siempre el camino más corto es el más eficiente, y tampoco transmitir las mediciones en forma inmediata la mejor política a aplicar [24].

La cantidad y ubicación de los puntos de agregación es un aspecto relevante en el diseño de los algoritmos [42], y los desarrollos realizados pueden clasificarse de acuerdo con el punto de vista de Fasolo et al. [24] en:

- *Basados en árboles:* La idea es construir una estructura con origen en la estación base. Los puntos de agregación se eligen con distintos criterios como la posición en el árbol, los recursos remanentes en los nodos, el tipo de información a procesar o el costo de la operatoria. En este sentido los dispositivos responsables de realizar agregación proveen un soporte efectivo para el procesamiento de los datos. Protocolos como Difusión dirigida [39], *Tiny AGgregation* [49] o PEGASIS [45] pertenecen a esta categoría. La principal desventaja de este enfoque es su debilidad frente a fallas o caídas de los nodos por agotamiento de sus baterías. Si algún dispositivo intermedio resulta afectado no sólo se pierden sus datos sino también las correspondientes a los hijos que lo utilizan como enlace para enviar sus mediciones. Además si se presentan interferencias o colisiones, puede perderse algún agregado, lo que implica que todos los mensajes que representa no llegarán a la estación base.
- *Basados en clusters:* La estructura de encaminamiento es especialmente adecuada para aplicar agregación, ya que en la misma los CH actúan como puntos de procesamiento de las mediciones de su *cluster*. Presenta las desventajas de la categoría anterior, que pueden atenuarse redefiniendo periódicamente los *clusters* de la red, aunque se debe tener en cuenta que dicho procedimiento siempre tiene un

costo asociado respecto del consumo de recursos. Entre los algoritmos con estas características se pueden mencionar LEACH [33], COUGAR [83], y también CLUDITEM cuyo desarrollo se describe en esta tesis.

- *Algoritmos multipaso:* Este enfoque busca lograr algoritmos más robustos que los pertenecientes a los dos grupos anteriores. La idea básica es que los nodos envíen sus mediciones a más de un vecino, aprovechando las características del medio inalámbrico. Si bien es un enfoque mucho más tolerante a fallas de los nodos, se debe tener en cuenta que se propagan duplicados que formarán parte de varios mensajes agregados, y que la circulación de mensajes será mayor. *Synopsis diffusion* [53] es un algoritmo de esta categoría.
- *Algoritmos híbridos:* Algunos investigadores han intentado aprovechar los beneficios de los esquemas basados en árboles y los algoritmos multipaso, y han desarrollado protocolos como *Tributaries and Deltas* [50]. En este algoritmo se utiliza uno u otro enfoque en distintas regiones de la red dependiendo del tráfico existente. Si hay poca circulación de mensajes se elige un esquema basado en un árbol y, en caso de existir un tráfico importante se recurre a un enfoque multipaso para disminuir la pérdida de información.

2.4.2 Funciones de agregación

Existe un gran número de funciones de agregación que pueden utilizarse con el fin de distribuir el procesamiento de los datos en la red. Estos agregados difieren respecto a su complejidad de obtención y memoria requerida. Sin embargo, las propiedades de cada función de agregación determinan si pueden o no ser procesadas en forma eficiente por la red, como se puntualiza en [49]. Siguiendo a estos últimos autores, se puede clasificar los agregados más usuales, basándose en propiedades que se relacionan con las características más destacables de las RISI:

- *Sensibilidad a duplicados:* Los agregados sensibles a valores duplicados no computan el mismo resultado cuando los datos se repiten. Teniendo en cuenta que la información en las RISI es habitualmente redundante, el hecho de que la función utilizada posea esta propiedad, afectará la fiabilidad de los resultados. Ejemplos de

funciones sensibles a duplicados son el PROMEDIO, CUENTA y MEDIANA. En cambio agregados como MAX, MIN o CUENTA_DISTINTOS son insensibles a datos repetidos.

- *Dato versus cálculo:* Algunas funciones devuelven un único valor representativo, como MAX o MIN. Otras, como PROMEDIO o CUENTA realizan algún cálculo utilizando los datos recibidos. Estas últimas pueden estimarse con márgenes de error reducidos aún en ambientes ruidosos con importante pérdida de mensajes. En cambio las funciones que devuelven un único valor son muy imprecisas, a pesar de que la pérdida de mensajes sea baja.
- *Monotonicidad:* El valor del agregado parcial obtenido por estas funciones resulta siempre mayor o igual, o en su defecto, menor o igual al que se logra cuando se procesan un mayor número de mediciones. Es el caso de funciones como MAX, MIN, CUENTA o CUENTA_DISTINTOS, que garantizan el cumplimiento de los umbrales parciales aunque se hayan procesado pocos valores. Por lo tanto, si estos valores umbral son suficientes para tomar decisiones, el uso de estos agregados permite obtener resultados en forma temprana y reducir el tráfico de mensajes. Por otro lado, si los nodos han “escuchado” los valores enviados por sus vecinos, pueden decidir no transmitir, en el caso de que sus mediciones no modifiquen el resultado logrado.
- *Requerimiento de estado parcial:* Las funciones de agregación necesitan distinta cantidad de información parcial. Hay funciones muy exigentes en este aspecto como MEDIANA o HISTOGRAMA, que requieren del conjunto total de mediciones realizadas, a menos que se implemente algún tipo de aproximación como se propone en [64]. Otras funciones como SUMA o CUENTA requieren un almacenamiento parcial del mismo tamaño que el resultado final. Esta propiedad no sólo afecta el monto de información que debe almacenar un nodo, sino fundamentalmente, el tamaño de los mensajes que deberán intercambiarse.

Algunos autores como Fasolo et al. [24], destacan otra propiedad importante de las funciones de agregación referida a si reducen o no el tamaño de los mensajes a transmitir. Los agregados con reducción de tamaño, como MAX, MIN, SUMA o PROMEDIO, no

permiten la reconstrucción de las mediciones originales cuando son recibidos en la estación base. En cambio, cuando no se disminuye la longitud resultante del mensaje, y simplemente se concatenan valores, los datos medidos pueden ser recuperados por el *sink* a partir del agregado que recibe.

La mayoría de los trabajos reportados en el área seleccionan la función de agregación en base a las características del problema abordado, utilizando conocimiento sobre los fenómenos estudiados y las variables involucradas. Las propiedades de los agregados descriptas en este apartado son, desde este punto de vista, muy importantes no sólo para elegir el agregado más conveniente sino también para evaluar la calidad de los resultados obtenidos. Sin embargo, se puede mencionar que existen algunos enfoques diferentes como el de los investigadores que dan cuenta de sus resultados en [32]. Ellos dejaron de lado la selección de un agregado específico y se centraron en aplicaciones donde los requerimientos temporales resultan prioritarios. Por lo tanto, su trabajo se orientó a optimizar la utilización del canal cumpliendo con las restricciones temporales de entrega de los datos, definiendo la agregación a realizar en base a patrones locales de tráfico en la red.

Los resultados obtenidos por la gran mayoría de los autores permite afirmar que el monto del ahorro que se obtiene al incorporar agregación, depende en gran medida de la aplicación que define la función de agregación a utilizar [24]. Algunas aplicaciones admiten agregados sencillos como la obtención del valor máximo, mínimo o promedio. El estudio de otros fenómenos en cambio, requiere de la transmisión de todos los valores obtenidos o gran parte de ellos. Estos últimos casos son los más desafiantes al momento de diseñar alguna técnica de agregación, que manteniendo los errores acotados, permita cumplir con los requerimientos del sistema [64].

2.4.3 Técnicas de agregación periódicas

La técnica de agregación a utilizar está íntimamente relacionada con la modalidad de recolección de información de la red de sensores. Es importante diferenciar las aplicaciones en las cuales todos los dispositivos deben reportar la información recolectada a la estación base, de aquéllas en las que un subconjunto de nodos detecta la presencia de algún evento

en su zona de influencia [9]. En este último grupo, en el cual las restricciones temporales son muy importantes, hay dos aspectos fundamentales [87]:

- *Evitar detecciones falsas:* Para eventos con una distribución normal en un rango limitado de valores la definición de umbrales es en general suficiente para eliminar lecturas erróneas. Sin embargo, este procedimiento resulta insuficiente para capturar eventos complejos presentes en escenarios como los descritos en [46], donde los autores proponen trabajar comparando la información recolectada con un conjunto de patrones espacio-temporales del ambiente.
- *Llevar la información hasta la estación base cumpliendo restricciones temporales:* La gran mayoría de los desarrollos utiliza encaminamiento basado en árboles [31]. Existen sin embargo autores que afirman que mantener una estructura de encaminamiento cuando la región de detección es cambiante resulta demasiado costoso. En este sentido, en [23] se reporta un trabajo donde se realiza agregación sin estructura subyacente, en el cual los nodos toman las decisiones de encaminamiento “en vuelo”.

Esta tesis se enfoca hacia las aplicaciones de supervisión ambiental, en las cuales es habitual realizar una adquisición periódica de datos, con el fin de mantener un estado de situación actualizado del fenómeno sobre el que se desea obtener información. Por lo tanto se relaciona con la modalidad de recolección de datos que involucra a todos los nodos de la red. En estas aplicaciones, la pronta llegada al *sink* de la información no es en general el aspecto más importante, sino que se pone acento en la calidad y cantidad de datos reportados. Se habla de “rondas de recolección o medición”, donde todos los dispositivos obtienen valores de las variables de interés y los transmiten hacia la estación base. Estas “rondas” se repiten en períodos cuya duración queda determinada por la naturaleza del fenómeno bajo estudio. Las mediciones realizadas en cada una no deben superponerse ni procesarse con datos correspondientes a otras rondas, por lo que la temporización ligada al proceso de agregación de la información es muy importante. En este sentido, autores como los que reportan sus resultados en [22] y [66], enfatizan el impacto del esquema temporal sobre el desempeño de la red y proponen clasificar las técnicas de agregación periódicas de la siguiente forma:

- *Periódica simple:* En este tipo de algoritmos, los nodos encargados de realizar agregación esperan un período fijo, procesan los datos recibidos y transmiten un único paquete de datos. Si algún nodo se retrasa en el envío de datos o falla, su información no formará parte del agregado enviado hacia la estación base. Un ejemplo muy conocido es el algoritmo de Difusión Dirigida [39]. CLUDITEM cuyo desarrollo se reporta en esta tesis, también pertenece a esta categoría.
- *Periódica por salto:* En estos algoritmos, los nodos que son puntos de agregación deben saber quiénes de entre sus vecinos le enviarán información en cada ronda de medición. Entonces, los nodos agregan los datos recibidos y los envían cuando han colectado la información de todos sus hijos. En caso de que algún hijo no se reporte, se espera un período adicional predeterminado, y si no se obtiene respuesta, se difunde el agregado logrado hasta ese momento. Es importante mencionar que para que los puntos de agregación conozcan la identidad de los nodos que dependen de ellos, se deberán intercambiar un mayor número de mensajes de control. En esta categoría pueden mencionarse algoritmos como LEACH [33] y COUGAR [83].
- *Periódica por salto ajustado:* Estos algoritmos utilizan el mismo principio que los de agregación periódica por salto, pero incorporan un esquema de retardos para la difusión de los mensajes, basado en la posición de cada nodo en el árbol de distribución. El algoritmo de *cascading timeouts* propuesto por los autores de [66] pertenece a este grupo.

El esquema temporal de agregación deberá compatibilizarse con las especificaciones establecidas y ajustarse de manera de integrar el procesamiento de los datos con la estructura de encaminamiento, teniendo siempre presente que tipo de información disponen en forma local los nodos de la red. La técnica que se defina deberá ser ponderada respecto del ahorro de energía que constituye un requerimiento fundamental para su diseño, pero los algoritmos que la utilicen serán evaluados también en otros aspectos que no resultan independientes entre sí [24] [66]:

- *Frescura:* Evalúa la diferencia temporal entre el momento de obtención de los datos y su llegada a la estación base.

- *Precisión:* Mide la exactitud con la que el agregado recibido en el *sink* representa las mediciones realizadas por la red. Resulta afectada por errores en las mediciones y también, por pérdida de mensajes debida a fallas en los dispositivos o interferencias en las comunicaciones. Los autores de [24] puntualizan que además se debe tener en cuenta la pérdida de información debida a la reducción de tamaño de los mensajes agregados.
- *Sobrecarga:* Cuantifica el grado de complejidad y las necesidades de almacenamiento local que se agrega al implementar el procesamiento de las mediciones.

El problema a abordar será el que determine la magnitud de error aceptable, el período de espera admisible para la llegada de los datos, el porcentaje de mensajes que pueden perderse, la sobrecarga de procesamiento tolerable y el tiempo de vida necesario. El diseñador deberá llegar a una solución de compromiso entre todos estos requerimientos, que brinde un balance adecuado para alcanzar el desempeño esperado para la aplicación. Además, como ya se ha puntualizado, la técnica elegida debe lograr una integración exitosa con la estructura de encaminamiento en base a la información local disponible, e incorporar representaciones de los datos acordes con las necesidades de almacenamiento y operatoria.

Capítulo 3. Requerimientos para redes de sensores

Las RISI encuentran cada vez mayor campo de aplicación, y existe una amplia bibliografía sobre sus características y las técnicas que se desarrollan para mejorar su desempeño [9] [84]. La mayoría de los autores aborda la temática desde la perspectiva de las redes de comunicaciones y las distintas capas de los protocolos utilizados, situación que resulta absolutamente natural considerando que las RISI han recibido importantísimos aportes desde dicha comunidad. Sin embargo, teniendo en cuenta que las redes de sensores cuentan ya con una década de intensa actividad, y constituyen una tecnología sumamente promisoría, se han incorporado profesionales e investigadores de otras áreas que han sumado su esfuerzo para mejorar el proceso de desarrollo de estos sistemas. En este sentido, algunos trabajos como [7], [48] y [63], destacan la importancia de abordar el estudio del espacio de diseño de las aplicaciones, haciendo énfasis en el análisis de los requerimientos y la definición de un lenguaje estandarizado para describirlos. Dichos autores opinan que profundizar el estudio de estos aspectos permite lograr soluciones mucho más ajustadas a las necesidades de las aplicaciones, y además facilita el reconocimiento de propiedades comunes de redes de sensores en distintos dominios, que pueden ser aprovechadas para lograr desarrollos más flexibles ampliando su campo de aplicación. En esta tesis se acuerda con el punto de vista antes mencionado, y se siguen los lineamientos establecidos por los autores de [48] y [63] para proponer las categorías de especificaciones para RISI que se detallan a continuación.

1. Requerimientos Temporales

- 1.1. *Latencia*: Tiempo transcurrido entre que se realiza una medición (o se detecta un evento) y llega la información a la estación base. Este requerimiento se caracteriza como crítico cuando se debe alertar sobre una situación de peligro, pero puede resultar irrelevante cuando sólo se necesita recolectar información para construir un modelo o realizar un estudio.

1.2. *Tiempo de vida*: Período de tiempo durante el cual la red cumple con sus objetivos de diseño. Su definición está estrechamente ligada con la información que los usuarios esperan recibir de la red.

2. Requerimientos Espaciales

2.1. *Distribución de los nodos*: Manual o aleatoria. En el primer caso, se conoce la ubicación inicial de cada dispositivo, en el segundo los nodos se arrojan desde algún medio de transporte en el área bajo estudio.

2.2. *Área a supervisar*: Dimensiones y características del ambiente en el cuál se desplegará la red.

2.3. *Resolución espacial*: Cantidad de puntos de medición y distancia entre ellos. Ubicación respecto del área estudiada de la/las estaciones base.

2.4. *Movilidad*: Nodos fijos o móviles. El caso de movilidad se puede presentar porque los dispositivos cambian de posición o porque el fenómeno analizado se desplaza.

2.5. *Cobertura del área bajo estudio*: Fracción de área supervisada al menos por un sensor. Se caracteriza como una cobertura parcial, total, o incluso redundante cuando más de un dispositivo de medición cubre cada zona bajo estudio.

3. Requerimientos de Medición

3.1. *VARIABLES A MEDIR*: Detalle de las variables respecto a cantidad, resolución exigida y características de la correlación de valores en espacio y tiempo para cada una.

3.2. *Puntos de medición*: Especificación de cuales nodos de la red miden cada variable de interés.

3.3. *Modalidad y frecuencia*: La red puede funcionar reaccionando a eventos del medio, a demanda o realizando mediciones periódicas de los parámetros. En el último caso se debe definir el período a partir del cual cada nodo adquiere información del ambiente. Estos requerimientos definirán las características del tráfico en la red.

3.4. *Conocimiento sobre el entorno de cada nodo*: Se refieren al momento en que se realiza cada medición y a la ubicación del dispositivo en el área estudiada.

4. **Requerimientos de Integración:** Pueden existir otros sistemas con los cuales la red debe interactuar. En este caso se deben establecer las especificaciones respecto a la conexión y el tipo de intercambio de información necesario. Estas especificaciones son importantes para el diseño de la red pero absolutamente particulares a cada situación o usuario, y por lo tanto, no se las considerará al efecto de agrupar las aplicaciones.

Si bien los requerimientos son particulares a cada situación, es importante lograr la máxima generalidad posible, tratando de agrupar aplicaciones de forma de ampliar la utilidad de los algoritmos que se desarrollen. En este sentido, en la próxima sección se trabaja sobre las aplicaciones orientadas a realizar supervisión ambiental.

3.1 *Las aplicaciones de supervisión ambiental*

La investigación sobre ecosistemas y medioambiente ha cobrado cada vez mayor importancia en los últimos tiempos debido al impacto provocado por las actividades humanas. Los investigadores necesitan supervisar los parámetros presentes en distintos ecosistemas no solo para comprender la evolución de los fenómenos que allí ocurren, sino también para estudiar la respuesta de las especies animales y vegetales a los cambios climáticos y ambientales, e incluso alertar sobre situaciones de riesgo [74]. Por otro lado, la necesidad creciente de producir mayor cantidad de alimentos, ha llevado a la industria agroalimentaria a incorporar tecnología con el objetivo de mejorar la producción. La agricultura moderna debe cumplir con altos estándares de calidad, pero además debe controlar el impacto ambiental minimizando los posibles daños sobre el entorno. En este sentido es muy importante la medición de variables ligadas a las áreas productivas, con el fin de mejorar los rendimientos y prevenir efectos ecológicamente indeseables en áreas aledañas [58].

Las redes inalámbricas de sensores inteligentes son una herramienta muy promisoría para tareas de supervisión de ambientes, tanto exteriores como interiores. Permiten abarcar áreas extensas, simplifican la instalación de los dispositivos al eliminar el cableado, y pueden alimentarse con baterías funcionando en forma autónoma en locaciones alejadas o de difícil acceso [9] [26]. Además es importante destacar que, en muchos casos, la presencia humana puede distorsionar e incluso dañar el ambiente bajo estudio, como ocurre cuando se trabaja

con poblaciones de animales silvestres. Estas redes obtienen datos en forma dinámica, durante períodos prolongados, lo que permite a los científicos acceder a información antes impensable para realizar sus estudios, hacer seguimiento de situaciones o construir modelos, contribuyendo a mejorar distintas actividades ligadas al desarrollo de la vida humana y a la preservación del ambiente natural [30] [47].

Las aplicaciones de supervisión ambiental pueden, en general, dividirse en dos grandes grupos considerando sus objetivos principales:

1. *Aplicaciones que responden a eventos específicos.* Ejemplos típicos son las redes de alerta sobre situaciones de peligro como incendios o inundaciones, en las cuales la velocidad de respuesta es fundamental. Los sistemas deben ser capaces de reconocer las situaciones problemáticas a partir de las mediciones realizadas y transmitir la alarma hasta la estación central lo antes posible. Por lo tanto el requerimiento de latencia es crítico.
2. *Aplicaciones que colectan datos periódicamente en un área específica de trabajo.* Estos sistemas realizan mediciones periódicas de las variables de interés, que pueden utilizarse con fines tan diversos como construir modelos de los ambientes, estudiar los fenómenos que se desarrollan en los mismos o controlar dispositivos que pueden actuar para alterar algunos de los parámetros en cuestión. En estas aplicaciones el acento está puesto en la calidad de la información y no en la velocidad de respuesta, por lo que la latencia no resulta prioritaria.

En esta tesis, el interés está centrado en el segundo grupo de aplicaciones, que habitualmente posee un conjunto de requerimientos con características similares:

- *Distribución de los nodos:* los dispositivos, que son fijos, se ubican manualmente en determinados puntos de interés.
- *Frecuencia de las mediciones:* periódicamente todos los nodos abocados a supervisar una variable del medio envían información a la estación base, ya que se necesitan datos sobre toda el área bajo estudio. Esto significa que cuando corresponda enviar información hacia la estación base, se producirá un tráfico importante en la red, mientras

que en los períodos en los que no se realizan mediciones no existirá intercambio de mensajes entre los nodos.

- *Variables a medir:* Los parámetros a supervisar tienen variación lenta en el tiempo y en el espacio y es habitual que los valores estén fuertemente correlacionados.
- *Área a supervisar:* Posee dimensiones variables, por lo que el diseño de la red de sensores debe prever su escalabilidad.
- *Autonomía de funcionamiento:* La red puede ser desplegada en zonas alejadas y se necesita que cumpla con sus tareas sin mantenimiento durante varios meses.
- *Calidad de la información colectada:* Las aplicaciones requieren cierta calidad de servicio (QoS) relacionada con los datos que llegan a la estación base, y por lo tanto, el requerimiento sobre el tiempo de vida de la red se define en base a dicha característica.

La lista no cubre la totalidad de las especificaciones puntualizadas en el apartado anterior. Esto se debe a que algunos de los requerimientos detallados en la taxonomía son particulares para cada aplicación, como la cantidad exacta de sensores, distancia entre ellos, cuales son las variables a medir o la resolución necesaria en los valores a obtener. En la lista previa sólo se recogen especificaciones que en general están presentes en aplicaciones ambientales de medición periódica, con el objetivo de establecer lineamientos de diseño para los algoritmos de encaminamiento en el dominio.

3.2 Lineamientos de diseño para un algoritmo de encaminamiento

Las características de un algoritmo de encaminamiento adecuado para las aplicaciones elegidas se definieron en base a los requerimientos enumerados en el apartado anterior. Para ello se trabajó sobre la base de nodos sensores homogéneos en cuanto a sus recursos, y se definió la QoS de las aplicaciones como un porcentaje máximo admisible de pérdida de información. Este último requerimiento es típico en aplicaciones que necesitan información sobre toda la región donde se ha desplegado la red, para obtener un mapa de variación de las variables con el fin de construir un modelo de comportamiento del sistema, tomar decisiones o llevar adelante acciones correctivas.

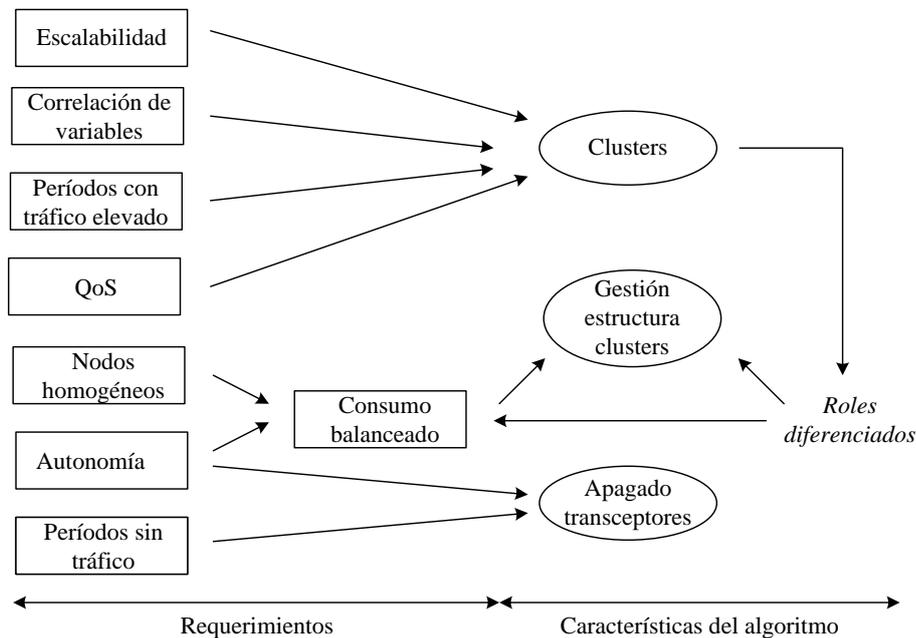


Figura 3. Relación entre los requerimientos de las aplicaciones elegidas y las características del algoritmo.

En la figura 3 los requerimientos se destacan con un rectángulo, mientras que las características a implementar en el algoritmo se representan con óvalos. En este sentido, la figura 3 muestra que los requerimientos de escalabilidad, QoS definida como un porcentaje máximo de pérdida de datos, alta correlación de variables y necesidad de acotar el tráfico cuando los nodos transmiten hacia la estación base, apuntan a definir un algoritmo jerárquico basado en *clusters* constituidos por nodos próximos. Esto se fundamenta en que los *clusters* proporcionan un conjunto de ventajas acordes con los requerimientos planteados, a saber [51]:

- Permiten aprovechar la correlación de los datos, simplificando su procesamiento al aplicar agregación.
- Se orientan al procesamiento local de la información, reduciendo la circulación de mensajes de datos y acotando la congestión de la red.
- Facilitan el control por zona del porcentaje de pérdida de datos, ya que están constituidos por nodos próximos físicamente.

- Proporcionan escalabilidad a la red ya que ampliar el área de trabajo se reduce a agregar *clusters*, cuya gestión es simple ya que están constituidos por grupos acotados de nodos próximos físicamente.

La elección de una red jerárquica basada en *clusters*, supone la definición de roles diferenciados entre los nodos que la constituyen. Los nodos que asumen la tarea de ser cabeceras de *cluster* (CH), están más exigidos en cuanto al consumo de energía que los nodos comunes (NC) [51]. Esta propiedad de las redes basadas en *clusters*, sumada a las especificaciones de autonomía y homogeneidad de recursos en los nodos, ocasionan el agregado de un nuevo requerimiento, en principio no considerado, que es la necesidad de balancear el consumo de los dispositivos. De no ser así, los nodos más exigidos agotarían su energía en forma prematura, produciendo desconexiones en la red. Por su parte, este último requerimiento, apunta a la necesidad de realizar una gestión de la estructura de *clusters*, teniendo en cuenta los dos tipos de nodos presentes en la red (CH y NC). Finalmente, la figura 3 muestra que los requerimientos de autonomía de funcionamiento y presencia de períodos de inactividad en la red, apuntan a la necesidad de apagar los transceptores de los nodos todo el tiempo posible, con el fin de ahorrar energía.

Capítulo 4. CLUDITEM

4.1 *Características generales*

CLUDITEM es un algoritmo de encaminamiento para redes de sensores, orientado hacia aplicaciones de supervisión ambiental con adquisición periódica de datos. Para su desarrollo se siguieron los lineamientos de diseño establecidos en el capítulo 3, y se asumió la presencia de una única estación base fuera del área bajo estudio. Este algoritmo divide su funcionamiento en tres etapas bien diferenciadas que se repiten en forma periódica durante todo el tiempo de vida de la red. La primera se ocupa del establecimiento del árbol de encaminamiento, la segunda del envío de datos al *sink*, y durante la tercera todos los nodos de la red permanecen en estado de bajo consumo. Esta última etapa se relaciona con una característica importante de CLUDITEM que prevé el apagado de los dispositivos todo el tiempo posible, manteniendo la QoS establecida, en respuesta al requerimiento de autonomía de funcionamiento. Es importante destacar que al ser CLUDITEM un algoritmo jerárquico distribuido, el apagado de los transceptores lo define cada nodo utilizando su propio reloj. Por lo tanto, se requiere sincronizar los relojes de los miembros de la red, con el fin de que el intercambio de mensajes sea efectivo. Los aspectos específicos del logro de esta sincronización quedan excluidos del alcance de esta tesis, y por ello en todo el desarrollo se trabaja suponiendo un desfasaje máximo de relojes intra e intercluster, que se incluye en las expresiones asociadas con parámetros del algoritmo.

La gestión de la estructura de *clusters* que se define, es otra característica a implementar de acuerdo con los lineamientos del capítulo 3, y se aborda dividiendo el área de trabajo en una grilla virtual, en la cual cada división o celda corresponde a una zona del área supervisada. Las dimensiones de la cuadrícula se definen según los requerimientos de la aplicación, pero en cada celda se distribuye manualmente la misma cantidad de dispositivos sensores. La grilla se utiliza para definir los postulantes a CH en cada reconfiguración de la red y para plasmar estrategias tendientes a lograr una distribución de cabeceras en toda el área a supervisar balanceando el tamaño y profundidad de los *clusters* que se definen. Además, se agrega una técnica de rotación periódica de CH, que apunta a balancear el consumo entre

los dispositivos teniendo en cuenta los diferentes roles que cumplen en la estructura de encaminamiento.

Los nodos de la red transmiten los mensajes que intercambian con dos potencias diferentes. La potencia de transmisión intracluster se utiliza en la definición de los grupos de nodos y en el envío de las mediciones hacia las cabeceras. La segunda potencia, que corresponde a un radio de transmisión mayor, se utiliza en la configuración del árbol de CH y en el envío de los agregados hacia la estación base. La definición del valor de cada uno de estos dos radios de transmisión, depende de las dimensiones de las celdas de la grilla y de la distancia entre los dispositivos de medición.

La comunicación entre los nodos de una red basada en CLUDITEM es multisalto, y para las capas inferiores del protocolo de comunicaciones se adoptó el estándar IEEE 802.15.4, recomendado para este tipo de redes [38]. En este sentido, el algoritmo combina CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) sin slots en la capa de acceso al medio, con un esquema TDMA (*Time Division Multiple Access*) definido a nivel de encaminamiento con el fin de reducir las colisiones intra e intercluster.

Finalmente se debe destacar que las aplicaciones de supervisión ambiental, hacia las cuales CLUDITEM está orientado, comparten un conjunto importante de requerimientos, pero cada una posee sus particularidades. Por lo tanto, otra característica del algoritmo consiste en la inclusión de un conjunto de parámetros orientados a flexibilizar su funcionamiento para adaptarlo a variaciones acotadas de algunas especificaciones.

4.2 *Trabajos relacionados*

Los algoritmos de encaminamiento jerárquicos basados en *clusters*, proporcionan importantes ventajas para las RISI, respecto a su escalabilidad y eficiencia en la comunicación entre los nodos [4]. Por estos motivos, se han desarrollado numerosos algoritmos con estas características, como los mencionados en [1], [2] y [24]. En particular, CLUDITEM utiliza la idea de rotación de cabeceras de *cluster* propuesta por Heinzelman en su algoritmo LEACH [33]. La rotación de CH permite distribuir el consumo de energía entre los nodos, ya que el rol de cabecera es exigente respecto al uso de recursos. Sin

embargo, un inconveniente importante de LEACH es que al seleccionar las cabeceras en forma aleatoria, las mismas pueden quedar concentradas en algún sector de la red. Esto podría ocasionar la desconexión de un número indeterminado de nodos, como se puntualiza en [4]. Para prevenir esta situación, CLUDITEM divide el área bajo estudio en una cuadrícula virtual cuyo uso ya fue propuesto por Al-Karaki en [5]. En su algoritmo, los *clusters* coinciden con las divisiones de la cuadrícula, y el rol de cabecera se rota entre los nodos miembros de la misma teniendo en cuenta su energía remanente. En CLUDITEM, cuando se rotan las CH, se vuelven a definir los *clusters*, pudiendo incorporarse nodos de otras divisiones de la grilla. Esto involucra un mayor número de mensajes de control, pero dichos mensajes son muy breves. La importante ventaja que se obtiene, es lograr *clusters* de menor profundidad, reduciendo el intercambio de mensajes de datos, típicamente de mayor longitud, en el envío de las mediciones hacia los CH. CLUDITEM combina una grilla virtual con la definición dinámica de *clusters* con el objetivo de lograr una distribución uniforme de CH en el área bajo estudio, y un consumo de energía menor y mejor distribuido entre los nodos de la red. De esta manera implementa las características establecidas a partir de los requerimientos de las aplicaciones seleccionadas.

Las capas inferiores del protocolo se basan en el estándar de la IEEE 802.15.4, ampliamente utilizado en redes de sensores [9] [84]. En el trabajo de Deliang [19] se clasifican los protocolos de acceso al medio (MAC) en cuatro categorías. La primera corresponde a los protocolos basados en la contención que compiten por el uso del canal, orientados a la solución de conflictos, pero costosos desde el punto de vista del consumo [89]. La segunda agrupa a los protocolos que utilizan un esquema de tipo TDMA, asignando *slots* de tiempo a cada nodo que debe enviar datos, naturalmente libres de colisiones. En tercer lugar se encuentran los protocolos híbridos, como el IEEE 802.15.4, en los cuales se combinan las ventajas de las dos categorías anteriores. Por último se consideran los protocolos de capas relacionadas que son un área de investigación muy promisoria [20]. CLUDITEM incorpora para la capa de acceso al medio el estándar IEEE 802.15.4 en modo *beaconless*, recomendado para redes multisalto como se reporta en [67]. La estructura de encaminamiento de la red es responsabilidad de CLUDITEM, y en este sentido se consideró la importancia que tienen las interferencias intra e intercluster, en lo que a pérdida

de información se refiere, para los algoritmos jerárquicos basados en *clusters* [29]. Por estos motivos, CLUDITEM incluye un esquema TDMA para prevenir la pérdida de mensajes por colisiones en la fase de definición de *clusters* y también en las fases correspondientes al envío de mediciones hacia la estación base.

Existen pocos trabajos sobre algoritmos basados en *clusters* que combinan información de encaminamiento con algún mecanismo de acceso al medio para mejorar el desempeño del sistema. El algoritmo presentado en [89] incluye *clusters* con topología estrella, y la asignación de los *slots* está a cargo del *sink* y las cabeceras de *clusters*, que deben conocer la topología completa de la red. Por su parte los autores de [29] describen un algoritmo centralizado que requiere muchos mensajes de control para organizar las transmisiones, incrementando el tráfico en la red.

CLUDITEM combina CSMA/CA sin *slots* en la capa MAC con un esquema TDMA definido a nivel del encaminamiento. Es un algoritmo distribuido sencillo que sólo utiliza información local a cada nodo para organizar la transmisión de datos, minimizando el número de mensajes de control, de forma de disminuir el tráfico de la red y el consumo de energía. Sin embargo, la definición de *slots* de transmisión de mediciones incrementa el tiempo en que los dispositivos deben permanecer en actividad, y por lo tanto también aumenta el consumo de energía. En este sentido, se trabajó para lograr un esquema TDMA que apunte a un balance entre la pérdida de información admisible y el tiempo que los nodos deben mantener sus transceptores activos, como se reporta en el capítulo 5.

4.3 Etapas de funcionamiento

El funcionamiento de CLUDITEM en cada ciclo de medición se divide en dos o tres etapas bien diferenciadas que se muestran en la figura 4. La primera transcurre durante T_{AE} y se ocupa del establecimiento del árbol de encaminamiento, al comienzo de cada ronda con reconfiguración de la red. Durante la segunda, que abarca un tiempo T_{DATOS} , se envían las mediciones realizadas hacia la estación base, y en la tercera que se desarrolla durante T_{SLEEP} , la red permanece inactiva. La suma de los tres tiempos mencionados es igual a T , que representa el período de medición fijo que se repite cíclicamente durante toda la vida útil de la red. En este sentido T_{SLEEP} se ajusta de forma tal que la duración de todos los

ciclos resulte siempre igual a T , independientemente de si se reconfigura y envía datos o si sólo se realiza la última tarea. La reconfiguración del árbol de encaminamiento se repite cada X ciclos, y el valor óptimo para X se determina analíticamente en la sección 4.7.3.

La definición de la estructura de encaminamiento se aborda n veces durante la vida útil de la red, de manera de producir una rotación completa del rol de CH entre todos los nodos en cada cuadrícula de la grilla virtual, siendo n el número de dispositivos por celda. La vida útil del sistema se puede calcular como indica la expresión (1).

$$T_{vida} = n \cdot X \cdot T \quad (1)$$

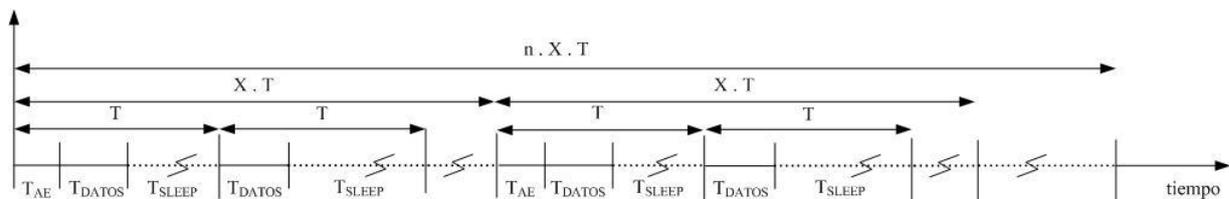


Figura 4. Etapas de funcionamiento de CLUDITEM durante la vida útil de la red.

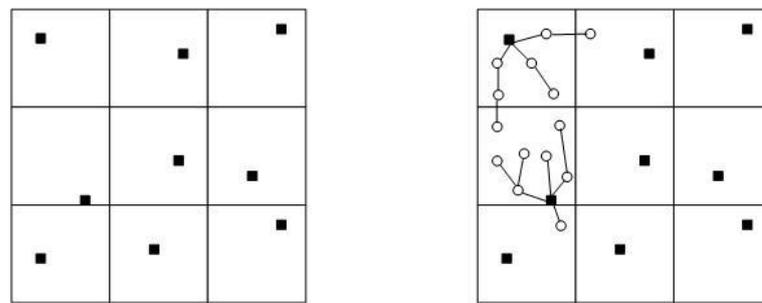
La figura 4 destaca la repetición de las distintas etapas de funcionamiento durante todo T_{vida} , mostrando además que cada ronda que incluye la reconfiguración de la estructura de encaminamiento, es seguida de $X-1$ que se ocupan exclusivamente del envío de datos hacia el *sink*. Para ajustar la duración de las etapas descriptas, se utilizaron un conjunto de temporizadores relacionados con las distintas tareas que CLUDITEM debe abordar para recolectar información del ambiente y hacerla llegar hasta la estación base.

4.4 Etapa de definición del árbol de encaminamiento

El árbol de encaminamiento jerárquico posee dos niveles y se define durante la primera etapa de funcionamiento de CLUDITEM, cuya duración es T_{AE} . El primer nivel de encaminamiento, cuya definición abarca el tiempo T_{EC} , consiste en la estructura de *clusters* de la red. El segundo nivel, definido durante T_{AC} , corresponde al árbol de CH utilizado por los nodos cabeceras para enviar los datos agregados hasta la estación base. La definición de estos dos niveles del árbol de encaminamiento se realiza cada vez que se rotan las cabeceras de *cluster*.

4.4.1 Definición de clusters

La definición de los *clusters* se inicia con la elección de los CH. Se trata de un aspecto importante en el diseño del algoritmo, ya que lograr una buena distribución y una cantidad adecuada de cabeceras es un hito fundamental del algoritmo en vistas a que la red pueda desempeñar de forma correcta sus funciones. CLUDITEM divide el área a supervisar sobre la base de una grilla virtual que utiliza durante el proceso de elección de CH y definición de *clusters*, con el objetivo de implementar una adecuada gestión de la estructura de los grupos de nodos, acorde con los requerimientos de las aplicaciones elegidas.



(a) Cuadrícula genérica con CH (b) Ejemplo de formación de *clusters*

Figura 5. Uso de una cuadrícula virtual para la definición de clusters

En la figura 5 (a) se muestra la cuadrícula virtual, en la cual el número y dimensiones de las celdas que la constituyen, dependerán de la aplicación concreta. Se asume que en cada división se distribuirá un número n de nodos, y el proceso de selección de cabeceras se basa en la idea de que en cada rotación idealmente uno de ellos asume el rol de CH. Las cabeceras se postulan respetando un esquema de tipo TDMA. Los nodos comunes adhieren a un CH y eligen su nodo de enlace (NDE), ya que la comunicación intracluster es multisalto. Es importante destacar, que un nodo común que está en una división de la cuadrícula puede sumarse a un *cluster* cuya cabecera pertenece a otra, si obtiene una situación más ventajosa (como por ejemplo, nivel de profundidad en la red). En este caso, un *cluster* quedará formado por nodos que pertenecen a más de una división como se muestra en la figura 5 (b).

El mensaje que circula durante esta fase, denominado mensaje de estructura de *cluster* (EC), tiene el formato mostrado en la figura 6. Cada nodo que escucha a un CH, y decide unirse al *cluster* que dicho CH coordina, reenvía el mensaje reemplazando los campos emisor y nivel del emisor con sus datos, de forma que otros vecinos puedan adoptarlo como NDE. El nivel del nodo en un *cluster* representa la cantidad de saltos que lo separan de su cabecera. Por lo tanto, al elegir un NDE cada nodo define su nivel sumando 1 al nivel de su enlace. Como es de esperar, los CH tienen nivel 0.

Tipo de mensaje	CH elegido	Emisor	Nivel del emisor
-----------------	------------	--------	------------------

Figura 6. Mensaje para la definición de los *clusters* (EC)

El comportamiento de un nodo durante la definición de los *clusters* se detalla en la figura 7. En la misma se muestra que cada nodo de una cuadrícula se postula como CH, respetando el TDMA definido, sólo si se cumplen las condiciones de postulación. Inicialmente, se adoptó como condición para que un nodo se proponga como CH, que carezca de enlace al momento de postularse. Esta regla hubiera resultado efectiva si los relojes de los nodos estuvieran perfectamente sincronizados, y las postulaciones en las distintas celdas resultaran simultáneas. Sin embargo, no es realista suponer la existencia de un tiempo global para la red, y cada nodo posee un tiempo local desfasado respecto de sus vecinos. Esta situación ocasiona que al postularse el primer CH, un gran número de nodos se unen a su *cluster*, resultando una red con muy pocos nodos cabecera. Por lo tanto, la estructura de *clusters* que se define a partir de esta regla es inaceptable, ya que no permite obtener una distribución uniforme de CH en el área supervisada. Como consecuencia, el proceso de definición de *clusters* se modificó estableciendo una nueva regla para postulación de los nodos. De acuerdo al TDMA definido, el nodo se propone como cabecera si carece de NDE, o posee un nivel mayor a 1 y su NDE corresponde a una celda distinta a la suya. La primera condición es para no quedar desconectado, mientras que la segunda apunta a privilegiar la elección de cabeceras en el mismo cuadro, a no ser que el CH elegido sea muy cercano (nivel 1). Otro aspecto de importancia relacionado con las condiciones establecidas, se refiere a la cantidad de postulantes por celda de la grilla que se permiten en cada

reconfiguración de la red. En el capítulo referido a simulación del algoritmo, se analizan las estructuras de *clusters* logradas para distintos valores de este parámetro con el objetivo de elegir el más conveniente de acuerdo con las especificaciones.

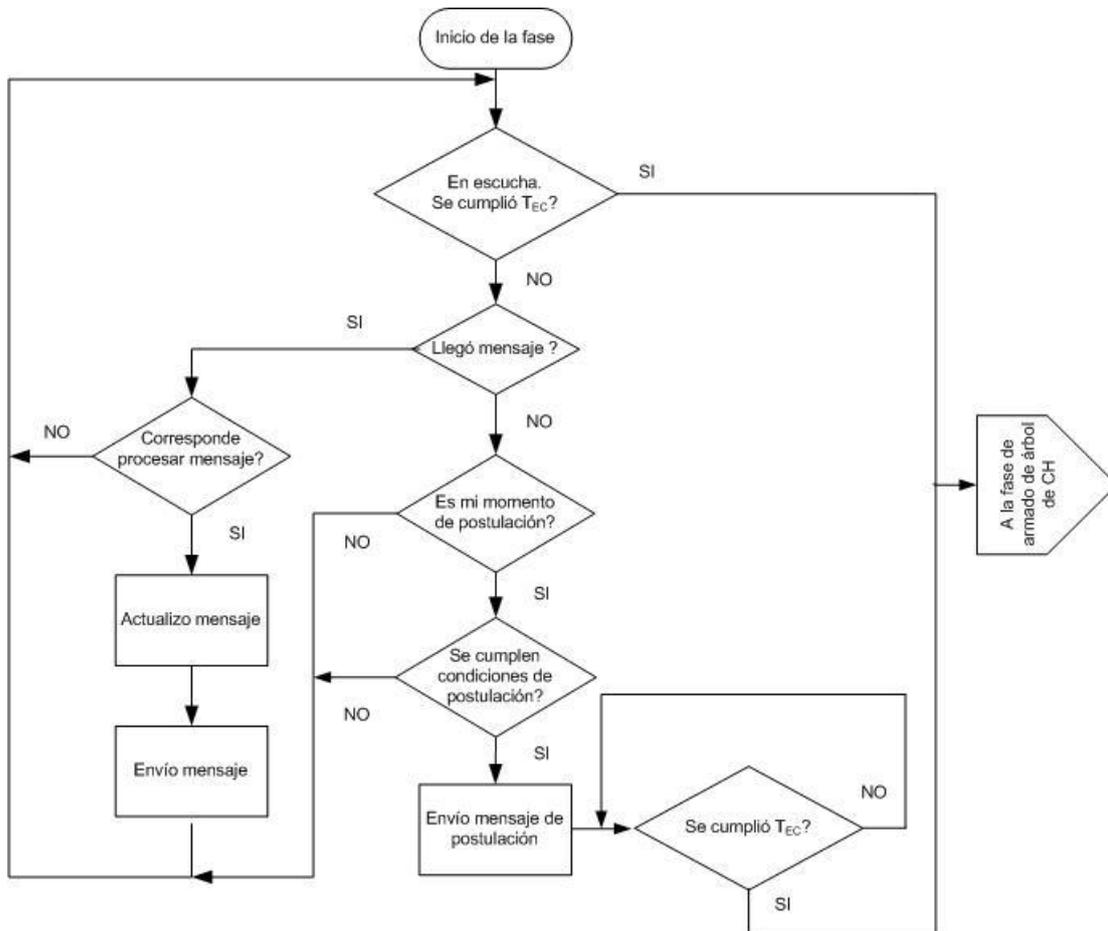


Figura 7. Comportamiento de un nodo durante la fase de definición de *clusters*.

Las condiciones bajo las cuales un nodo común decide unirse a un *cluster*, también fueron ajustadas luego de las pruebas de funcionamiento del algoritmo. Esto era previsible pues ambas condiciones, las correspondientes a postulación de CH y las correspondientes a procesamiento de mensajes EC, tienen un fuerte impacto sobre la definición de los *clusters*. La situación ideal, corresponde a un único CH por celda, con *clusters* aproximadamente uniformes respecto al número de integrantes. Esto plasma la idea de una distribución adecuada de cabeceras en el área bajo estudio, con una carga de trabajo intracluster similar. Se comenzó trabajando con la hipótesis de que un nodo adoptaría un CH, si podía mejorar

su nivel en el *cluster*. Esta condición se ajustó a medida que se avanzó con las pruebas de funcionamiento, quedando finalmente definidas las condiciones de procesamiento de un mensaje EC por parte de un nodo de la red como se resume en la figura 8.

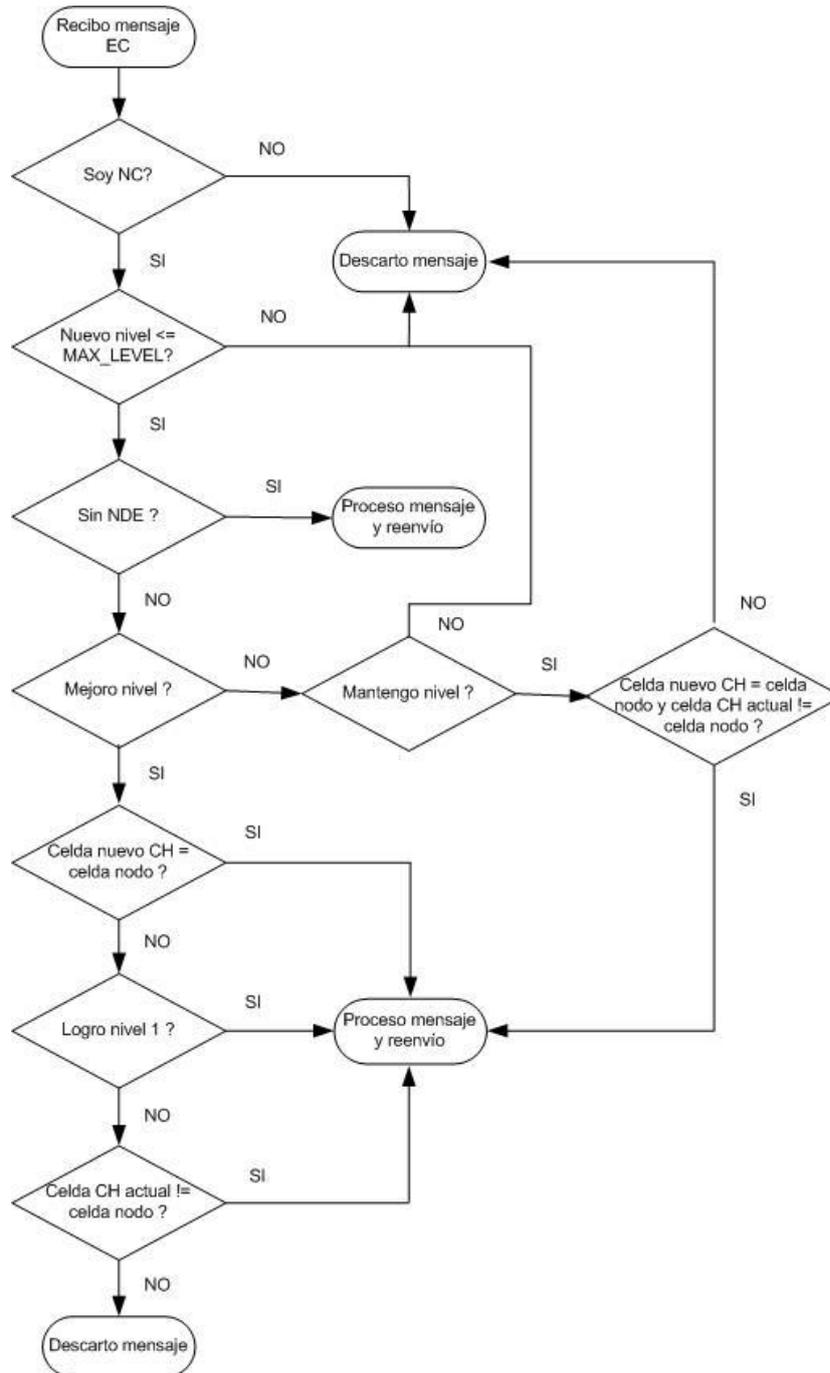


Figura 8. Condiciones de procesamiento de mensajes EC

Los nodos comunes son los únicos que procesan los mensajes EC para decidir a que *cluster* se unen. En la decisión es determinante el hecho de que estén desconectados y el nivel que logran al adoptar una cabecera, ya que se limitó la profundidad máxima con el objetivo de balancear la carga de trabajo de los *clusters*. Las otras condiciones adoptadas, tienden a privilegiar la elección de cabeceras en la misma cuadrícula, salvo que el nodo consiga una situación muy conveniente (nivel 1) en un *cluster* coordinado por un CH de otra celda. La preferencia de un CH de la misma celda apunta a la idea de que cada coordinador reúna las mediciones de nodos de la misma zona física.

La combinación de las condiciones de postulación a CH, cantidad de postulantes admitidos por celda, profundidad máxima del cluster y las reglas de procesamiento de mensajes EC, permitieron obtener *clusters* con una estructura que se acercaba a la deseada, como se detalla en la sección 5.5 de simulación de funcionamiento.

Es importante remarcar, que cada nodo sólo es consciente del cuadro al que pertenece, no conoce sus propias coordenadas ni la ubicación relativa de sus vecinos, las cuales tienen una influencia determinante en los resultados que se obtienen. Por estos motivos, se trabajó para lograr la definición de *clusters* cuyas características se aproximaran a las deseadas, utilizando los resultados de simulación obtenidos con un escenario compatible con las aplicaciones elegidas. Fue en base a estos resultados que se ajustaron las condiciones que definían el comportamiento de los nodos en esta fase.

4.4.2 Rotación de cabeceras

Los nodos que constituyen la red se ubican en forma manual y se conoce de antemano en qué cuadro de la grilla están colocados. La cantidad de divisiones de la grilla y las dimensiones de cada una dependen de la aplicación, pero en cada cuadro se distribuyen n nodos diferenciados entre sí por un ID. El identificador se divide en dos partes, una corresponde a la división o celda de la grilla en la que se encuentra el nodo, y la segunda al dispositivo propiamente dicho. Las postulaciones en cada cuadro se realizan intentando lograr la situación ideal de un único CH por división de la grilla.

El rol de cabecera de *cluster* es exigente respecto del consumo de energía, por lo cual se hace necesario rotarlo periódicamente con el fin de distribuir el consumo en forma más uniforme. Cuando un nodo asume este rol lo cumple durante una cierta cantidad de rondas de envío de datos (X) y luego se realiza la rotación de CH. Si las cabeceras mantienen su rol demasiado tiempo, se agotan y se producen desconexiones en la red. Si X es demasiado pequeño, se consume mucha energía en el rearmado del árbol de encaminamiento. Por lo tanto, debe existir un valor de X que equilibre el gasto de energía en la red [4]. El valor de X se obtiene en la sección 4.6.3, analizando el consumo de los nodos, que asumen distintos roles durante el tiempo de vida del sistema.

4.4.3 Análisis de tiempos para la definición de clusters

El cálculo de los tiempos que los nodos con distintos roles deberán estar despiertos para cumplir con las tareas propias de las etapas de trabajo del algoritmo, se realiza teniendo en cuenta el desfasaje máximo de relojes locales T_{DIC} . La fase de armado de *clusters* es la primera de una ronda con reconfiguración, luego de que los nodos han estado inactivos hasta cumplir el período T . Por lo tanto, al despertar lo harán en tiempos distantes entre sí a lo sumo en T_{DIC} , que debe ser tenido en cuenta para definir el inicio de las tareas de la fase. Un nodo al que le corresponde postularse para CH, lo hará durante esta fase luego de iniciar las actividades, calculando el momento de envío de su postulación (T_{Pos_i}) de acuerdo a la fórmula (2).

$$T_{Pos_i} = T_P \cdot K_{Pi} \quad (2)$$

En la misma, T_P es el tiempo correspondiente a una postulación, y K_{Pi} identifica el *slot* de postulación correspondiente al nodo i . El valor de T_P está asociado con el tiempo necesario para transmitir el mensaje de postulación, considerando la máxima cantidad de saltos en el *cluster*, y puede calcularse como (3).

$$T_P = SM \cdot (T_M + T_{MAC}) \quad (3)$$

En (3), SM es la máxima cantidad de saltos intracluster, T_M es el tiempo de transmisión de una trama y T_{MAC} contempla el tiempo de acceso al medio.

El tiempo total de la etapa de definición de la estructura de *clusters* (T_{EC}) puede definirse a partir del tiempo que se invierte en una postulación como indica la expresión (4):

$$T_{EC} = T_P \cdot K_{P_{MAX}} + 2 \cdot T_{DIC} \quad (4)$$

En (4), el parámetro $K_{P_{MAX}}$ representa la máxima cantidad de postulaciones que se permitirán en la fase. Cuando un nodo se despierta al inicio de la fase, espera T_{DIC} antes de iniciar el conteo correspondiente a su momento de postulación. Esto es necesario para garantizar que todos sus vecinos estarán despiertos al momento de proponerse como CH, como se muestra en la figura 9, donde se presenta la situación de dos nodos de una división con máximo desfase de relojes.

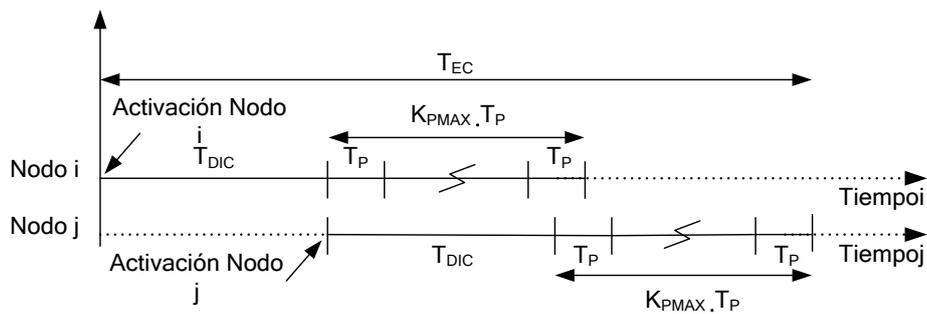


Figura 9. Tiempos locales a los nodos en la fase de definición de *clusters*

Como primera aproximación, se estableció que la fase de armado de *clusters* duraría el tiempo suficiente como para que potencialmente los n nodos de la cuadrícula se postularan. Esta situación aseguraba que no quedaban nodos desconectados en las celdas. Cuando se simuló el funcionamiento de la fase, se encontró que el número de nodos sin conexión al cabo de pocas postulaciones, era muy bajo. Se concluyó entonces, que no era necesario esperar n períodos de posibles postulaciones para considerar los *clusters* estructurados. La reducción del tiempo de postulaciones resulta beneficiosa pues permite ahorrar energía. En el capítulo 5 sobre simulación del funcionamiento de CLUDITEM, se analiza este ítem y se detallan las decisiones tomadas.

4.4.4 Definición del árbol de cabeceras

El segundo nivel del árbol de encaminamiento define la ruta de comunicación al *sink* de todas las cabeceras de *cluster*, con el objetivo de que la información agregada alcance la estación base. La comunicación entre las cabeceras y el *sink* puede resultar multisalto, ya que el área a supervisar es de tamaño variable y el radio de transmisión de los CH está limitado.

Esta fase se inicia con un mensaje que envía la estación base hacia la red, con un radio de transmisión igual al que utilizan los CH para comunicarse entre sí. Los nodos que escuchan en forma directa dicho mensaje lo registran. Entre estos últimos, aquellos que asumieron el rol de cabecera durante la fase de definición de *clusters*, envían un mensaje de armado del árbol de CH (ACH), anunciando que la estación base está dentro de su radio de transmisión y que por lo tanto su nivel es 1. La estructura del mensaje ACH se muestra en la figura 10, y a partir de los datos que incluye, cada CH elige su nodo de enlace como la cabecera que escucha con menor nivel. Luego, reenvía el mensaje ACH comunicando su ID y su nivel para permitir que otros nodos lo elijan como NDE en el árbol.

Tipo de mensaje	Emisor	Nivel del emisor
-----------------	--------	------------------

Figura 10. Formato del mensaje ACH

Un CH que no escucha a ningún vecino queda como cabecera sin enlace. Esta situación puede tener dos causas, una por corrupción de mensajes, y la otra por utilizar una potencia de transmisión insuficiente. El último caso se previene proponiendo una potencia que garantice la comunicación entre los cabeceras y los CH de nivel uno con la estación base, como se detalla en la sección 4.7.2 sobre potencia de transmisión intercluster. Por lo tanto, sólo persiste la primera situación como causante del problema, y debe ser tomada en cuenta.

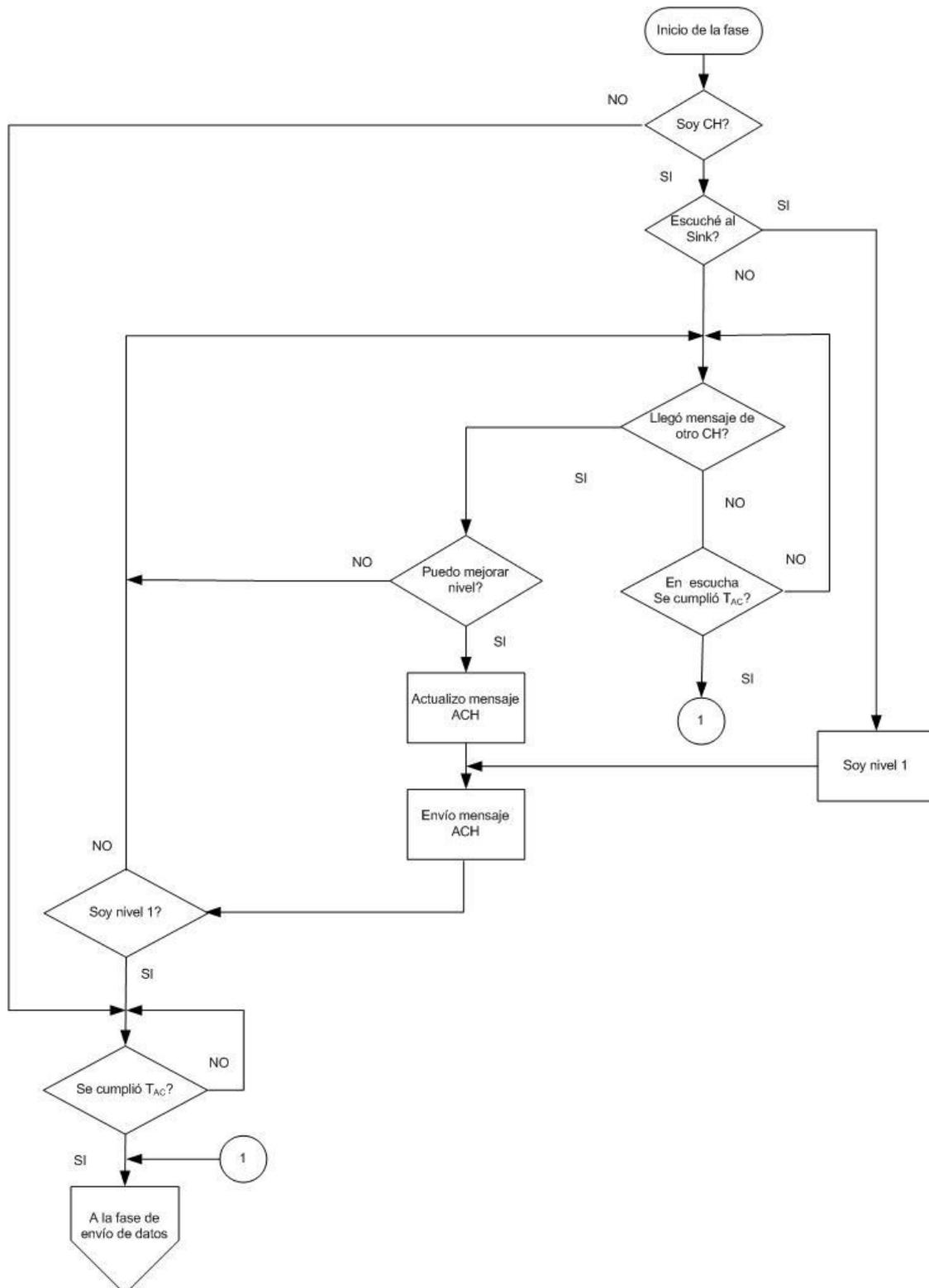


Figura 11. Comportamiento de un nodo durante el armado del árbol de CH

La desconexión de un CH por congestión en la red, constituye una falla grave ya que cada cabecera es la encargada de transmitir la información de todos los miembros de su *cluster*. Por este motivo, en la sección 4.6.3 que trata el envío de datos agregados se propone una solución para este problema, con el fin de prevenir la pérdida de la información colectada. El comportamiento de un nodo durante la fase de armado del árbol de cabeceras, anteriormente descrito, se resume en el diagrama de flujo de la figura 11.

Es importante destacar que un CH puede recibir mensajes ACH más de una vez desde diferentes nodos cabecera. Cada vez que se presenta esta situación, analiza si puede mejorar su nivel en el árbol de CH. Si su nivel no mejora desecha el mensaje, en caso contrario adopta un nuevo NDE, actualiza su nivel y lo anuncia mediante un nuevo mensaje ACH, brindando a otras cabeceras la oportunidad de adoptarlo como enlace para el envío de sus agregados hacia la estación base.

Luego de la circulación de los mensajes ACH entre los distintos nodos cabecera, queda perfectamente definida la estructura de comunicación de los CH. Al finalizar la definición de los dos niveles del árbol de encaminamiento, se procede al envío de los datos colectados en el área bajo estudio.

4.4.5 *Análisis de tiempos para la definición del árbol de CH*

El cálculo de la duración de esta fase se realiza sin tener en cuenta el desfasaje de relojes, ya que los nodos están activos desde el inicio de la fase de definición de *clusters*. Por lo tanto, el tiempo que insume el armado del árbol de cabeceras (T_{AC}) se estima utilizando la expresión (5).

$$T_{AC} = P \cdot M \cdot (T_M + T_{MAC}) \quad (5)$$

P: profundidad máxima del árbol

M: número estimado de mensajes enviados por cada CH

T_M : tiempo para transmitir una trama

T_{MAC} : tiempo de acceso al medio

Durante el armado del árbol de cabeceras, los CH transmiten un número variable de mensajes ACH. Esta cantidad está relacionada con los mensajes recibidos y la posibilidad de mejorar su nivel. Por lo tanto se define para el cálculo una constante M , que representa el número estimado de mensajes que envía cada CH. La máxima profundidad del árbol también se estima en base al número de cuadros de la grilla, que se definirá para cada aplicación. Una vez cumplida la fase de armado del árbol de CH, se procede al envío de datos hacia la estación base.

4.5 Etapa de Envío de datos

La etapa de envío de datos que se desarrolla durante T_{DATOS} se realiza en dos fases: en la primera, de duración T_{ED} , los nodos comunes envían sus mediciones hacia su cabecera de *cluster*, y en la segunda que transcurre durante T_{DA} , los CH utilizan el árbol de cabeceras para hacer llegar hasta la estación base el mensaje agregado, que resume la información recolectada por el *cluster* que coordinan. Esta fase se desarrolla a continuación de la definición del árbol de encaminamiento (T_{AE}) si se trata de una ronda de reconfiguración, o al inicio del período T de recolección de información en una ronda de transmisión exclusiva de mediciones.

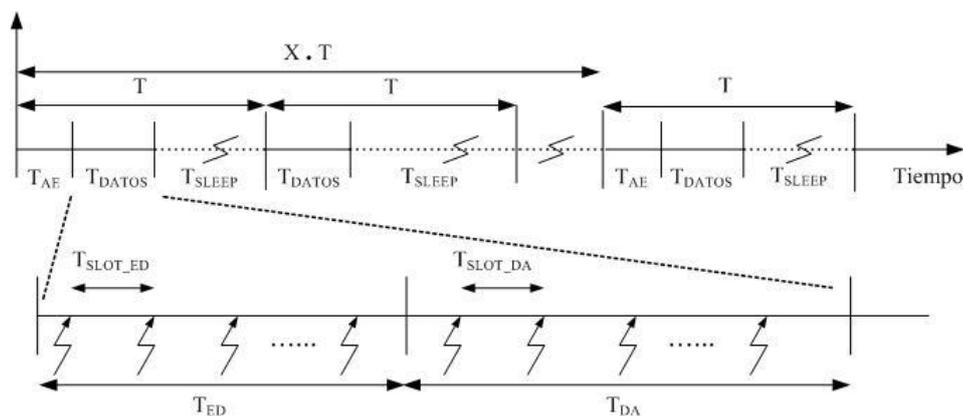


Figura 12. Funcionamiento de CLUDITEM. Detalle de las fases de envío de datos

La figura 12 muestra las fases de transmisión de información y destaca que el envío de datos intra e intercluster se realiza respetando en cada uno de ellos un esquema de tipo

TDMA. En la misma T_{SLOT_ED} y T_{SLOT_DA} representan los *slots* de tiempo adoptados para el envío de datos y datos agregados respectivamente.

4.5.1 Envío de datos intracuster

En cada período de recolección de datos T , los NC envían las mediciones realizadas a su cabecera utilizando un mensaje de datos, cuya estructura se muestra en la figura 13.

Tipo de mensaje	Origen de los datos	Nodo de enlace	Datos
-----------------	---------------------	----------------	-------

Figura 13. Formato del mensaje de datos.

El envío de los valores medidos por cada nodo común se realiza en base a un esquema de tipo TDMA. Se adopta este criterio para reducir las eventuales colisiones que podrían producirse si varios nodos enviaran su información al mismo tiempo.

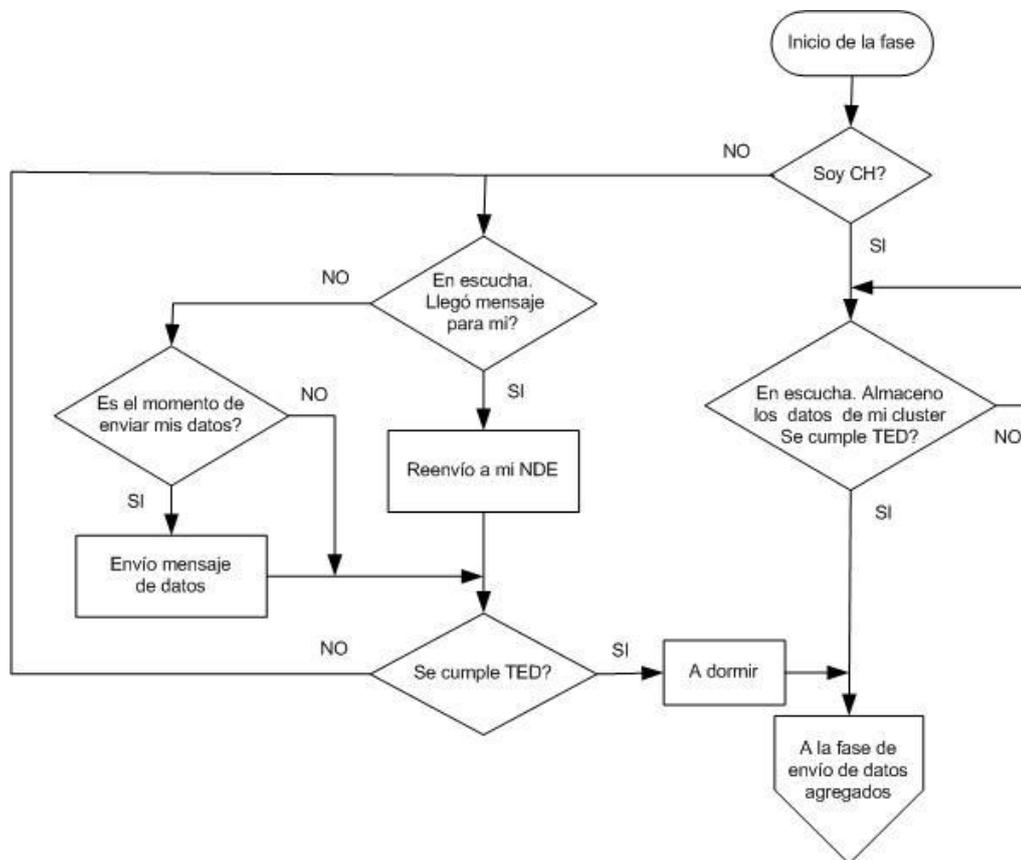


Figura 14. Comportamiento de un nodo en la fase de envío de datos intracuster

Cada nodo común que recibe el mensaje y verifica que su ID coincide con el campo nodo de enlace, coloca en dicho campo su propio NDE y reenvía el mensaje en forma inmediata. El origen de los datos es importante para que el CH sepa cuántos nodos reportan datos en esa ronda. Si bien los CH no son conscientes de la cantidad de nodos del *cluster* (que no coincide necesariamente con la cuadrícula), esta información puede ser de interés para la definición del mensaje agregado, o para establecer el porcentaje de pérdida de información por zona. Además, si ha llegado el momento de envío de sus propios datos, determinado por el esquema TDMA adoptado, el nodo los transmite utilizando el formato de mensaje de datos. Una vez finalizada la etapa de envío de mediciones hacia las cabeceras, los nodos comunes, que han cumplido con sus responsabilidades en la ronda, adoptan el estado de bajo consumo. El comportamiento de un nodo de la red durante la fase de envío de mediciones intracluster se resume en el diagrama de la figura 14.

4.5.2 *Análisis de tiempos para el envío de datos intracluster*

Cada nodo miembro de un *cluster* enviará los valores medidos en un período de recolección de datos T , en el momento indicado por la expresión (6), a partir del inicio del envío de mediciones durante la fase de datos intracluster.

$$T_{envio_i} = T_{SLOT_ED} \cdot K_i \quad (6)$$

$$T_{SLOT_ED} = SM \cdot (T_M + T_{MAC}) \quad (7)$$

En la misma, T_{SLOT_ED} es un período suficiente para reducir la posibilidad de colisiones y se obtiene en base al tiempo estimado de envío de un mensaje de datos desde un nodo hoja hasta su CH a partir de la expresión (7). En su definición, interviene el tiempo de envío del mensaje T_M , el tiempo de acceso al medio T_{MAC} y la máxima cantidad de saltos intracluster SM . El factor K_i de (6), puede fijarse con distintos criterios, y su definición se establece a partir de los resultados de simulación en el capítulo 5. Por lo tanto, el máximo período durante el cual los NC estarán enviando sus mediciones hacia sus respectivos CH queda definido por la expresión (8). En la misma, N representa el número máximo de *slots* adoptados para la fase de envío de datos intracluster.

$$T_{envio_{MAX}} = T_{SLOT_ED} \cdot N \quad (8)$$

El tiempo necesario para que los datos medidos por los nodos comunes lleguen hasta el CH (T_{ED}), se obtiene mediante la expresión (9) que utiliza las fórmulas anteriores, e incorpora el máximo desfasaje de relojes intracluster T_{DIC} . Este último parámetro contempla el caso en que los nodos estén apagados y activen sus transceptores para enviar los datos, de acuerdo con sus relojes locales. Esta situación se presenta en las fases de datos correspondientes a rondas sin reconfiguración de la red. En las mismas, los nodos se activan e inician la fase de datos intracluster en forma inmediata.

$$T_{ED} = 2 \cdot T_{DIC} + T_{envio_{MAX}} \quad (9)$$

4.5.3 Envío de datos intercluster

En esta fase, cada CH procesa los mensajes recibidos desde los miembros de su *cluster* de acuerdo con la función de agregación elegida a partir de los requerimientos de la aplicación. Luego envía la información agregada a su NDE en el árbol de cabeceras con el fin de hacerla llegar a la estación base. Los nodos comunes no participan de las actividades y duermen hasta el siguiente período T . El formato del mensaje que circula entre los CH es el mostrado en la figura 15.

Tipo de mensaje	Origen de los datos	Nodo de enlace	Datos agregados
-----------------	---------------------	----------------	-----------------

Figura 15. Formato del mensaje de datos agregados.

Las cabeceras que escuchan un mensaje agregado comparan su ID con el valor del campo nodo de enlace. Si coincide lo reenvían a su propio NDE, en caso contrario lo descartan. Si algún CH está desconectado del árbol de cabeceras, enviará su mensaje agregado con un código de ayuda en el campo nodo de enlace. Todas las cabeceras que escuchen un mensaje agregado que contenga el código de ayuda, lo reenviarán a sus NDE. De esta forma los agregados de los CH sin enlace llegarán a la estación base, que será la responsable de filtrar eventuales repeticiones, a partir del campo origen de datos.

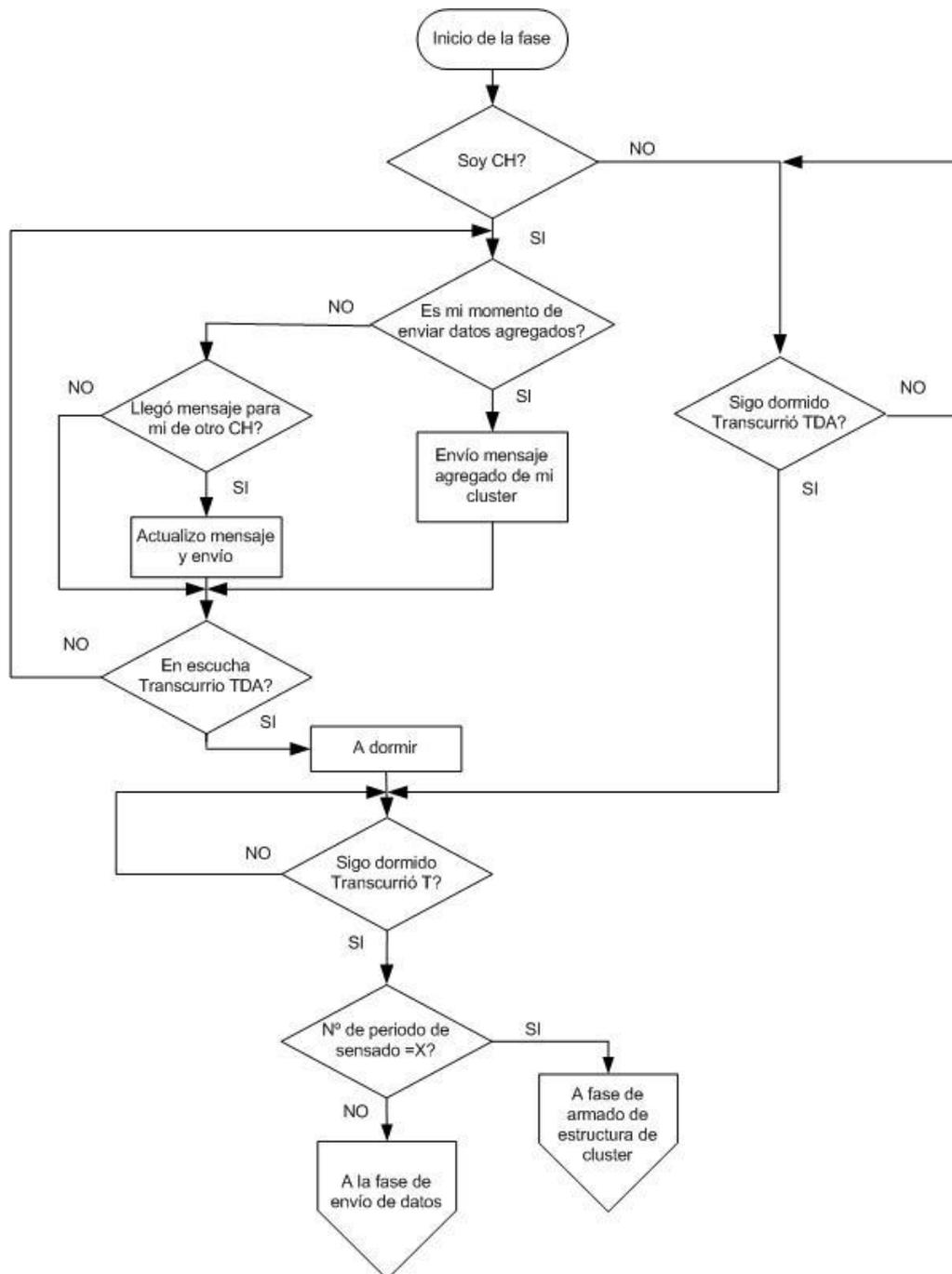


Figura 16. Comportamiento de un nodo en la fase de envío de datos intercluster

Los nodos que participan en esta etapa son aquellos que se han postulado para CH, que según se espera serán aproximadamente uno por cuadro de la grilla. La potencia de transmisión será mayor a la correspondiente a la fase de envío de datos intracluster, por lo

tanto, pese a la disminución de nodos participantes, pueden producirse colisiones que degradan la cantidad de mensajes que llegan hasta la estación base. Esta situación se verificó por simulación, y por este motivo también se incluyó un esquema de tipo TDMA para esta fase de envío de mediciones. La definición de los valores de los *slots* del esquema se analiza en el capítulo 5 de simulación del funcionamiento del algoritmo.

Una vez finalizada la etapa todos los CH duermen esperando el cumplimiento del período de adquisición de datos T . Cuando esto ocurre, si se han cumplido las X rondas necesarias para realizar la rotación de cabeceras, se aborda una nueva definición del árbol de encaminamiento, en caso contrario se inicia una nueva etapa de envío de datos. La figura 16 resume el comportamiento de un nodo durante la fase de envío de datos agregados.

4.5.4 *Análisis de tiempos para el envío de datos intercluster*

El tiempo necesario para cumplir con las tareas asociadas con el envío de datos agregados queda definido por las expresiones (10) y (11). En (10), que establece el momento de envío de datos agregados de cada CH a partir del inicio de la fase, intervienen T_{SLOT_DA} que es el tiempo estimado de envío de una trama de datos agregados, y KAg_i que depende del esquema TDMA. Por su parte, (11) calcula la duración total de la fase, que tiene en cuenta el número máximo de *slots* de envío de datos agregados NAg . Los valores de KAg_i y NAg se definen a partir del esquema TDMA que se adopta en base a los resultados de simulación que se describen en la sección 5.7.

$$T_{envioAg_i} = T_{SLOT_DA} \cdot KAg_i \quad (10)$$

$$T_{DA} = T_{SLOT_DA} \cdot NAg \quad (11)$$

4.5.5 *Características de la agregación de datos*

CLUDITEM, se diseñó para aplicaciones con adquisición periódica de información sobre toda el área supervisada. La clasificación propuesta en [66], divide las técnicas de agregación utilizadas por estos algoritmos en tres categorías, de acuerdo con las características del período de espera que cumplen los CH durante la recepción de los mensajes de datos de sus hijos. En este sentido, CLUDITEM incorpora una técnica de

agregación periódica simple, ya que luego del cumplimiento de un período fijo, cada CH aplica la función de agregación elegida a las mediciones colectadas y transmite el mensaje agregado a su enlace en el árbol de cabeceras. Por lo tanto, si algún nodo hijo no se reporta, sus datos no serán procesados y no llegarán a la estación base. Se debe recordar que las cabeceras no son conscientes de quienes son los miembros del *cluster* que coordinan, y por lo tanto no pueden saber si todos sus hijos se han reportado en un determinado momento de la fase.

El retardo o *delay* introducido al aplicar agregación y enviar la información resultante hacia la estación base no es una característica importante para las aplicaciones seleccionadas. La QoS se define en base a la cantidad de mediciones recibidas por el *sink*, favoreciendo la recolección de información por zona a partir de la gestión de la estructura de *clusters*. Como los mensajes agregados resumen la información de cada zona de interés, es fundamental evitar su pérdida, y en este sentido se han implementado esquemas TDMA intra e intercluster para prevenir las colisiones y un mecanismo de reenvío de agregados si algún CH queda desconectado del árbol de cabeceras.

La función de agregación elegida contempla el envío de todas las mediciones realizadas, ya que se enfoca a lograr un mapa que refleje las variaciones de las variables estudiadas en cada período de recolección. Esta función, se traduce en un mensaje agregado que reúne los valores adquiridos, y en simulación se ha representado mediante la concatenación de los ID de los nodos que reportan a cada CH. Por este motivo, la longitud en bytes del agregado es significativamente mayor que la de un mensaje de datos de un nodo común, pero su circulación en la red está mucho más acotada. Las características de la función de agregación, de acuerdo con el análisis realizado en el capítulo 3, incluyen:

- *Sensibilidad a duplicados:* Los CH filtran duplicados en cada *cluster*, ya que sólo reciben los mensajes de datos que se les envían a ellos (su ID está en el campo NDE del mensaje), si escuchan mensajes de datos que no les están dirigidos, los desechan. La estación base puede recibir agregados duplicados causados por el mecanismo de reenvío de mensajes de CH desconectados, pero es capaz de filtrarlos a partir del campo origen del mensaje.

- *Monotonicidad:* La información agregada siempre tiene un valor. Aún cuando el CH sólo envíe sus propias mediciones, siempre transmitirá información de interés. Si la recepción de mensajes de datos es exitosa, la cantidad de información del agregado será cada vez más significativa, ya que reunirá las mediciones de su zona de influencia.
- *Dato vs. cálculo:* Las cabeceras no son conscientes de los hijos que deben reportarles información, pero todos los datos son importantes y el único procesamiento consiste en concatenarlos en un único mensaje. Por lo tanto, si el ambiente es muy ruidoso y la pérdida de mensajes es alta, la QoS de la aplicación se resentirá notablemente.
- *Requerimientos de estado parcial:* Es muy exigente, se deben enviar todas las mediciones recibidas.

Puesto que la función de agregación elegida se caracteriza por no presentar pérdida de información, la estación base puede reconstruir las mediciones originales a partir de los agregados recibidos. El hecho de que cada CH pueda definir su mensaje agregado a partir de los datos de sus hijos permite disminuir la cantidad de mensajes circulantes en la red. Esto redundará en beneficios significativos respecto del consumo de energía, como han demostrado He et al. en [32].

En aplicaciones en las que las variables que se reporten estén, además, altamente correlacionadas se podrían implementar funciones de agregación que aprovechen esta característica. Con ello se emitirían mensajes agregados que concatenen un menor número de datos, o que representen la información de manera más concisa, optimizando aún más el consumo de energía.

4.6 Modelo de energía

Una red de sensores consume energía al realizar tres actividades: medición de los parámetros del ambiente, procesamiento de información, y comunicación entre los dispositivos para hacer llegar los datos hasta la estación base. El consumo asociado con las tareas de comunicación entre los nodos se consideró la principal causa de agotamiento de la

energía disponible en cada dispositivo, como se asume en la mayoría de las publicaciones del área [9]. Por este motivo, para el análisis del tiempo de vida del sistema y el cálculo de los parámetros del algoritmo, no se tuvieron en cuenta los consumos asociados con mediciones y procesamiento de los datos colectados.

Los transceptores de los nodos, que son la única fuente de consumo considerada, pueden asumir uno de los siguientes estados: transmisión, recepción, en escucha y apagado. En transmisión, el nodo ha ganado acceso al medio y envía información a la red. En recepción, detecta la llegada de un mensaje enviado por un vecino, en estado de escucha permanece analizando el medio y cuando se apaga, no realiza actividad alguna consumiendo el mínimo de energía posible [8]. Por lo tanto, para el análisis realizado se tuvieron en cuenta los consumos asociados con los estados de los transceptores y el tiempo que permanecen en cada uno de ellos, de acuerdo con su rol y la etapa de funcionamiento del algoritmo en la que se encuentren.

El intercambio de mensajes se realiza en dos niveles, el primero entre los miembros de cada *cluster*, y el segundo entre las cabeceras. Por lo tanto, se trabajó con dos radios de transmisión diferentes, uno más reducido para las comunicaciones intracluster, y otro de mayor alcance asociado con las comunicaciones entre CH.

4.6.1 Potencia de transmisión intracluster

Se adopta un radio de transmisión intracluster r de forma que las divisiones de la cuadrícula virtual tengan una longitud $2r$. La mayor distancia entre un CH y un nodo común en una celda (d), se presenta cuando el CH está un vértice y el nodo común se ubica en el vértice en diagonal, como se muestra en la figura 17. Por lo tanto, las expresiones (12) y (13) relacionan d con r :

$$d = \sqrt{(2r)^2 + (2r)^2} \quad (12)$$

$$d = 2r \cdot \sqrt{2} < 3r \quad (13)$$

Entonces, según (13) la distancia d resulta menor a 3 saltos. Como en los *clusters* pueden intervenir nodos de celdas adyacentes, no se puede asegurar que todos los nodos alcancen

su CH con una cantidad de saltos menor o igual a tres, pero como se detalla en la sección 4.4.1 se estableció una profundidad máxima para los *clusters*, y por lo tanto, el número de saltos posibles está limitado a ese valor.

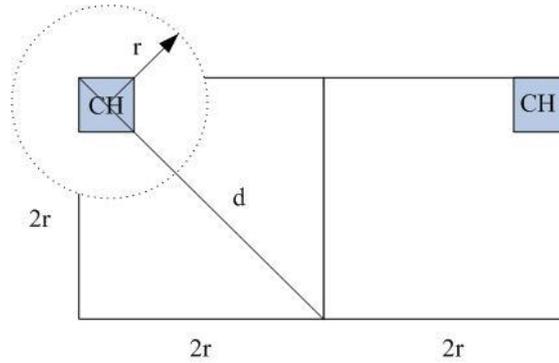


Figura 17. Radio de transmisión intracluster

4.6.2 Potencia de transmisión intercluster

El intercambio de mensajes entre los CH se realiza con una potencia mayor que la utilizada en las comunicaciones intracluster. El cálculo del radio de transmisión correspondiente R , tiene en cuenta la posibilidad de que dos cabeceras de cuadros adyacentes de la grilla virtual estén ubicadas en vértices opuestos, como se muestra en la figura 18. A partir de la misma, sabiendo que L ($2r$) es el lado de los cuadros de la grilla, se obtiene la expresión (14) que relaciona los dos radios de transmisión con los que trabaja el algoritmo:

$$R^2 = (2 \cdot 2r)^2 + (2r)^2 \quad (14)$$

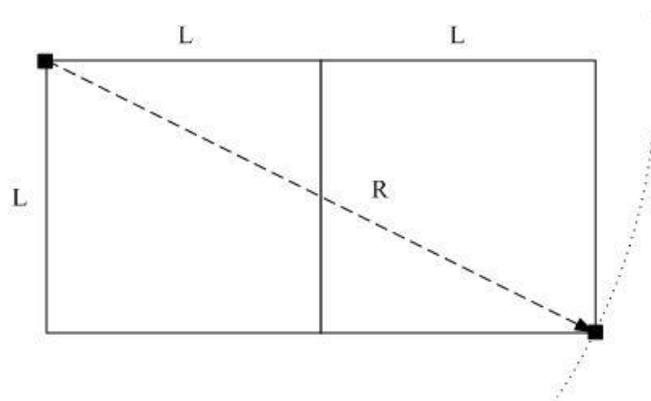


Figura 18. Radio de transmisión intercluster

En el capítulo 5 donde se simula el funcionamiento de CLUDITEM, se detallan los valores adoptados para ambos radios de transmisión, compatibilizando las consideraciones aquí realizadas con los parámetros del transceptor CC2420 de la empresa Texas Instruments [68], para el escenario de trabajo adoptado.

4.6.3 Cálculo del momento de rotación de CH

Un nodo de la red consume energía de acuerdo al estado en que se encuentra. En los estados de transmisión, recepción y escucha, el consumo resulta comparable, por el rango de potencia de transmisión que se maneja (de acuerdo con la hoja de especificaciones del circuito transceptor CC240 utilizado [68]). La energía consumida durante los períodos en que los transceptores permanecen apagados se considera despreciable para el análisis.

Es importante destacar, que la energía inicial del nodo (E_{IN}) debe ser mayor o igual que la necesaria para un número entero de rotaciones completas de CH en una celda (Z), para lograr una distribución uniforme del consumo. Esto garantiza que todos los nodos han sido CH la misma cantidad de veces (Z).

$$E_{IN} \geq Z \cdot (E_{AECH} + E_{CH} \cdot X + E_{AENC} \cdot (n-1) + E_{NC} \cdot (n-1) \cdot X) \quad (15)$$

En la expresión (15):

E_{CH} : Energía gastada por un CH para hacer llegar datos a la estación base en un período T.

E_{NC} : Energía gastada por un NC para hacer llegar datos a la estación base en un período T.

E_{AECH} : Energía gastada por un CH para armar la estructura de encaminamiento.

E_{AENC} : Energía gastada por un NC para armar la estructura de encaminamiento.

X: Número de períodos T que un nodo cumple el rol de CH.

La definición del árbol de encaminamiento implica un gasto de energía considerable, y es deseable reducir la cantidad de veces que se lleva a cabo. Por lo tanto, para lograr que todos los nodos sean CH la misma cantidad de veces y minimizar el consumo, se asume que cada nodo es una única vez cabecera en su celda. Esto significa que Z es igual a uno. En

consecuencia, a partir de la expresión anterior, se obtiene el valor más conveniente para X según (16):

$$X \leq \frac{E_{IN} - E_{AECH} - E_{AENC} \cdot (n-1)}{E_{CH} + E_{NC} \cdot (n-1)} \quad (16)$$

Es importante aclarar que existe una relación directa entre las energías utilizadas en el desarrollo anterior y los tiempos de encendido de los nodos en las distintas fases del algoritmo calculados en las secciones anteriores. Esta relación queda definida por la potencia consumida en cada uno de los estados de funcionamiento de los nodos, y los valores se obtienen a partir de los parámetros del transceptor utilizado.

Capítulo 5. Simulación del funcionamiento de CLUDITEM

5.1 *El simulador NS2*

El comportamiento del algoritmo se analizó utilizando NS2 v2.31, un simulador de eventos discretos orientado al análisis de protocolos para redes de comunicación, desarrollado en el marco del proyecto VINT (*Virtual InterNetwork Testbed*), ampliamente difundido y utilizado tanto para redes cableadas como para redes inalámbricas [54]. Se trata de una plataforma de código abierto, que incluye numerosos módulos que abarcan distintos aspectos relacionados con la tecnología de las redes de comunicaciones, y que se ha enriquecido con el aporte de diversas instituciones e investigadores del área. Este simulador se desarrolló en C++ con una interface a OTcl, que agiliza la definición de los escenarios de trabajo, facilitando la realización de pruebas de funcionamiento de los algoritmos en diferentes situaciones. En este sentido, soporta una jerarquía de clases C++ estrechamente relacionada con otra jerarquía similar en OTcl. En el ambiente NS2 los objetos y variables están representados en ambos lenguajes de forma que desde el punto de vista del usuario, existe una correspondencia uno a uno entre la jerarquía compilada y la interpretada. El lenguaje C++ se utiliza para la implementación detallada de los protocolos, brindando velocidad de funcionamiento, mientras que OTcl agiliza los cambios necesarios en los escenarios de simulación, evitando los tiempos de compilación asociados [71].

La versión del simulador utilizada implementa el estándar IEEE 802.15.4, elegido para las capas inferiores de CLUDITEM, a partir del desarrollo del módulo WPAN de Zheng y Lee [88] y la actualización de Ramachandran [62].

5.2 *Implementación de CLUDITEM en NS2*

En el ambiente NS2, se define la pila completa del protocolo de comunicaciones, lo que permite simular el comportamiento de los algoritmos de encaminamiento en condiciones cercanas a la realidad. Para las capas inferiores de CLUDITEM se eligió utilizar el estándar IEEE 802.15.4, una tecnología flexible y especialmente adecuada para redes inalámbricas de baja tasa con un radio de transmisión acotado (LR-WPAN) [9]. Este estándar especifica

las capas física y de acceso al medio de la pila de protocolos y se enfoca a disminuir la complejidad y el consumo de energía incrementando de esta forma el tiempo de vida de los dispositivos [84]. En este sentido la capa MAC del estándar se utilizó en modo *beaconless*, y CLUDITEM se incorporó al ambiente de NS2 como un agente en el nivel de encaminamiento. Por otro lado, se definió la estructura que incluye los distintos tipos de mensajes del algoritmo desarrollado, y se la agregó al archivo de *trace* que proporciona el simulador con el fin de analizar el flujo de información.

La potencia de la señal emitida P_t , se estableció a nivel de mensaje, ya que según las características de CLUDITEM, se trabaja con dos radios de transmisión, uno para la comunicación intracluster y otro para el envío de mensajes intercluster. Por este motivo, cada vez que un nodo envía información, dependiendo del tipo de mensaje involucrado, se establece el radio de transmisión que se utilizará. Esto hizo necesario acceder desde la capa de encaminamiento a la capa física, donde se define la potencia con que se transmitirá el mensaje correspondiente. En este sentido, la implementación del protocolo se esquematiza en la figura 19.

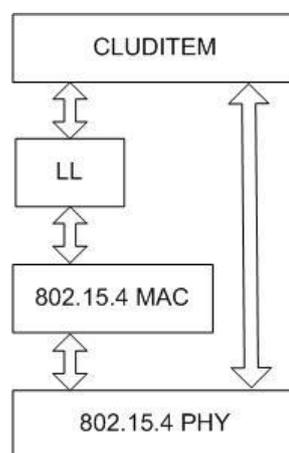


Figura 19. Esquema de la implementación del protocolo de comunicación

La configuración de las capas del protocolo implementado en los nodos de la red, se detalla en el *script* de simulación que se presenta en la figura 20.

```

#=====
# Node Configuration
#=====

set val(chan) Channel/WirelessChannel ;#channel type
set val(prop) Propagation/TwoRayGround;#radio-model
set val(netif) Phy/WirelessPhy/802_15_4;#net interface
set val(mac) Mac/802_15_4 ;#MAC type
set val(ifq) Queue/DropTail/PriQueue ;#interface queue
set val(ll) LL ;#link layer
set val(ant) Antenna/OmniAntenna ;#antenna model
set val(ifqlen) 50 ;#max packet ifq
set val(nn) $nodeNum_ ;#n°of mobilenode
set val(rp) CLUDITEM ;#routing protocol
#=====

```

Figura 20. Configuración de las capas del protocolo de comunicaciones en un nodo CLUDITEM

Para distribuir las tareas de las distintas fases se incorporaron temporizadores en cada nodo, asumiendo un desfase máximo de relojes locales. Los períodos asociados a estos temporizadores se definieron como parámetros para permitir el ajuste del funcionamiento del algoritmo. Durante la simulación, cada nodo va trabajando de acuerdo con el cronograma de tareas asociadas con sus temporizadores locales, e inicia el conteo de otros, para cumplir adecuadamente todas las fases de cada ronda de adquisición de datos. Todos los nodos de la red son homogéneos en cuanto a energía inicial y recursos, pero cumplen distintas tareas dependiendo del rol que asumen en cada reconfiguración del árbol de encaminamiento. El *script* de simulación de la figura 21 muestra la definición de parámetros del agente CLUDITEM. En el mismo, el número máximo de rotaciones se estableció asociando el tiempo de vida de la red con una rotación completa de roles en cada celda. El valor de X (rondas_para_rotación), se definió en cada simulación de acuerdo con la energía inicial de los dispositivos y los consumos en cada fase, respetando la expresión (16).

La estación base se definió como un nodo especial, con reservas de energía que no se agotan durante la simulación y cuyas funciones están centradas en:

- Iniciar la fase de armado del árbol de cabeceras con el envío de un mensaje que permite a los CH que lo escuchan asumir el nivel 1 en dicha estructura.

- Acumular los mensajes agregados que le llegan desde las cabeceras en cada ronda de adquisición de datos.

```

#=====
# Configuracion del agente CLUDITEM
# #=====
Agent/CLUDITEM max_rotaciones $nodos_por_celda
Agent/CLUDITEM nodos_por_celda $nodos_por_celda
Agent/CLUDITEM nro_postulantes 2
Agent/CLUDITEM rondas_para_rotacion 20
Agent/CLUDITEM nro_celdas [expr $cellPerX_ * $cellPerY_]
Agent/CLUDITEM open-traceplacement.txt
Agent/CLUDITEM open-nam-traceplacement.nam $width_ $height_
Agent/CLUDITEM tx_power_EC .00000316
Agent/CLUDITEM tx_power_ACH .001
Agent/CLUDITEM size_DATA 200
#=====

```

Figura 21. Parámetros del agente CLUDITEM

El funcionamiento de la estación base, también está basado en el disparo de temporizadores, cuyos valores de conteo están ligados a los definidos para los nodos de la red. Los mensajes agregados que recibe, se reportan en el archivo de *trace* que entrega el simulador, para analizar el monto de información que la red logra hacer llegar hasta el nodo *sink*. La potencia de transmisión de los mensajes de la estación base corresponde al radio de transmisión intercluster como se muestra en el *script* de la figura 22.

```

#=====
# Configuracion del SINK
#=====
Agent/CLUDITEM_Sink rondas_para_rotacion 20
Agent/CLUDITEM_Sink tx_power_ACH.001
#=====

```

Figura 22. Configuración de la estación base

5.2.1 Definición de parámetros del modelo de energía

NS2 incluye tres modelos de propagación de radio, *free-space*, *two-ray ground reflection* y *shadowing* que permiten predecir el valor de la potencia de recepción de los paquetes intercambiados por los nodos de la red, teniendo en cuenta los parámetros del sistema. El simulador trabaja definiendo un umbral de recepción a nivel de la capa física. Cuando ingresa un mensaje, si la potencia con la que se lo recibe está por debajo de dicho umbral se lo descarta, en caso contrario se lo envía a la capa MAC.

En la implementación de CLUDITEM se seleccionó el modelo de propagación de dos rayos o *two-ray ground reflection*, que considera no sólo la transmisión directa entre el emisor y el receptor, sino también una componente originada por reflexión en la superficie terrestre. Este modelo proporciona una predicción del valor de la potencia de recepción de los paquetes intercambiados, mediante la expresión (17).

$$P_r(d) = (P_t \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2) / (d^4 \cdot L) \quad (17)$$

En la misma, G_t y G_r son las ganancias de las antenas de transmisión y recepción, h_t y h_r sus correspondientes alturas, y L representa la pérdida del sistema. La potencia de la señal emitida P_t , se estableció a nivel de mensaje, para tener en cuenta los dos radios de transmisión con los que trabaja CLUDITEM. El modelo de dos rayos resulta más adecuado para el escenario elegido que el de propagación en espacio libre o *free-space*, ya que los radios de transmisión adoptados superan la distancia de cruce o *cross-over* [71], definida por la expresión (18).

$$d_{cross} = (4 \cdot \pi \cdot h_t \cdot h_r) / \lambda \quad (18)$$

Los valores de los parámetros del modelo, que se detallan en la tabla 1, se definieron utilizando las especificaciones de la hoja de datos del transceptor CC2420 de Texas Instruments [68], que permite establecer distintos valores de potencia de transmisión. Esta característica permitió trabajar con los dos alcances previstos en CLUDITEM, el primero para la comunicación de los NC con su cabecera y el segundo para el envío de datos agregados por parte de los CH.

Definición	Parámetro	Valor
Ganancia antena de transmisión	G_t	0 dB
Ganancia antena de recepción	G_r	0 dB
Altura antena de transmisión	h_t	0,33 m
Altura antena de recepción	h_r	0,33 m
Pérdida del sistema	L	0 dB
Longitud de onda	λ	$1,25 e^{-01}$ m
Frecuencia	freq	$2,40 e^{+09}$ Hz
Potencia de transmisión intracluster	tx_power_EC	0,00000316 W
Potencia de transmisión intercluster	tx_power_ACH	0,001 W
Umbral de recepción (sensibilidad)	RXThresh	$3,16 e^{-13}$ W
Consumo intracluster en transmisión	tx_consume_EC	0,01516 W
Consumo intercluster en transmisión	tx_consume_ACH	0,03067 W
Consumo en recepción	Pr_consume	0,03528 W
Consumo en escucha	P_idle	0,03528 W
Consumo en apagado	P_sleep	0,000000144 W

Tabla 1. Valores de los parámetros del modelo de radio y consumo en los distintos estados del transceptor

Los valores definidos para el modelo de propagación, establecen radios de transmisión intra e intercluster de 18,56 m y 78,27 m respectivamente, para el escenario de simulación adoptado que se describe en el apartado 5.4. Estos alcances compatibilizan el análisis realizado en las secciones 4.7.1 y 4.7.2 referidos a potencias de transmisión, con el modelo de propagación de radio adoptado y los valores de la hoja de datos del transceptor utilizado.

5.2.2 Temporización del funcionamiento

CLUDITEM es un algoritmo distribuido y por lo tanto cada nodo, que dispone de un reloj propio, decide en base a la información y la temporización local las tareas que realizará. El funcionamiento del agente de encaminamiento CLUDITEM de cada dispositivo, se basa en el disparo de un conjunto de temporizadores asociados con las tareas a realizar,

dependiendo de la fase de trabajo y el rol del nodo en cada ronda de recolección de información. Los temporizadores incorporados al agente CLUDITEM son los siguientes:

- **TIMER_DESPIERTOS:** Tiene en cuenta el desfase estimado para los relojes cuando se inicia una ronda de recolección.
- **TIMER_POSTULACION:** Indica al nodo postulante a cabecera de *cluster* el momento en el que debe enviar un mensaje avisando que se propone como CH. El desfase entre postulaciones está definido por un esquema TDMA, basado en la cantidad de postulantes por celda permitidos. Los potenciales postulantes enviarán el mensaje de postulación si se cumplen las condiciones establecidas.
- **TIMER_RUTA_AL_SINK:** Define el momento de envío de mensajes ACH por parte de las cabeceras, para establecer el árbol de CH. Esta fase se inicia con un mensaje de la estación base.
- **TIMER_RUTA_ARMADA:** Indica que la estructura de encaminamiento está completa
- **TIMER_ENVIO_DATOS:** Establece el momento de envío de datos para cada nodo común, de acuerdo al esquema TDMA adoptado.
- **TIMER_DORMIR:** Avisa a los nodos comunes que deben pasar a modo de bajo consumo hasta la próxima ronda.
- **TIMER_ENVIO_AGR:** Define el momento de envío de mensajes agregados para cada CH de acuerdo al esquema TDMA adoptado.
- **TIMER_FIN_AGR:** Indica la finalización de envío de información hacia la estación base, y por lo tanto el momento en el cuál los CH quedan inactivos hasta el cumplimiento de T.
- **TIMER_INICIO:** Define la espera hasta próxima ronda de recolección, ajustando su valor para rondas con reconfiguración y rondas de envío exclusivo de datos.

La temporización de la red se completa con el agente CLUDITEMSINK, que corresponde a la estación base, e incorpora los siguientes temporizadores:

- **TIMER_SINK_ROUND**: Establece la duración de la ronda dependiendo si corresponde definir estructura de encaminamiento, o sólo recolectar datos.
- **TIMER_SINK_START_ACH**: Define el inicio del armado del árbol de cabeceras con el envío de un mensaje ACH por parte de la estación base.

Los tiempos asociados con los temporizadores de los nodos de la red y de la estación base se definieron tomando en cuenta las consideraciones del capítulo 4. La definición de los esquemas TDMA se realizó en base a los resultados de simulación de la sección 5.6.

5.3 Escenario de simulación

El escenario de simulación se definió como un área cuadrada de 135 metros de lado dividida en 9 celdas, numeradas de 0 a 8 (ID de celda), con 16 nodos cada una, separados 12 metros entre sí, valores que resultan compatibles con las aplicaciones seleccionadas.

```

#=====
# Placement nodes
#=====
set nodeNum_      144 ; # number of mobilenode
set nodesPerX_   12  ; # number of mobilenode in X
set nodesPerY_   12  ; # number of mobilenode in Y
set nodeDistance_ 12  ; # distance between nodes
set width_       135 ; # simulation width area
set height_      135 ; # simulation height area
set cellPerX_    3   ; # number of cells in X
set cellPerY_    3   ; # number of cells in Y
set cellnodes_   [expr $nodeNum_/($cellPerX_*$cellPerY_)]
set namSize_     0.5 ; # size nam
#=====
# Placement SINK
#=====
set sinkX_       67.0 ; # sink location
set sinkY_       -10.0 ; # sink location
set sinkNamSize_ 0.8  ; # size nam
#=====

```

Figura 23. Definición del escenario de simulación

La estación base se ubicó fuera del entorno supervisado garantizando la definición del árbol de cabeceras de acuerdo con los cálculos basados en el análisis realizado en la sección 4.7.2. Las características del escenario se asociaron con parámetros para permitir ajustes del funcionamiento del algoritmo, como se indica en el *script* de la figura 23.

5.4 Evaluación de CLUDITEM

Para ajustar el funcionamiento de CLUDITEM, se definieron en su implementación dos conjuntos de parámetros perfectamente diferenciados, uno trata sobre el escenario de distribución de los nodos y el otro se relaciona con las fases de funcionamiento del algoritmo. El primer grupo se refiere a las dimensiones del área a supervisar, cantidad de celdas y tamaño de la grilla virtual, número de nodos por celda y distancia entre ellos. El segundo grupo incluye la cantidad de postulantes a CH por celda, profundidad máxima de los *clusters*, valores de la potencia de transmisión y de los tiempos que disparan los temporizadores que definen las distintas fases del algoritmo.

Los parámetros relacionados con el escenario en el que se despliega la red, están orientados a definir distintas situaciones acordes con cada aplicación. Por su parte, los parámetros ligados a las fases de trabajo, se definieron con la idea de poder ajustar el funcionamiento del algoritmo considerando las especificaciones y las métricas de evaluación definidas.

En el capítulo 3 se establecieron lineamientos de diseño para CLUDITEM focalizados en dos aspectos. El primero se refiere al logro de una buena gestión de la estructura de *clusters* y está orientado a balancear la carga de trabajo y definir zonas de recolección de información ligadas a los CH. El segundo apunta a realizar un manejo adecuado del apagado de los transceptores, combinado con la rotación de los roles de los nodos durante el funcionamiento de la red, con el objetivo de balancear el consumo. Estas dos importantes características del algoritmo, colaboran además en el cumplimiento de la QoS establecida, respecto de no superar un porcentaje de pérdida de mensajes. Para evitar la degradación de la QoS requerida debido a posibles interferencias intra e intercluster se incorporaron al algoritmo, tal como se detalla en el capítulo 4, esquemas TDMA para el envío de mensajes. En consecuencia, durante el proceso de evaluación se definieron tres grupos de métricas, destinados a cuantificar los resultados obtenidos en los aspectos antes mencionados.

Además, la incorporación de una grilla virtual es una característica de CLUDITEM íntimamente relacionada con la gestión de la estructura de encaminamiento. Por este motivo, se complementa el proceso de evaluación, con la comparación del desempeño de CLUDITEM respecto a la de otro algoritmo similar que no utiliza la grilla.

5.4.1 Métricas para evaluar la gestión de la estructura de clusters.

Para evaluar el desempeño del algoritmo desarrollado respecto de la gestión de la estructura de encaminamiento que establece, se definieron las siguientes métricas:

- Cantidad de CH
- Cantidad de nodos desconectados
- Cantidad de hijos con CH en la misma celda
- Cantidad de hijos por CH.

Los nodos CLUDITEM no son conscientes de su ubicación física, pero conocen la celda de la grilla a la que pertenecen. El algoritmo implementa estrategias para que en cada reconfiguración de la red, los nodos comunes elijan el *cluster* al que adhieren favoreciendo el establecimiento de conjuntos de nodos balanceados en profundidad y cantidad de miembros. En este sentido, divide el área a supervisar en una grilla virtual, que apoya la tarea de gestionar la estructura de los *clusters*. La situación ideal, corresponde al logro de *clusters* con igual número de hijos, cuyos CH estén distribuidos en el área de trabajo, uno por celda de la grilla. Además, es deseable que mayoritariamente los miembros de cada *cluster* pertenezcan a la misma división de la grilla que su cabecera, para agrupar nodos cercanos físicamente. Esta situación es favorable ya que las variables bajo estudio tienen un alto grado de correlación, que puede aprovecharse al hacer agregación, y apunta a la idea de un control por zona de la información recolectada. Se agrega la métrica de cantidad de nodos desconectados que impacta directamente sobre la QoS de la aplicación.

5.4.2 Comparación de CLUDITEM con otro algoritmo de encaminamiento

El desempeño de CLUDITEM, se comparó con el de otro algoritmo jerárquico, que elige los postulantes a CH en forma aleatoria entre todos los nodos de la red. Este nuevo algoritmo, denominado SIN GRILLA, se implementó en NS2 utilizando el mismo

escenario, el mismo esquema de tiempos y las mismas fases de funcionamiento previstas para CLUDITEM. La principal diferencia radica en que para cada reconfiguración de la estructura de encaminamiento, SIN GRILLA permite la postulación de exactamente 9 CH en toda el área supervisada, seleccionados aleatoriamente entre todos los miembros de la red. Cabe destacar que el nuevo algoritmo, tendrá siempre la cantidad ideal de cabeceras previstas para CLUDITEM, y que también se rotarán cada X rondas de medición, de forma de garantizar que cada dispositivo asuma una única vez el rol de cabecera durante el tiempo de vida del sistema.

Los parámetros del modelo de energía para el algoritmo SIN GRILLA se definieron en base al transceptor CC2420, utilizado también para CLUDITEM, quedando establecidos los mismos valores para ambos algoritmos, salvo en el caso de la potencia de transmisión intracluster. Esto fue necesario, ya que la ubicación de los CH de SIN GRILLA es aleatoria y se debe garantizar pese a ello la definición del árbol de cabeceras. Por lo tanto, se adoptó tx_power_ACH como 0,02 W, para un radio de transmisión intercluster de 165,5 metros.

Las métricas que se utilizaron para la comparación de ambos algoritmos fueron:

- Cantidad promedio de nodos desconectados
- Ronda de salida de servicio del sistema
- Cantidad de mensajes perdidos en dicha ronda

El principal objetivo de la comparación de los algoritmos, fue cuantificar el beneficio de la incorporación de la grilla virtual en CLUDITEM. La elección de cabeceras entre los nodos miembros de cada división favorece una distribución por zona de los CH en toda la superficie bajo estudio, con el fin de balancear la carga de trabajo y disminuir la presencia de dispositivos sin enlace en la estructura. En cambio, al elegir las cabeceras en forma aleatoria en toda el área a supervisar, su ubicación resulta azarosa, y por lo tanto es probable que los *clusters* sean más desparejos en cuando al número de componentes y que se incremente el número de nodos desconectados, con la correspondiente pérdida de información. Por estos motivos se consideró importante comparar además la salida de servicio del sistema en función del número de mediciones que no llegan a la estación base, cuando la red se configura con uno u otro algoritmo de encaminamiento.

5.4.3 Métricas relacionadas con la QoS

La QoS requerida está basada en el porcentaje de mensajes que la red es capaz de hacer llegar hasta el *sink*. Las métricas definidas en el punto 5.4.1, se enfocan a establecer los mejores valores de los parámetros del algoritmo para lograr una buena gestión de la definición de los *clusters*. Sin embargo, el logro de *clusters* conformados de acuerdo con el análisis realizado no garantiza por sí solo la QoS. Los algoritmos basados en *clusters* sufren de importantes pérdidas de información, debido al problema de la terminal oculta, cuando los dispositivos transmiten las mediciones realizadas. El agregado de esquemas TDMA, puede ser muy conveniente, pero si el número de *slots* de transmisión es grande, se incrementa el tiempo que los nodos deben estar activos y por lo tanto, también aumenta el consumo de energía. Por estos motivos se decidió evaluar distintos esquemas TDMA para las fases de envío de datos, definiendo las siguientes métricas para compararlos:

- Cantidad de mensajes recibidos por las cabeceras de *cluster*
- Cantidad de mensajes recibidos por la estación base
- Cantidad de *slots* de transmisión necesarios

La primera métrica está orientada a evaluar el esquema utilizado para el envío de información desde los NC a su cabecera, mientras que la segunda se refiere al esquema aplicado para que los CH transmitan hacia la estación base. La última, se aplica a ambos casos para tener en cuenta el impacto sobre el tiempo que los dispositivos deben permanecer activos. No se evalúa el retardo introducido para el arribo de la información a la estación base, ya que el mismo es irrelevante para las aplicaciones elegidas.

5.4.4 Métricas referidas al consumo de energía

Los valores de consumo del transceptor CC2420 que se utilizó para definir los parámetros del modelo de energía en simulación, se resumen en la tabla 1 y ponen en evidencia que el consumo más relevante corresponde al estado de escucha. Esto se debe a que la potencia consumida por el transceptor en ese estado es similar, e incluso superior, a las correspondientes a recepción y transmisión, siendo el estado de escucha el que asumen los nodos la mayor parte del tiempo que están activos. En este sentido, el período que los dispositivos permanecen despiertos, es fundamental en la definición de la energía que

consumen. Por lo tanto, dado que las rondas con reconfiguración son más prolongadas que las de envío exclusivo de datos, y que los CH permanecen activos más tiempo que los NC en todas las rondas, se consideraron los siguientes valores para analizar el consumo de la red:

- Consumo de los nodos por rol y tipo de ronda
- Energía remanente de los dispositivos al finalizar el tiempo de vida de la red
- Salida de servicio de la red por incumplimiento de la QoS al variar X

Los nodos que asumen el rol de CH son los más exigidos respecto del gasto de energía, pero su situación debería equilibrarse respecto a los NC a partir del mecanismo de rotación de roles. Por lo tanto, se decidió analizar la energía remanente en el conjunto de la red al finalizar una rotación completa de CH, con el fin de evaluar si la técnica incorporada al algoritmo resulta efectiva para balancear la situación de los dispositivos.

Otro aspecto considerado fue el cálculo del valor óptimo del momento de reconfiguración de la red, realizado en la sección 4.7.3 considerando que cada nodo consume el total de su energía al cumplirse una rotación completa de roles. Para verificar por simulación el resultado analítico, se utilizó la expresión (16) para definir el parámetro X, `rondas_para_rotacion` en NS2, a partir del valor de energía inicial adoptado y los consumos por rol en cada fase de trabajo. Se simuló el funcionamiento del algoritmo con los parámetros utilizados para el cálculo teórico, variando únicamente el valor de X y se analizaron los resultados obtenidos.

5.5 *Análisis de la definición de la estructura de clusters*

Se realizaron 1920 configuraciones del árbol de encaminamiento, con el objetivo de analizar la estructura de los *clusters* definidos en el escenario adoptado. La figura 24 ejemplifica una de las configuraciones logradas según la presenta la herramienta gráfica asociada con el simulador. En dicha figura, pueden apreciarse los 144 nodos de la red, identificados por la celda a la que pertenecen (ID de celda) y por un ID de nodo. Los CH, uno por celda de la grilla en este caso, están remarcados y se puede apreciar la estructura generada mediante las líneas que indican el NDE de cada dispositivo en su *cluster*. La estación base, queda fuera del área graficada.

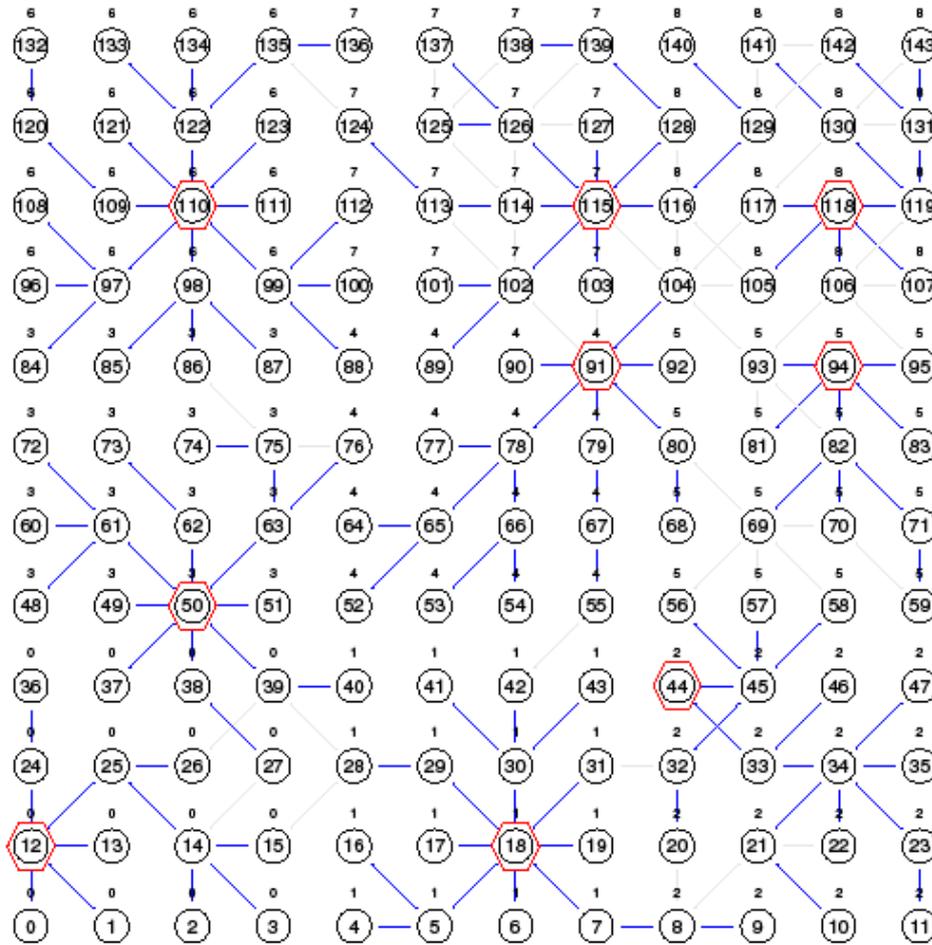


Figura 24. Ejemplo de definición de la estructura de *clusters*

Las métricas definidas en la sección 5.2.1 para evaluar la gestión de la definición de los *clusters*, se utilizaron para analizar los resultados de simulación. En este sentido, se consideró el impacto de la máxima profundidad permitida en cada *cluster*, y del número de postulantes a cabecera admitido. Los resultados obtenidos para las métricas mencionadas al variar la máxima profundidad en los *clusters* se muestran en figura 25.

Analizando la profundidad de los *clusters*, se observa que el valor de 3 resulta inaceptable pues el número de nodos desconectado es muy superior al obtenido para las otras profundidades. El valor 4, es el que brinda mejor resultado para el promedio de nodos con CH en la misma celda, reforzando la idea de medición por zona, y el que más aproxima al número deseado de 15 hijos para balancear la carga de trabajo entre *clusters*. Por estos

motivos, se adoptó este valor como la cantidad máxima de saltos intracluster para la implementación del algoritmo.

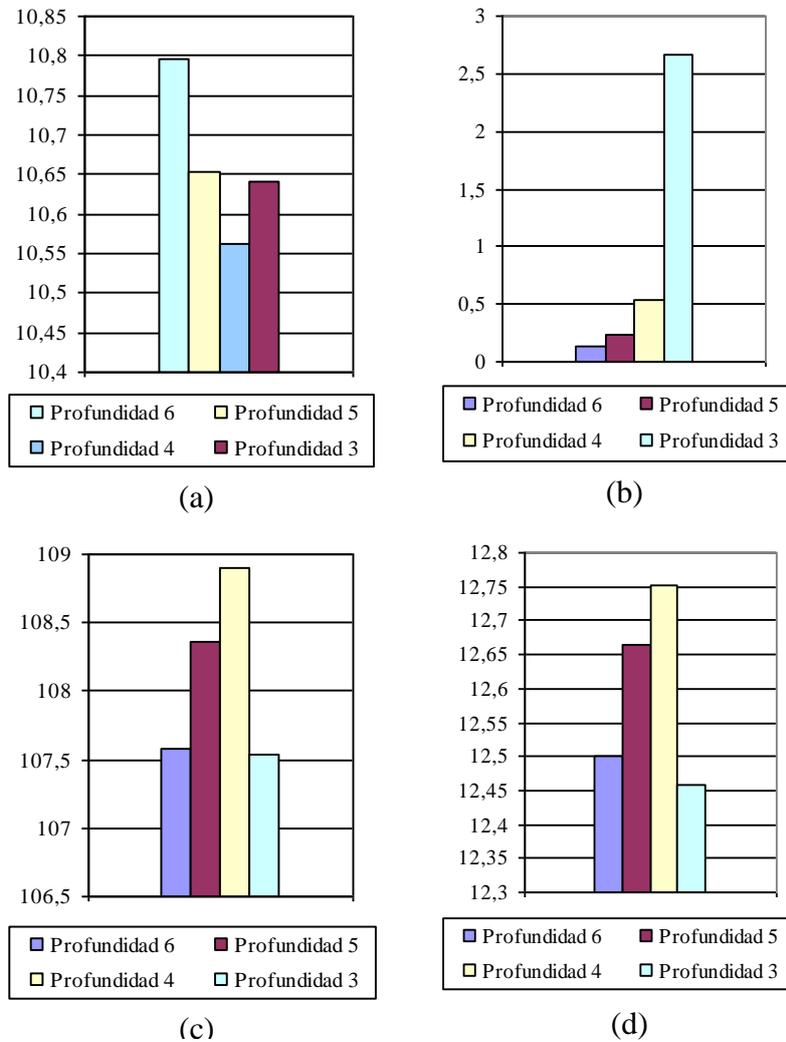


Figura 25. Análisis de la máxima profundidad de los *clusters*.

(a) Promedio de CH, (b) Promedio de nodos desconectados, (c) Promedio de nodos con cabecera en la misma celda, (d) Promedio de hijos por cabecera.

A continuación, para igual número de reconfiguraciones del árbol de encaminamiento, se estudió la influencia del número de postulantes por celda de la grilla variando la cantidad entre 1 y 3. Los resultados obtenidos se resumen en la figura 26.

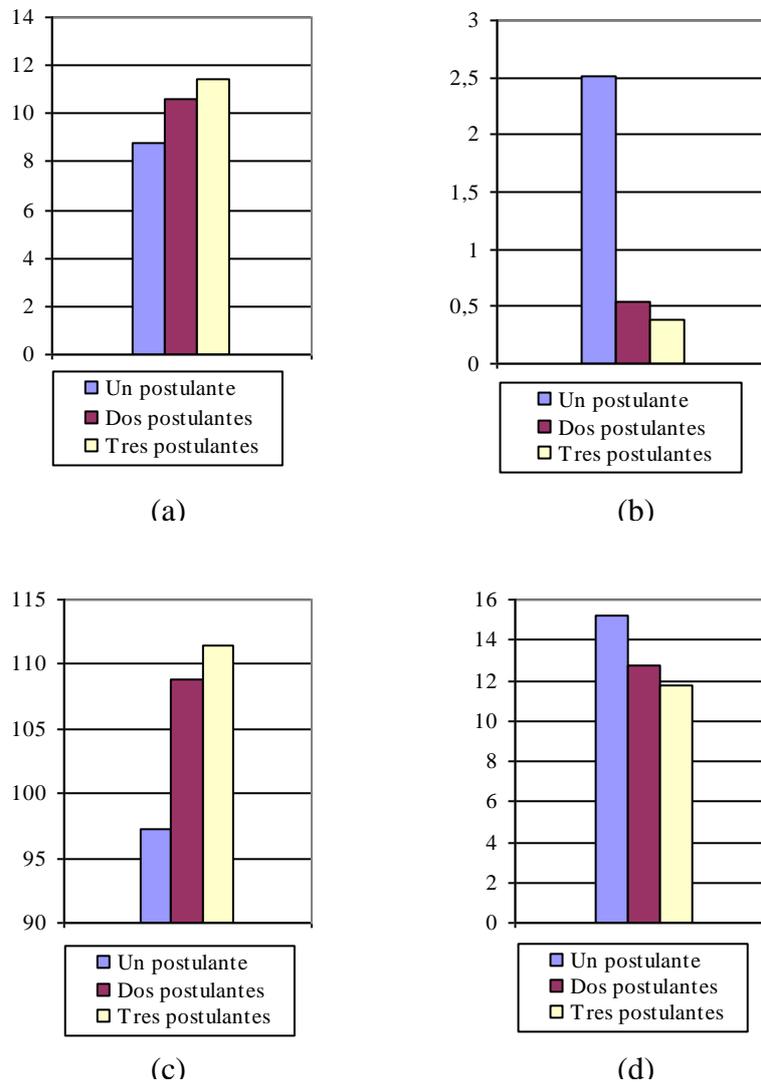


Figura 26. Análisis del número de postulantes por celda.

(a) Promedio de CH, (b) Promedio de nodos desconectados, (c) Promedio de nodos con cabecera en la misma celda, (d) Promedio de hijos por cabecera.

La opción de un único postulante se descartó ya que reducía notablemente el número de nodos con cabecera en la misma celda, y elevaba la cantidad de dispositivos desconectados en la red. Los valores de las métricas para dos y tres postulantes resultaron en cambio bastante aproximados. Por lo tanto, para elegir el valor del parámetro, se le dio peso al hecho de que al permitir dos postulantes por celda, disminuía el promedio de cabeceras resultante en la red. Si bien esta característica no apunta directamente al requerimiento de

controlar las mediciones por zona, es importante respecto del ahorro de energía, ya que los CH son nodos más exigidos en cuanto al consumo.

Es importante destacar que los parámetros definidos aportan un interesante grado de flexibilidad al algoritmo respecto a variaciones de las especificaciones, siempre dentro del grupo de aplicaciones elegidas. Por ejemplo, si no es importante agrupar los nodos físicamente próximos en cada zona del área estudiada, y en cambio resulta fundamental reducir la pérdida de mensajes, podría considerarse utilizar una profundidad máxima de 5 para los *clusters*, que reduce aún más el número de nodos desconectados. Esto quiere decir que al ponderar la importancia de cada métrica de acuerdo con las variaciones de las especificaciones en el grupo de aplicaciones elegido, puede personalizarse el funcionamiento del algoritmo, para obtener mejores resultados.

5.6 Comparación de CLUDITEM con el algoritmo SIN GRILLA

Se analizó el funcionamiento de los dos algoritmos utilizando el escenario descrito en la sección 5.3, mediante 10 simulaciones con una duración superior a la definición del tiempo de vida de la red. Por lo tanto, cada simulación involucró al menos 16 reconfiguraciones con $X = 18$, o sea, en ambos casos se redefinió la estructura de encaminamiento luego de 18 rondas de envío de datos. La energía inicial de cada dispositivo garantizaba que ninguno de ellos la agotara al cabo de las 288 rondas correspondientes a una rotación completa de roles, con el fin de centrar el análisis en la estructura de encaminamiento lograda por cada algoritmo, y que la pérdida de información en las 288 rondas correspondientes a la definición del tiempo de vida del sistema, no fuera atribuible a la desconexión por falta de recursos. Asimismo, se minimizó la posibilidad de interferencias utilizando los esquemas TDMA con mayor cantidad de *slots* de transmisión en las dos fases de envío de datos, para reducir a un mínimo la pérdida de mensajes por el problema de la terminal oculta. En estas condiciones se obtuvo el promedio de nodos desconectados en cada período de adquisición de datos, la ronda de medición en la cual el sistema queda fuera de servicio por no cumplir la QoS y la cantidad de mensajes perdidos en ese momento.

Las pruebas pusieron en evidencia un incremento significativo del promedio por ronda de nodos desconectados en el algoritmo SIN GRILLA respecto a CLUDITEM. Este último

logró un valor de 0,69 frente a 7,13 de SIN GRILLA. Las simulaciones confirmaron el hecho de que la ubicación azarosa de los CH en toda el área supervisada, produce configuraciones de encaminamiento muy desfavorables para cumplir con el umbral de recepción de mensajes en la estación base. En la figura 27 se muestra uno de estos casos en el cual queda una porción importante de la superficie bajo estudio sin cabeceras. Esto ocasiona que en esa zona exista un elevado número de nodos desconectados, cuyas mediciones no llegan nunca al *sink*.

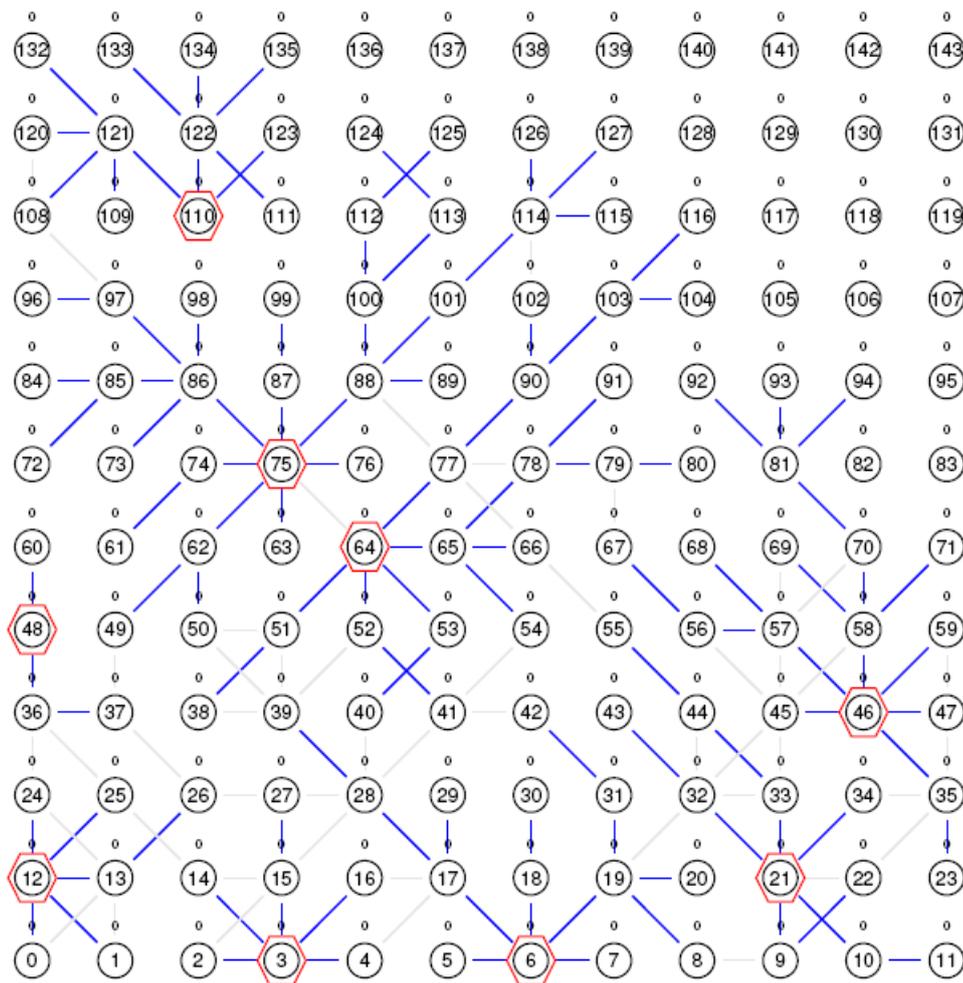


Figura 27. Ejemplo de estructura de clusters del algoritmo SIN GRILLA

Se debe destacar que las configuraciones desfavorables de la estructura de encaminamiento se utilizan durante X rondas de envío de información. En las pruebas se adoptó X igual a 18, pero en campo, considerando la energía inicial provista por dos baterías de tipo AA las

rondas para reconfiguración calculadas según la expresión (16) ascienden aproximadamente, dependiendo de la temporización del envío de datos, a 5000 rondas. Esto quiere decir que el impacto sobre la QoS de estructuras de red con un número importante de dispositivos desconectados es muy alto. Para verificarlo por simulación se estableció como requerimiento de QoS la ocurrencia de 5 rondas de medición con más del 5% de pérdida de mensajes en la estación base. En la tabla 2 que resume los resultados de las primeras 5 corridas de ambos algoritmos, se observa que debido a que la definición de estructuras altamente desfavorables en SIN GRILLA es aleatoria, el límite impuesto por la QoS puede superarse incluso en la primera configuración de la red. Por lo tanto, puede concluirse que el desempeño de SIN GRILLA es inaceptable para las aplicaciones seleccionadas, y que en cambio el apoyo de la grilla virtual para la gestión de la estructura de encaminamiento pone a CLUDITEM en una situación muy favorable para abordar dichas aplicaciones. En este sentido, la tabla 2 muestra que en todos los casos CLUDITEM superó las 288 rondas de mediciones previstas cumpliendo los requerimientos de QoS establecidos, saliendo de servicio aproximadamente en la misma ronda en todas las simulaciones. Esta situación pone en evidencia que una vez cumplido el tiempo de servicio teórico de 288 períodos de recolección de datos comienza a pesar el problema de la energía remanente, que se analiza en detalle en la sección 5.8. Por su parte, el algoritmo SIN GRILLA que puede finalizar su vida útil muy tempranamente, presenta un número variable de mensajes perdidos, atribuible específicamente a desconexiones en la red.

SIN GRILLA			CLUDITEM		
Simulación	Ronda de fin de servicio	Nodos no reportados en esa ronda	Simulación	Ronda de fin de servicio	Nodos no reportados en esa ronda
1	77	8	1	293	22
2	95	12	2	293	26
3	148	10	3	292	22
4	95	7	4	293	24
5	5	13	5	293	17

Tabla 2. Comparación del funcionamiento de CLUDITEM vs. SIN GRILLA

Por último es importante destacar que, al presentar el algoritmo SIN GRILLA estructuras de encaminamiento como la mostrada en la figura 27, con las cabeceras concentradas en una porción de la superficie donde se despliega la red, no resulta conveniente si se considera analizar la información por zona del área bajo estudio.

5.7 Evaluación de esquemas TDMA

Se realizaron simulaciones del funcionamiento del algoritmo con el objetivo de elegir un esquema TDMA para las fases de envío de datos capaz de cumplir con los requerimientos de QoS de las aplicaciones. Primero se trabajó para definir la temporización del envío de datos intracelular, y con esos valores se ajustó luego, la fase de transmisión de datos agregados.

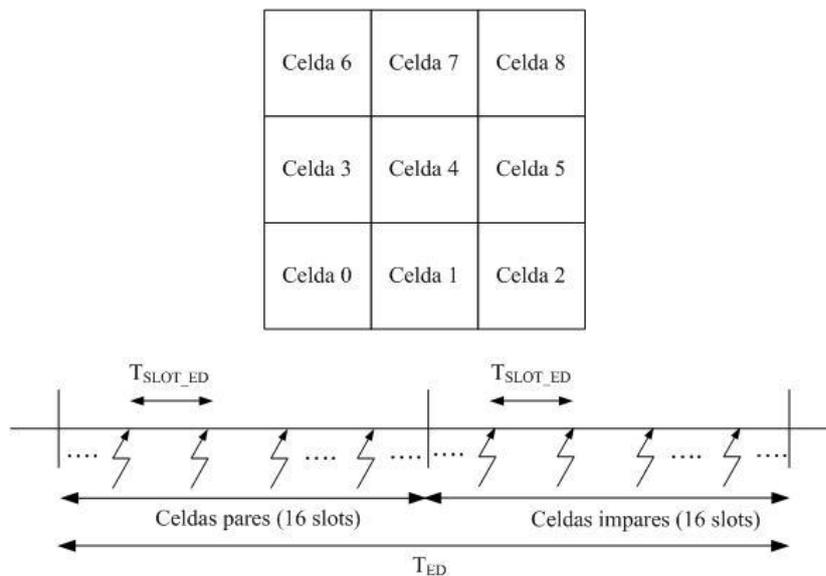


Figura 28. Esquema intracelular de 32 slots para transmisión de datos

Se consideraron tres posibilidades para el esquema TDMA de envío de información a cargo de los NC. La primera contempla la definición de slots de transmisión asociados con el ID de cada nodo en su cuadrícula. Las celdas de la grilla virtual contienen n nodos, con n igual a 16 en el escenario utilizado para simulación, por lo que se definieron 16 slots de transmisión para este caso. Cabe destacar que, salvo el desfasaje de relojes (T_{DIC}), habrá 9 nodos, uno por celda de la grilla, transmitiendo datos hacia su CH en el mismo slot de

tiempo. El segundo esquema considerado, amplía el número de *slots* a 32 dividiendo las celdas en dos grupos. Primero transmiten los nodos de las celdas pares y luego los de las celdas impares, ya que de esta manera, los nodos pertenecientes a celdas adyacentes que son las más críticas respecto de la posibilidad de colisiones, transmiten en *slots* diferentes como muestra la figura 28.

Finalmente se incrementó aún más el número de *slots*, definiendo uno para cada nodo de la red, resultando para el escenario elegido 144 *slots* de transmisión de datos. Para cada caso se realizaron 10 simulaciones de funcionamiento del algoritmo, con una rotación completa del rol de CH en todas las celdas. Estas simulaciones, incluyeron 16 rondas de reconfiguración, seguida cada una de ellas de 17 rondas de transmisión exclusiva de datos ($X = 18$), por lo que se analizaron en total para cada esquema 2880 rondas de medición.

La energía inicial de los nodos se calculó de forma de que ninguno de ellos la agotara durante el funcionamiento. De esta manera, la pérdida de mensajes intracluster sólo puede atribuirse a dos motivos, que existan nodos desconectados o que se produzcan colisiones por interferencias. Como ya se reportó en la comparación de CLUDITEM con SIN GRILLA, el promedio de nodos desconectados para el primero es muy bajo (0,69 nodos por ronda), por lo que puede afirmarse que prácticamente la totalidad de los mensajes perdidos se debe al problema de la terminal oculta.

La tabla 3 que resume los resultados para cada modalidad de funcionamiento, pone en evidencia que los esquemas de 32 y 144 *slots* de transmisión presentan una mejora importante, considerando los requerimientos de QoS, respecto del esquema de 16 *slots*. Por otro lado, los dos esquemas con mayor número de *slots* no difieren demasiado entre ellos, al comparar el número de mensajes que colectan las cabeceras. En base a estos resultados, se decidió adoptar el esquema de 32 *slots* por dos motivos, primero porque disminuye significativamente el tiempo que los nodos deben estar prendidos contribuyendo a la disminución del consumo, y segundo porque resulta mucho más escalable si se debe supervisar un área más extensa. La superficie en estudio, se divide en base a la grilla virtual, y una vez establecidas las dimensiones de las celdas conforme al tamaño de las zonas a analizar y el número de nodos en cada una, ampliar el área a supervisar significa agregar celdas a la cuadrícula. Por lo tanto, con el esquema elegido se mantiene el número

de *slots* de transmisión, a diferencia del tercer esquema considerado que necesita un *slot* exclusivo para cada nodo de la red.

	Esquema 16 slots	Esquema 32 slots	Esquema 144 slots
Simulación	Promedio de mensajes recibidos por los CH	Promedio de mensajes recibidos por los CH	Promedio de mensajes recibidos por los CH
1	139,833	143,573	143,688
2	140,410	143,201	143,938
3	141,281	143,153	143,813
4	140,792	143,222	143,750
5	140,323	143,521	143,875
6	141,559	143,337	143,684
7	141,226	143,184	143,875
8	140,962	143,059	143,438
9	140,240	143,608	143,875
10	140,955	143,611	143,743
<i>Promedio General</i>	<i>140,758</i>	<i>143,347</i>	<i>143,768</i>

Tabla 3. Resumen resultados de simulación para los esquemas TDMA intracluster

El esquema de 16 *slots*, que se descartó en base a los requerimientos de QoS establecidos, podría sin embargo resultar útil para aplicaciones muy exigidas respecto del consumo de energía, en las que fuera posible flexibilizar la especificación referida al porcentaje admisible de pérdida de mensajes.

El envío de información agregada se analizó en base a dos posibilidades para el esquema TDMA de la fase. La primera define los *slots* de transmisión en base al ID de cada celda, o sea 9 *slots* en total, pudiendo ocurrir si se postularon dos CH en una celda, que ambos envíen su agregado hacia la estación base al mismo tiempo, salvo T_{DIC} . La segunda incluye en la definición del *slot* de envío de agregados un término relacionado con la cantidad de CH por celda, lo que eleva el número de *slots* a 18 teniendo en cuenta que se permiten a lo sumo dos cabeceras por división de la cuadrícula.

Las pruebas realizadas pusieron en evidencia la importancia de tomar recaudos para la fase de envío de agregados. La pérdida de cada agregado significa resignar la información de una zona completa, y efectivamente el esquema de un único *slot* por celda, incurrió en una disminución muy importante de la información que la red fue capaz de hacer llegar hasta la estación base. Para demostrarlo se presentan en la tabla 4 el número de ronda de medición en la cual el sistema sale de servicio por no cumplir con la QoS definida, respecto de perder en 5 rondas de recolección de información más del 5% de las mediciones.

Simulación	Ronda de salida de servicio	Simulación	Ronda de salida de servicio	Simulación	Ronda de salida de servicio
1	41	11	18	21	8
2	82	12	36	22	25
3	19	13	174	23	24
4	169	14	22	24	21
5	6	15	73	25	137
6	22	16	13	26	10
7	37	17	36	27	35
8	27	18	6	28	38
9	12	19	20	29	5
10	6	20	39	30	21

Tabla 4. Salida de servicio de la red con esquema de 9 *slots* de envío de agregados

Las simulaciones de la tabla 4 se diseñaron para que se realicen 288 rondas de medición correspondientes a una rotación completa de roles en cada celda, pero en ninguna de las 30 simulaciones realizadas la red fue capaz de cumplir con el número de rondas previstas, produciéndose la salida de servicio por no respetarse el requerimiento de QoS. Por este motivo, el esquema de 9 *slots* de envío de agregados resultó inadmisibles para las aplicaciones de interés. Por su parte, el esquema de 18 *slots* de transmisión intercluster arrojó muy buenos resultados, como se resume en la tabla 5, que reúne 10 simulaciones completas del funcionamiento de la red utilizando 32 *slots* de transmisión intracluster y 18 para envío de agregados. Sin embargo durante las simulaciones realizadas se presentaron

algunos casos en los cuales se superaba el límite de 7 mensajes perdidos por ronda debido a colisiones en la red, antes de cumplir las 288 rondas de vida útil. Esto no es sorprendente ya que los valores que se reportan son promedios y es posible que en algún caso el número de colisiones resulte superior a otro. Durante las pruebas la aparición de estas situaciones fue reducida, pero si se debe abordar una aplicación extremadamente exigente que no admita superar el límite establecido en ningún caso, se debería pensar en incluir en el algoritmo estrategias tendientes a detectar y corregir estas situaciones. La evaluación realizada en base a las aplicaciones de interés llevó a dejar de lado la aparición esporádica de las situaciones conflictivas mencionadas, y se adoptó la configuración de 32 *slots* de transmisión intracluster y 18 para mensajes agregados en la temporización de las fases de envío de datos de CLUDITEM.

Simulación	Promedio de mensajes recibidos por los CH	Promedio de mensajes recibidos por el sink
1	143,573	143,573
2	143,201	142,201
3	143,153	143,024
4	143,222	143,222
5	143,521	142,458
6	143,337	143,337
7	143,184	142,184
8	143,059	143,059
9	143,608	143,608
10	143,611	143,556
<i>Promedio General</i>	<i>143,347</i>	<i>143,022</i>

Tabla 5. Promedio de mensajes por ronda con 32 *slots* de envío de datos y 18 *slots* de envío de agregados.

5.8 Resultados referidos al modelo de energía: Tiempo de vida de la red

Los resultados de simulación obtenidos en las pruebas anteriores permitieron elegir los parámetros de configuración del algoritmo y los esquemas TDMA más adecuados para las aplicaciones seleccionadas. A continuación, utilizando estos valores se procedió a ajustar la

temporización del funcionamiento para obtener el consumo de los dispositivos y poder evaluar sobre esta base el desempeño de CLUDITEM respecto de la distribución de la carga de trabajo entre los nodos y el cumplimiento del tiempo de vida útil del sistema.

La tabla 6 muestra los valores adoptados para la simulación. Los *slots* de transmisión intra e intercluster se ajustaron en base a los archivos de trazado de resultados previos y se utilizaron para definir la duración de las fases de envío de mediciones, tomando en cada caso un factor de seguridad. El período hasta la siguiente ronda de recolección de datos se adoptó igual a 25 segundos. Este valor depende de las aplicaciones, pero si resulta más extenso, aún en el orden de 10 o 15 minutos, no afecta las conclusiones alcanzadas, ya que una vez finalizadas las actividades los dispositivos quedan inactivos, estado en el cual el consumo de energía resulta mínimo, como puede apreciarse en la tabla 1.

Definición	Parámetro	Valor[seg]
Duración de la fase de definición de <i>clusters</i>	T_ARMADO_CLUSTER	1,0
Duración de la fase de definición del árbol de CH	T_ARMADO_RUTA_SINK	0,1
Período de transmisión de un dato hasta el CH	T_SLOT_DATOS	0,02
Duración de la fase de transmisión intracluster	T_ENVIO_DATOS	0,8
Período de transmisión de un agregado hasta el <i>sink</i>	T_SLOT_AGR	0,1
Duración de la fase de transmisión de los CH	T_ENVIOS_MSG_AGR	3.0
Período hasta siguiente ronda de medición	T_RONDA	25,0

Tabla 6. Valores para la temporización del funcionamiento del algoritmo en simulación.

Se simuló el funcionamiento de CLUDITEM utilizando la temporización antes definida, y se obtuvieron valores promedio y máximos del consumo de los dispositivos, que se presentan en la tabla 7, donde puede apreciarse que los valores clasificados por rol y tipo de ronda tienen poca dispersión en cada categoría. También se muestra que el consumo de las cabeceras, como se esperaba, resulta significativamente superior al de los nodos comunes, lo que resalta la importancia del mecanismo de rotación de roles para lograr balancear la carga de trabajo entre los dispositivos.

Rol del nodo y tipo de ronda	Consumo promedio [W]	Consumo máximo [W]
NC en ronda sin reconfiguración	0,028182	0,028192
NC en ronda con reconfiguración	0,069103	0,069479
CH en ronda sin reconfiguración	0,134007	0,134024
CH en ronda con reconfiguración	0,1749267	0,175217

Tabla 7. Consumos de acuerdo al rol de los nodos y el tipo de ronda.

A partir de los consumos obtenidos, se utilizó la expresión (16) para establecer el valor óptimo de X , considerando que todos los dispositivos disponían de 12 Joules de energía inicial, adoptándose 17 rondas para reconfiguración. A continuación, para verificar el cálculo teórico, se realizaron 10 simulaciones con dicho valor, y también con otros valores mayores y menores, arribándose a los resultados que resume la tabla 8. La salida de servicio de la red se estableció a partir del requerimiento de QoS ya utilizado, referido a 5 rondas de medición en las que se supera el 5% de pérdida de mensajes en la estación base.

Simulación	Ronda de fin de servicio con $X = 15$	Ronda de fin de servicio con $X = 16$	Ronda de fin de servicio con $X = 17$	Ronda de fin de servicio con $X = 18$	Ronda de fin de servicio con $X = 19$
1	255	266	277	274	267
2	254	266	277	274	268
3	253	266	277	274	268
4	255	266	277	274	268
5	254	266	277	274	268
6	255	266	277	274	266
7	255	266	273	271	267
8	255	266	277	274	268
9	255	266	278	274	268
10	255	266	278	274	267

Tabla 8. Ronda de salida de servicio de la red para distintos valores de X

Los resultados ponen en evidencia que el cálculo teórico de X es efectivamente el más adecuado, ya que si se incrementa o se disminuye su valor, la red sale de servicio antes por

incumplimiento de los requerimientos establecidos. Las mayores diferencias ocurridas al disminuir el valor de X se atribuyen al hecho de que los nodos tienen la oportunidad de postularse para cabeceras una mayor cantidad de veces. El rol de CH es más exigente respecto del consumo y por lo tanto es previsible que agoten antes sus reservas provocando desconexiones tempranas en la red. Cuando se incrementa X , los nodos que asumen el rol de CH lo hacen por un número más elevado de rondas, lo que también tiene un impacto negativo sobre sus reservas. Por lo tanto, el valor teórico consigue para la energía inicial adoptada y la temporización definida, el mejor balance para la rotación de roles, brindando un tiempo de vida más prolongado para el sistema, compatible con los requerimientos de las aplicaciones.

El análisis continuó entonces en base al valor óptimo de $X = 17$ para el momento de rotación de roles, y se enfocó la atención sobre la energía remanente en las 10 simulaciones asociadas, una vez que la red sale de servicio en la ronda indicada en la tabla 8.

Los resultados del estudio se reportan en la tabla 9, donde puede apreciarse que el total de energía residual en el sistema es reducido y aproximadamente el mismo para todas las simulaciones realizadas.

Simulación	Energía remanente total [J]
1	201,018036
2	210,745063
3	201,435832
4	213,520508
5	213,163628
6	208,352504
7	202,242503
8	215,757158
9	212,008508
10	213,441527

Tabla 9. Energía residual en la red cuando sale de servicio por incumplimiento de QoS

La situación descrita a través de los valores de la tabla 9 es coherente con el objetivo de balancear la carga de trabajo en la red. Sin embargo, dicha tabla presenta información global del sistema, y por lo tanto no brinda detalles respecto del estado particular de cada nodo sensor. Por este motivo, se analizó a continuación la energía residual de los dispositivos, quedando en evidencia que en todas las simulaciones podían definirse claramente cuatro grupos de nodos con energía remanente muy similar entre sí, como muestran los resultados resumidos en la tabla 10.

Simulación	Grupo 1		Grupo 2		Grupo 3		Grupo 4	
	Cantidad de nodos	Energía promedio del grupo [J]	Cantidad de nodos	Energía promedio del grupo [J]	Cantidad de nodos	Energía promedio del grupo [J]	Cantidad de nodos	Energía promedio del grupo [J]
1	3	3,4968	108	1,6991	6	1,1704	27	0,0000
2	4	3,4961	111	1,6988	7	1,1701	22	0,0000
3	1	3,4962	111	1,6989	8	1,1701	24	0,0000
4	4	3,4962	114	1,6989	5	1,1711	21	0,0000
5	5	3,4692	111	1,6707	10	1,0364	18	0,0000
6	2	3,4966	113	1,6991	8	1,1700	21	0,0000
7	4	3,4958	106	1,6987	7	1,1704	27	0,0000
8	4	3,4968	116	1,6991	4	1,1682	20	0,0000
9	4	3,4689	113	1,6710	9	1,0349	18	0,0000
10	2	3,4686	118	1,6710	9	1,0357	15	0,0000

Tabla 10. Grupos de nodos clasificados por energía residual cuando sale de servicio la red

La situación de los nodos de cada grupo identificado en la tabla 10, puede resumirse de la siguiente manera:

- *Grupo 1:* Compuesto por un conjunto reducido de nodos que nunca asumieron el rol de CH durante el tiempo de vida de la red, y que por lo tanto son los que menos energía consumieron durante el funcionamiento. Esta situación era previsible, ya que

se definieron condiciones para postulación a cabecera, que pueden no cumplirse en algunas rondas de reconfiguración.

- *Grupo 2:* Constituido por la mayoría de los dispositivos, que cumplieron el rol de cabecera una única vez, durante una ronda completa de reconfiguración (17 rondas de envío de mediciones).
- *Grupo 3:* Integrado por pocos nodos que asumieron el rol de CH en una ronda de reconfiguración completa la primera vez, pero que se postularon por segunda vez en la última reconfiguración de la vida útil de la red, por lo que, en esa oportunidad, cumplieron con las responsabilidades del rol sólo por 4 o 5 rondas de medición.
- *Grupo 4:* Este conjunto de nodos, cumplió el rol de cabecera por dos rondas de reconfiguración completas. Por este motivo agotaron su energía, pese a lo cual, la red cumplió con el requerimiento de QoS, finalizando su vida útil en rondas posteriores a la correspondiente a una rotación completa del rol de CH.

La presencia de los conjuntos de nodos identificados a partir del análisis de la energía remanente, es coherente con las hipótesis de trabajo planteadas durante el desarrollo del algoritmo y los lineamientos de diseño adoptados. La situación ideal de un único CH por celda, que cumple su rol durante una ronda de reconfiguración completa, es la situación de la mayoría de los dispositivos de la red. Sin embargo, el algoritmo permite la postulación potencial de dos cabeceras por división de la cuadrícula, atendiendo a los resultados obtenidos durante la evaluación de la gestión de la estructura de encaminamiento. Esta situación, permite que existan nodos que asumen el rol más exigente, en cuanto a consumo de recursos, en más de una oportunidad, ocasionando que algunos grupos reducidos consuman más que el promedio, e incluso, agoten sus recursos por completo. Es importante destacar, que los dispositivos en la misma situación o grupo terminan su funcionamiento con una energía residual muy similar. Esto permite afirmar, que las técnicas aplicadas para balancear la carga de trabajo resultaron exitosas.

Luego de todas las fases de evaluación realizadas, puede concluirse que CLUDITEM cumple adecuadamente con sus objetivos y lineamientos de diseño, resultando un algoritmo de encaminamiento que puede ajustar su funcionamiento mediante los parámetros incorporados a distintas situaciones de trabajo. En este sentido es capaz de proveer un

grado interesante de flexibilidad para abordar el conjunto de aplicaciones elegidas, cumpliendo los requerimientos de QoS asociados.

Capítulo 6. Prototipo de nodo CLUDITEM.

Las redes inalámbricas de sensores suelen trabajar bajo fuertes restricciones de energía y esta limitación tiene un gran impacto en la definición de la plataforma de trabajo de los nodos [84]. CLUDITEM tiene en cuenta este aspecto particularmente importante, buscando por un lado mantener los transceptores apagados todo el tiempo posible, y por otro balancear la carga de trabajo entre los nodos de la red. Sin embargo, el diseño de la plataforma de *hardware* de los dispositivos es también un aspecto fundamental para prolongar el tiempo de vida del sistema, y constituye un área en la que se ha trabajado extensamente como se reporta en [34] y [59].

6.1 Plataformas *hardware* para redes de sensores

Las plataformas *hardware* (HW) para nodos sensores que se han presentado desde el comienzo de este siglo fueron inicialmente desarrolladas en el ámbito académico, como ocurrió con las distintas versiones de las Mica Motes en la Universidad de Berkeley [35]. Sin embargo, a partir del reconocimiento de la potencialidad de esta nueva tecnología, diversas empresas comerciales incursionaron en el área, como Intel con su Mica2 [52] o la empresa Crossbow [14] que ha comercializado varios de los desarrollos realizados por la Universidad de Berkeley.

Es importante destacar que la mayoría de las plataformas para RISI incluyen procesadores pequeños de 8 o 16 bits, que cuando no están realizando las tareas que tienen asignadas en la red permanecen en estado de bajo consumo [60]. Sin embargo, al ampliarse el campo de aplicación de las redes de sensores e incorporarse áreas de trabajo de complejidad creciente, se han incrementado las demandas de capacidad de cómputo incorporándose para estos casos procesadores de mayores prestaciones, que traen aparejado un aumento importante en el consumo [59]. Además, muchas aplicaciones agregan a los requerimientos de bajo consumo y capacidad de procesamiento la necesidad de plataformas flexibles y rápidas capaces de adaptarse a cambios en el ambiente de trabajo, o incluso, alterar su comportamiento en base al análisis de la información recolectada. Por estos motivos, muchos investigadores han comenzado a explorar otras opciones para las plataformas HW

incorporando dispositivos de lógica reconfigurable que, en muchos casos, reparten sus responsabilidades con los procesadores que tradicionalmente han sido la base de la arquitectura de los nodos sensores [16] [75].

6.1.1 Plataformas reconfigurables

Las FPGA (*Field Programmable Logic Arrays*), no han sido utilizadas como base de plataformas para redes de sensores por razones de costo y consumo, dos requerimientos importantes y habituales en las RISI [60]. Sin embargo, estos dispositivos poseen varias características que los vuelven muy atractivos para algunas aplicaciones:

- *Flexibilidad:* pueden configurarse en forma rápida y sencilla, esto no sólo reduce el tiempo de desarrollo de la aplicación, sino que permite modificar la funcionalidad del dispositivo sin retirarlo del circuito. Los integrados actuales incluyen la posibilidad de realizar una reconfiguración parcial reprogramando sólo una parte del *chip* mientras el resto permanece sin cambios. Además, esta actividad puede realizarse en forma dinámica, o sea, mientras una parte del dispositivo se reconfigura, el resto del mismo continúa desarrollando las tareas que le corresponden en la red.
- *Velocidad de respuesta:* Las FPGA poseen una arquitectura eminentemente paralela, lo que les permite trabajar a velocidades muy superiores a las de un procesador secuencial. Esto las vuelve un recurso muy valioso en aplicaciones exigidas respecto de la frecuencia de trabajo necesaria.
- *Capacidad de procesamiento:* Los altísimos niveles de integración existentes, y la heterogeneidad de las arquitecturas de los dispositivos reconfigurables actuales, convierten a las FPGA en integrados dotados de grandes capacidades de cómputo.

En la literatura se encuentra, por los motivos antes expuestos, un interesante número de propuestas ligadas a la incorporación de dispositivos reconfigurables en las plataformas HW de los nodos sensores. Las mismas son variadas y a continuación se enumeran algunas de ellas.

Los autores de [10] proponen una plataforma basada en una FPGA con un procesador embebido que utilizan para realizar una detección confiable asociada con un veloz reporte

de eventos aleatorios a la estación base. Su diseño permite reemplazar o agregar nodos a la red de forma muy simple gracias a la flexibilidad de los dispositivos reconfigurables.

En [13], se trabaja con redes de sensores heterogéneas que se relacionan con otras infraestructuras existentes en el ambiente en el que se las despliega. Para ello, utilizan nodos especiales con una arquitectura basada en FPGA, que cumplen el doble rol de ser interfaces con otras redes y coordinadores de *clusters* de la RISI. Se enfocan en aplicaciones que necesitan adaptar la función de agregación de datos a aplicar en los CH, en base a alteraciones en el ambiente o a cambios en los requerimientos del sistema una vez que la red ya está en actividad. Exploran la reconfiguración parcial dinámica de los dispositivos, y obtienen buenos resultados asociados con el desempeño de los CH cuando se incrementa el número de operaciones de agregación y la cantidad de nodos de la red.

Wilder et al. presentan en [75] el ambiente REWISE, que permite la reconfiguración de la lógica programable de los nodos a través de la infraestructura de comunicaciones de la red de sensores. Esta operación puede ser solicitada por la estación base o por algún nodo de la red, y la plataforma se orienta hacia aplicaciones que necesitan modificaciones una vez que están en funcionamiento por cambio de sus objetivos, actualizaciones o cambios significativos en el ambiente de trabajo.

Los autores de [27] y [70] trabajan con aplicaciones ligadas al seguimiento de objetivos críticos, en las que realizan estimaciones de trayectoria utilizando filtros de Kalman implementados sobre una arquitectura modular basada en FPGA. Utilizan configuración dinámica parcial de los dispositivos para adaptar el comportamiento de los nodos en base a la situación en el entorno, habilitando distintos módulos de detección de objetivos para evaluarlos o elegir el más adecuado de ellos. Los nodos pueden seguir con sus actividades mientras se reprograman los módulos necesarios.

En [17] se describe la implementación de un algoritmo adaptativo de reducción de datos sobre una FPGA incorporada a los nodos de la red, con el cual consiguen una reducción del orden del 95% de los datos a transmitir por los dispositivos. La flexibilidad de la FPGA permite la adecuación de los parámetros del filtro cuando se superan los umbrales de error admitidos.

En el trabajo [61] se describen aplicaciones de visión, como la detección de objetos a partir de imágenes de video, que típicamente son exigentes en consumo de energía y costosas computacionalmente. Los autores incorporan a la plataforma Mica2 un módulo con un microcontrolador de bajo consumo y un CPLD (*Complex Programmable Logic Device*) que proporciona la velocidad necesaria para la captura de imágenes.

Los autores de [36] proponen una plataforma modular y flexible para obtener prototipos de nodos sensores en forma sencilla y rápida. Para ello incluyen una FPGA con los objetivos de poder emular diversas arquitecturas, aún aquellas con alto grado de complejidad, implementar interfaces variadas con sensores y transceptores, e incluir sistemas de depuración y supervisión de la funcionalidad de los prototipos sin interferir con su operación.

Portilla presenta en [60] una arquitectura modular para un nodo sensor en la que se incorpora en la capa de procesamiento una FPGA para dotarla de flexibilidad y gran capacidad de cómputo. Este autor ha estudiado el impacto de la inclusión del dispositivo, así como la potencialidad que brinda la plataforma que propone, que permite la reconfiguración parcial dinámica de la FPGA.

En [73] se propone una plataforma de nodo sensor basado en una FPGA de muy bajo consumo de tecnología Flash (IGLOO de la empresa Actel), con un núcleo *software* de procesador embebido (ARM Cortex-M1). Estos nuevos dispositivos pueden asumir un estado de muy bajo consumo, en el cual preservan la configuración, y además asumen el estado activo muy rápidamente. Los autores afirman que su plataforma alcanza un desempeño mejor que el de una plataforma basada en un microcontrolador tradicional consumiendo lo mismo.

La mayoría de los investigadores coinciden en que, pese a los avances de la tecnología que ha permitido desarrollar dispositivos con mejores prestaciones, las FPGA aún presentan el problema de mayor costo y consumo respecto de los microcontroladores, lo cual es un inconveniente importante para su utilización en redes de sensores. Sin embargo, considerando las ventajas que proporcionan en flexibilidad, velocidad y capacidad de procesamiento, varios de ellos apuestan al avance tecnológico y opinan que en un futuro

cercano la utilización de estos dispositivos será más habitual en las plataformas HW para RISI [16] [27] [36] [60].

6.2 *Arquitectura del prototipo de nodo*

En esta tesis se desarrolla un prototipo de nodo sensor capaz de soportar el algoritmo de encaminamiento CLUDITEM. El diseño está enfocado a verificar la viabilidad de implementación y no constituye una arquitectura definitiva. Por este motivo para elegir el soporte HW se priorizó la flexibilidad proporcionada por los dispositivos de lógica reconfigurable, que permitieron evaluar la bondad del algoritmo desarrollado realizando las pruebas y ajustes necesarios sin necesidad de recurrir a cambios de componentes en la placa. Tampoco se realizaron, por la misma razón, pruebas o consideraciones respecto de la energía consumida por el prototipo, enfocándose la actividad a evaluar la posibilidad de implementar los módulos necesarios para soportar la funcionalidad de CLUDITEM.

En el capítulo 1 ya se ha comentado que la arquitectura de un nodo sensor puede dividirse en cinco módulos principales, esquematizados en la figura 2 de dicho capítulo, a saber: adquisición de datos, adecuación de señal, procesamiento, comunicaciones y potencia. El módulo de la unidad de proceso propuesto para implementar CLUDITEM, se ocupa de realizar las tareas inherentes al tratamiento local de la información y a la implementación del algoritmo de encaminamiento, definiendo los mensajes de datos y control asociados que luego transfiere al módulo de comunicaciones. Por lo tanto, su funcionalidad es fundamental, e involucra potencialmente una carga importante de trabajo que se reparte entre módulos *software* (SW) y *hardware* de acuerdo con la naturaleza de las tareas a abordar.

El análisis realizado para la definición de los bloques que constituyen la arquitectura del nodo, se basó en los requerimientos de las aplicaciones de interés y los objetivos del algoritmo. Las actividades de supervisión ambiental en los dominios elegidos, suponen la adquisición periódica de datos fuertemente correlacionados en toda el área bajo estudio, pero la demora entre la medición y la llegada de la información a la estación base no es crítica.

Las decisiones asociadas con la funcionalidad de CLUDITEM son numerosas e involucran un número importante de variables. Se deben procesar los mensajes recibidos sólo cuando corresponde, analizar el contenido de sus campos teniendo en cuenta la información local y elaborar nuevos mensajes para transmitir a la red. Esta funcionalidad podría haberse implementado utilizando una máquina de estados finitos (MEF) que soportara el algoritmo, pero la misma debería haber contado con un número importante de estados, conduciendo a un diseño y prueba dificultosos. Si el número de estados necesarios en una MEF resulta elevado y los requerimientos de tiempo de respuesta no son muy exigentes, es mucho más conveniente asignar la tarea a un módulo *software*. Este fue el enfoque adoptado, y se asignaron a un procesador las decisiones ligadas al encaminamiento y envío de información, responsabilizándolo del procesamiento local de los datos y de la definición de los mensajes necesarios.

CLUDITEM realiza un control estricto de la duración de sus fases de funcionamiento para organizar el trabajo y poder mantener apagados los nodos el máximo tiempo posible. Esta característica hace necesario definir un bloque HW de encendido del nodo, que permanece activo, contempla los desfasajes ligados a las distintas etapas de funcionamiento y se ocupa de activar y desactivar otros bloques de la arquitectura. Por otro lado, como se detalló en el capítulo 4 existen un conjunto de temporizadores estrechamente ligados con la funcionalidad del algoritmo, que son los encargados de determinar los momentos de toma de decisiones y cumplimiento de responsabilidades. Es por este motivo, que se decidió implementarlos también como bloques HW, capaces de interrumpir al procesador para indicarle que ha llegado el momento de llevar adelante acciones específicas.

En resumen, las unidades de comunicaciones y procesamiento correspondientes a la arquitectura de un nodo capaz de implementar el algoritmo CLUDITEM, están constituidas por los módulos de alto nivel que se muestran en la figura 29. El módulo de encendido del nodo, es el encargado de activar el transceptor y el procesador, que normalmente deben permanecer en estado de bajo consumo. Los temporizadores que interrumpen al procesador durante su actividad para lanzar diversas tareas se esquematizan aparte, pero también se planificaron como bloques de HW cableado. La figura no incluye las unidades de medición, adaptación de señales y potencia.

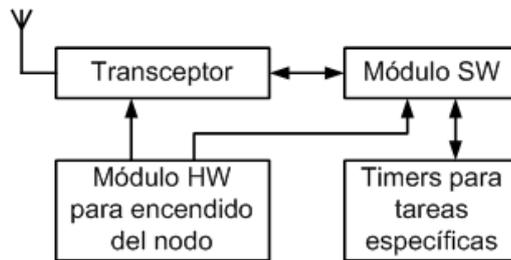


Figura 29. Módulos de alto nivel de las unidades de procesamiento y comunicaciones.

La decisión de utilizar un dispositivo de lógica reconfigurable como base para el prototipo, permitió implementar tanto el módulo SW como los bloques HW correspondientes al encendido y temporización del funcionamiento en el integrado de la FPGA, eliminando las conexiones entre *chips* y los inconvenientes que traen aparejados [43] [76], como se describe en la sección siguiente.

6.3 El módulo software

Los altísimos niveles de integración existentes, y la heterogeneidad de las arquitecturas de los dispositivos reconfigurables actuales, convierten a las FPGA en integrados dotados de grandes capacidades de cómputo [56]. Actualmente existen numerosos núcleos *software* de procesadores que pueden ser embebidos en dichos dispositivos. Entre ellos se pueden mencionar, PicoBlaze de libre distribución [79], Nios II de la empresa Altera [6], y MicroBlaze de Xilinx [82].

En este sentido, se observa un incremento significativo del uso de dispositivos reconfigurables con núcleos de procesadores embebidos, que en opinión de varios autores parece consolidarse como tendencia firme para el diseño electrónico [28] [55]. En particular, las plataformas FPGA y los ambientes de desarrollo asociados, resultan muy convenientes para llevar adelante la tarea de partir e interconectar los módulos asignados a funciones *hardware* y *software* dentro de la misma pastilla [3] [25] [76].

La implementación del módulo SW de la unidad de proceso se realizó con el enfoque antes mencionado, incorporándose núcleos *software* de procesadores de distintas características a la arquitectura del nodo sensor. Los ensayos realizados fueron incrementales, se comenzó con prototipos cableados basados en un procesador muy sencillo, y se continuó con un

prototipo inalámbrico de mayores prestaciones, capaz de soportar la arquitectura definida en la sección 6.2. Las implementaciones llevadas a cabo y los resultados obtenidos se describen en las secciones siguientes.

6.3.1 Implementación con un procesador de 8 bits

La primera experiencia realizada sobre el módulo SW, tuvo como objetivo explorar el nivel de complejidad asociado con la implementación de las responsabilidades asignadas al bloque. Se trabajó con el microprocesador Picoblaze, un núcleo *software* de propiedad intelectual (IP) optimizado para la familia Spartan 3 de Xilinx [79]. Se trata de un procesador RISC de arquitectura muy simple que junto con un bloque RAM que contiene el programa, de a lo sumo 1Kbyte, se embebe en la FPGA de la placa Spartan-3 Starter Kit Board [80], permitiendo la conexión con otros bloques HW de la arquitectura.

Es importante destacar que esta prueba preliminar, no incluyó el módulo HW de encendido ni los temporizadores para tareas específicas necesarios para la arquitectura general del nodo descrita en la sección 6.2. El trabajo se centró en el análisis de la opción de implementación del módulo SW con el procesador PicoBlaze, y su interconexión con los periféricos necesarios para su correcto funcionamiento.

En función del objetivo fijado, el análisis se recortó a la interacción de dos nodos: el transmisor (NTx) y el receptor (NRx), que se presentan en la figura 30.

La transmisión en radiofrecuencia, fue reemplazada por una transmisión serie con dos hilos, incorporándose al diseño el núcleo IP de una UART (*Universal Asynchronous Receiver-Transmitter*) de libre distribución [11], bajo el estándar RS-232, compatible con el puerto serie de la placa utilizada. La comunicación serie se estableció, entonces, utilizando el módulo Uart_Tx para el nodo transmisor y el módulo Uart_Rx para el receptor.

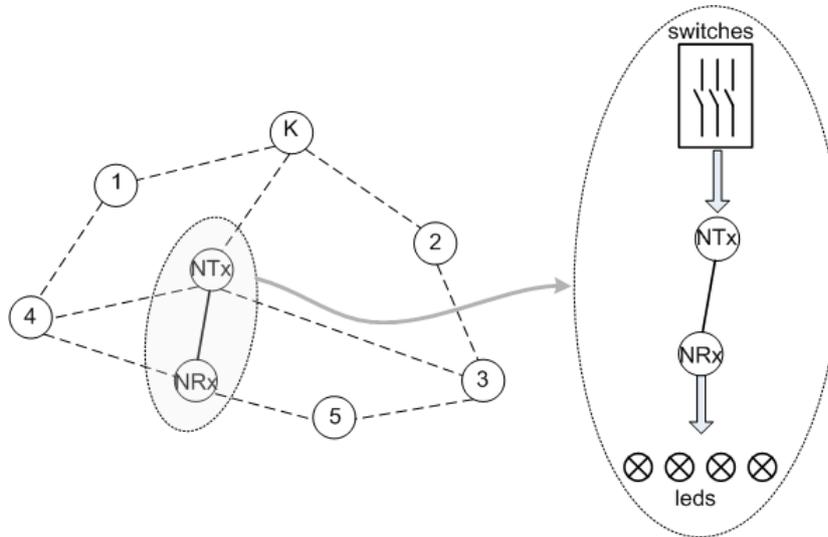


Figura 30. Conexión entre los nodos NTx y RTx en el prototipo cableado

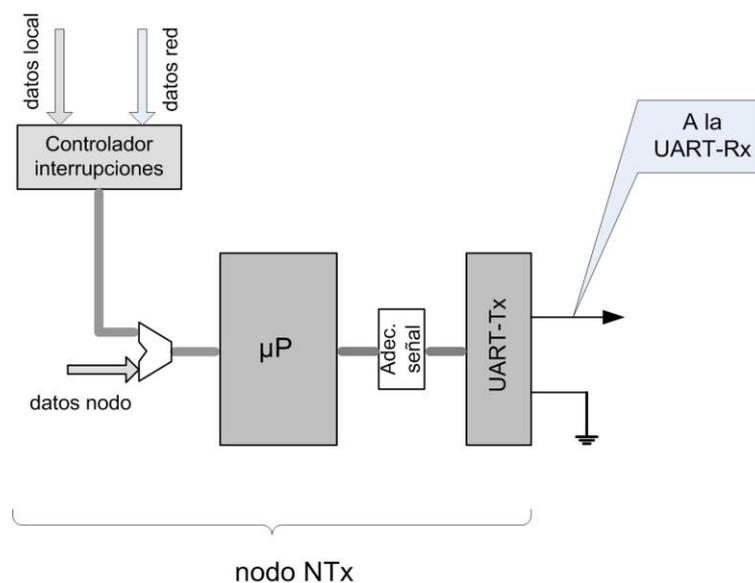


Figura 31. Diagrama simplificado de los nodos utilizando PicoBlaze: Nodo transmisor NTx.

Para ensayar la funcionalidad del diseño, se decidió que el procesador fuera capaz de tratar la información proveniente de distintas fuentes. Las figuras 31 y 32 presentan los bloques constitutivos de los nodos implementados, y también grafican el ingreso de datos de distintos orígenes a cada uno de ellos:

- Mensajes enviados por otros nodos de la red (datos red).
- Datos de identificación del nodo (datos nodo).

- Datos locales correspondientes a la medición de variables del medio (datos local).

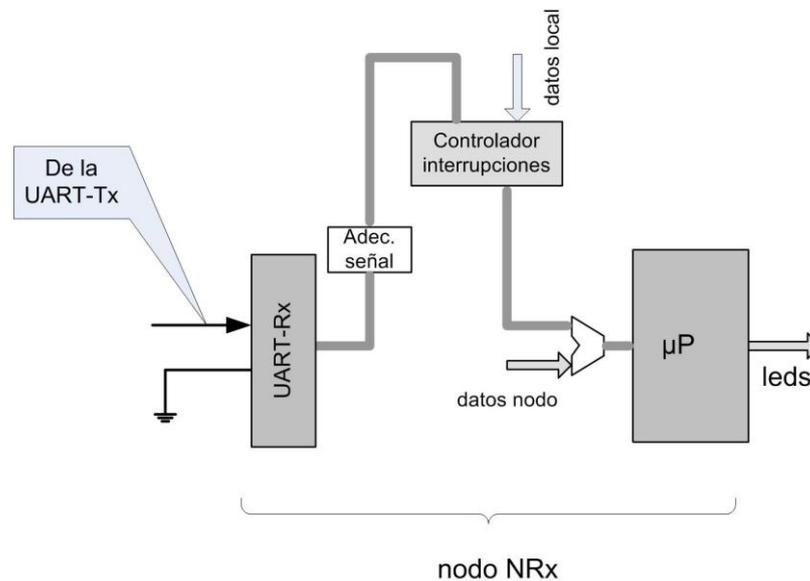


Figura 32. Diagrama simplificado de los nodos utilizando PicoBlaze: Nodo receptor NRx.

Los datos locales y los mensajes provenientes de la red ingresan al procesador con un pedido de interrupción. Como el PicoBlaze dispone de una única entrada para este fin, fue necesario implementar un módulo controlador de interrupciones que se desarrolló utilizando el lenguaje de descripción de *hardware* VHDL (*Very High Speed Integrated Circuit Hardware Description Language*). Asimismo se agregó una interface para adecuar los tiempos de las señales que entrega la UART a los requerimientos del microprocesador.

Las situaciones abordadas fueron las siguientes:

- *Situación 1:* el nodo NTx, envía por la UART de transmisión un mensaje simplificado de estructura de *cluster*, de 8 bits de longitud. Este mensaje puede consistir en su postulación como CH o en el anuncio del nivel con que se une a un *cluster* liderado por otro nodo. En ambos casos, el procesador embebido en NTx elabora el mensaje a transmitir a partir de los datos obtenidos desde las llaves de la placa. Por su parte, el nodo NRx recibe el mensaje de NTx desde la UART de recepción y, en caso de corresponder procesarlo, presenta el mensaje a transmitir en los diodos led de la placa. En caso contrario, no realiza ningún procesamiento y se

limita a presentar en los leds el mensaje enviado por NTx sin modificaciones. Cabe destacar que si el tipo de mensaje recibido no corresponde al armado de estructura de *cluster*, es desechado.

- Situación 2: Cuando corresponde tratar mediciones locales, a través de llaves, ambos nodos se limitan a presentarlas en los diodos led.

Se programó el microprocesador en código ensamblador, y se instanciaron, utilizando el ambiente ISE de Xilinx [77], los núcleos del PicoBlaze, las UARTs y los módulos en VHDL diseñados para lograr la integración de todos los componentes en el sistema final.

Se realizaron dos procesos de simulación a saber: con la herramienta IDE, perteneciente al ambiente de desarrollo del PicoBlaze [79], se simuló el comportamiento del programa en ensamblador corriendo en el microprocesador; con la herramienta ModelSim, se simuló el comportamiento de todo el sistema. Luego de un resultado satisfactorio de estos procesos de simulación, se realizó la síntesis sobre las placas de desarrollo, verificando el comportamiento del sistema conforme al modelo diseñado.

Si bien la implementación resultó viable y las pruebas satisfactorias, se hicieron evidentes algunas dificultades y limitaciones. El procesador PicoBlaze utilizado, posee una arquitectura muy sencilla que demanda pocos recursos de pastilla, pero hace necesario incorporar al diseño los periféricos requeridos para su funcionamiento. Estos dispositivos basados en núcleos IP deben ser adaptados y conectados en el sistema, tarea que puede resultar ardua si la complejidad de la plataforma *hardware* es grande. Además, la programación del procesador se realiza en ensamblador, lenguaje de bajo nivel, que demanda un esfuerzo considerable si se pretende implementar el comportamiento completo del nodo.

Teniendo en cuenta las consideraciones mencionadas, se decidió migrar el diseño al núcleo IP de un procesador con mayores prestaciones que permite trabajar en un lenguaje de alto nivel y dispone de un ambiente de desarrollo que facilita la conexión de periféricos.

6.3.2 Implementación con un procesador de 32 bits

El procesador seleccionado fue el núcleo de *software* MicroBlaze de 32 bits de la empresa Xilinx, que cuenta con un ambiente de desarrollo integrado para personalizar la plataforma *hardware* incorporando distintos tipos de periféricos, que se disponen en bibliotecas con sus correspondientes *drivers*, lo que reduce significativamente la cantidad de módulos necesarios para adaptación e interface [81]. Además, la programación *software* puede realizarse en un lenguaje de alto nivel como C, lo que facilita en gran medida la codificación de algoritmos complejos [82]. Estas características son muy ventajosas, ya que dependiendo de los requerimientos de cada aplicación, puede decidirse el número y tipo de periféricos a interconectar utilizándose sólo los recursos de pastilla necesarios.

Las características de Microblaze y su ambiente de desarrollo permitieron incorporar desde biblioteca, el módulo controlador de interrupciones, la UART, los temporizadores y los módulos GPIO (*General Purpose Input/Output*) para comunicarse con los recursos de la placa, que se utilizaron para ingresar la información de identificación y mostrar resultados del procesamiento local de mensajes. Además, se incorporó al prototipo un módulo de comunicación inalámbrica que reemplazó a la conexión de dos hilos implementada en las pruebas preliminares. El módulo de comunicación que se conectó con la placa Spartan 3 Starter Kit Board que implementa los módulos antes mencionados, fue el de la empresa Digi Internacional que soporta el estándar IEEE 802.15.4 elegido para las capas inferiores de CLUDITEM [21]. Estos dispositivos pueden programarse tanto local como remotamente para definir su modalidad de funcionamiento y los valores de los registros que permiten personalizarla. Esta característica es fundamental para implementar las distintas fases de CLUDITEM, en las cuales no sólo el módulo debe alternar entre el estado activo y el de bajo consumo, sino también cambiar la modalidad de comunicación con sus vecinos.

La implementación asociada con esta segunda opción para el módulo *software* incluyó la definición, adaptada para las condiciones del prototipo, del bloque de activación del nodo y los temporizadores para tareas específicas contemplados en la arquitectura general propuesta en la sección 6.2.

Es importante destacar que la plataforma elegida para el prototipo basada en un dispositivo reconfigurable, permitió implementar el diseño descrito en la sección 6.3.1 basado en un procesador de 8 bits y también este nuevo enfoque, adoptado a partir de los resultados de las primeras pruebas, en la misma placa de desarrollo. Esta situación pone de manifiesto la flexibilidad de la plataforma para evaluar distintas arquitecturas, que constituyó el motivo principal para su elección.

6.4 Ensayos con el prototipo inalámbrico basado en MicroBlaze

El objetivo del prototipo desarrollado fue verificar la viabilidad de implementación de CLUDITEM mediante los módulos SW y HW propuestos. Para ello se implementaron dos nodos idénticos, como el mostrado en la figura 33, plasmando la arquitectura antes definida.

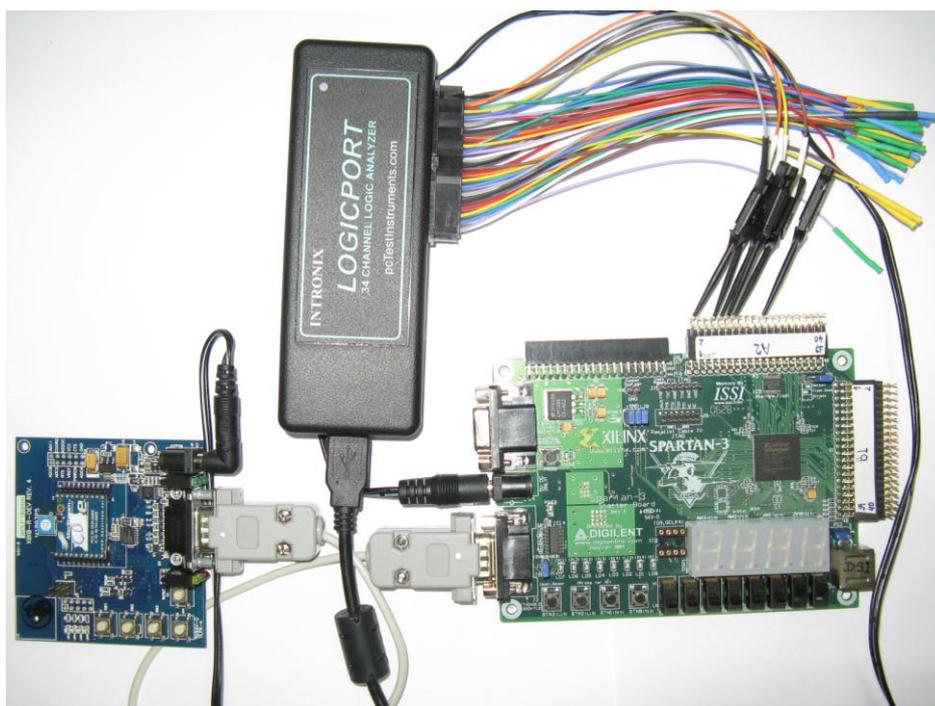


Figura 33. Imagen del prototipo de nodo sensor implementado.

6.4.1 Detalle de los módulos implementados para las pruebas

La arquitectura del prototipo para realizar las pruebas definidas en el apartado anterior quedó constituida por los módulos mostrados en la figura 34.

Los bloques definidos en el ambiente de desarrollo EDK del procesador MicroBlaze, y el módulo HW encargado del encendido del nodo, desarrollado en el ambiente ISE [77], se implementaron en la FPGA de la placa Spartan 3 Starter Kit Board, y se conectaron vía RS-232 con el módulo XBee de comunicación inalámbrica.

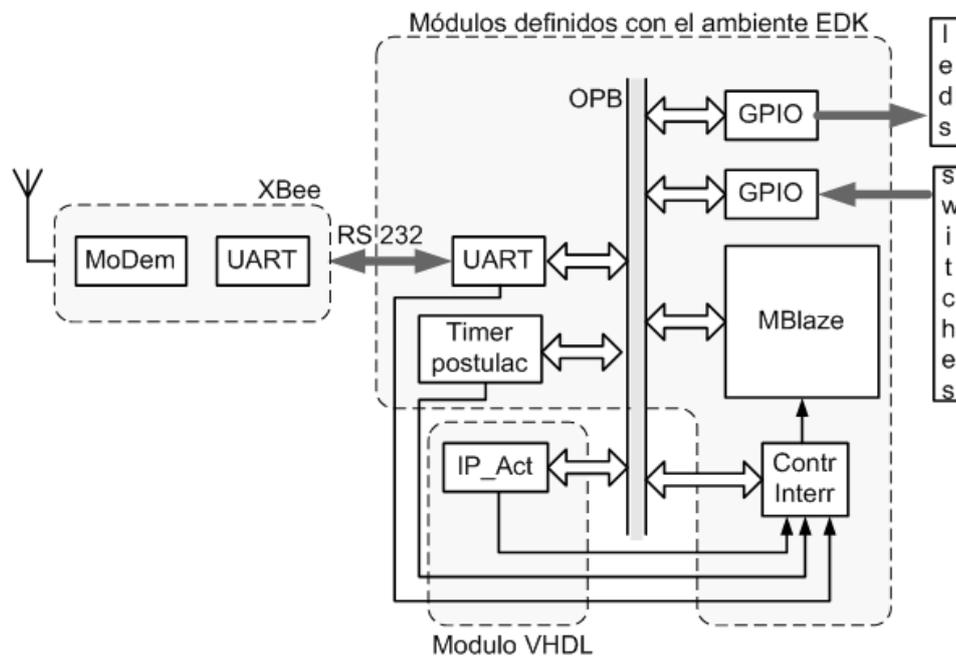


Figura. 34. Esquema de la arquitectura del nodo utilizando el procesador MicroBlaze

El detalle de los bloques de la unidad de proceso del nodo mostrados en la figura 34 es el siguiente:

- *Timer postulac*: Este módulo se ocupa de lanzar la rutina de postulación cuando se cumplen las condiciones establecidas en el algoritmo. El tiempo de postulación del nodo i (T_{Posi}) se calcula según la expresión (19), obtenida a partir de la expresión (2) de la sección 4.5.1, donde el parámetro Kp_i del esquema TDMA utilizado en CLUDITEM se define mediante el identificador del nodo.

$$T_{Posi} = T_p \cdot (1 + ID_i) \quad (19)$$

En (19), T_p es el tiempo de postulación establecido en la sección 4.5.1 mediante la expresión (3), e ID_i se refiere al identificador del nodo i .

- *Módulos GPIO*: Implementados con IP disponibles en bibliotecas para conectar los recursos de la placa (llaves y diodos led) con el procesador embebido.
- *Módulo UART*: Implementado con un IP de biblioteca (UartLite) que se comunica vía RS-232 con la UART del módulo XBee, que implementa el bloque transceptor.
- *Controlador de Interrupciones*: Implementado con un IP de biblioteca personalizado para recibir tres pedidos de interrupción (IP_Act, Timer postulac y UART).
- *IP_Act*: Es un módulo HW desarrollado en VHDL, que se ocupa de despertar en el momento apropiado los bloques que intervendrán en el procesamiento de la información y el intercambio de mensajes en la red (procesador y transceptor). Además este módulo tiene en cuenta el desfasaje local de los relojes en la fase de armado de *clusters* (T_{DIC}). Por lo tanto, en el inicio de dicha fase, IP_Act interrumpe a Microblaze simulando, en el prototipo, la activación del microprocesador. Una vez cumplido T_{DIC} el procesador habilita el temporizador de postulación del nodo.

Los periféricos personalizados en el ambiente EDK y el módulo hardware IP_Act se conectan con Microblaze a través del bus OPB (*On-chip Peripheral Bus*) de la arquitectura CoreConnect de IBM [78].

Se debe puntualizar que todos los módulos permanecen activos durante las pruebas realizadas, y como ya se mencionó, el encendido se simula en el prototipo mediante interrupciones al procesador que lanzan rutinas en los momentos precisos.

Los módulos descritos plasman las ideas básicas de funcionamiento de CLUDITEM, pero una implementación completa de la funcionalidad del algoritmo deberá incluir un mayor número de temporizadores locales responsables de lanzar las rutinas asociadas con las tareas de las distintas fases de trabajo. Sin embargo, para las pruebas programadas en base al número de dispositivos disponibles los bloques descritos fueron suficientes, y es importante destacar que la funcionalidad no ensayada puede implementarse agregando

módulos del mismo tipo que los ya desarrollados, utilizando los mismos ambientes de trabajo y la misma plataforma HW.

La plataforma HW definitiva para un nodo CLUDITEM, deberá contemplar el apagado y encendido efectivos de todos los bloques *software* y *hardware* posibles, permaneciendo en actividad sólo aquellos ligados al control de tiempos (Módulo IP_Act). Esta tarea queda planteada para un futuro, ya que esta tesis no incluye el desarrollo de una versión definitiva del nodo sensor.

6.4.2 Pruebas realizadas

El reducido número de nodos disponibles para las pruebas, no permitió realizar ensayos completos del funcionamiento del algoritmo, ya que básicamente se pudieron probar situaciones de comunicación inalámbrica entre los dos nodos implementados. Por este motivo, se centró el trabajo en el intercambio de mensajes relacionados con la definición de la estructura de *clusters* del algoritmo, a saber:

- *Situación 1:* Ambos nodos son despertados por el temporizador del módulo HW para encendido de los dispositivos que se muestra en la figura 29. Cuando esto ocurre, ambos nodos lanzan un temporizador local que define su momento de postulación como cabeceras de *cluster*. El nodo 1 se postula primero mediante el envío de un mensaje EC. El nodo 3 recibe el mensaje y se une al *cluster* liderado por el nodo 1 anunciándolo con otro mensaje EC con la información necesaria para que otros nodos vecinos puedan adoptarlo como enlace.
- *Situación 2:* Los nodos 1 y 3 son despertados por sus bloques de encendido local. Ambos disparan sus temporizadores de postulación. En este caso, el nodo 1 no se postula, por lo cual es el nodo 3 el que asume la responsabilidad enviando el mensaje EC correspondiente.

Además de las dos situaciones descritas, se probaron casos erróneos correspondientes a tipos de mensajes no existentes para verificar que los nodos los descartaban sin realizar procesamiento alguno.

6.4.3 Resultados obtenidos

Para realizar las correcciones necesarias y verificar el funcionamiento del prototipo se utilizó el analizador lógico LogicPort [40] para visualizar señales generadas a partir del intercambio de mensajes entre los dos nodos implementados en las situaciones planteadas.

En el ensayo realizado se inicializó el sistema ingresando mediante las llaves de las placas el identificador de cada nodo y lanzando el módulo IP_Act. A partir de ese momento, los nodos intercambiaron los mensajes correspondientes a las situaciones ligadas a la definición de *clusters* descriptas en el apartado anterior.

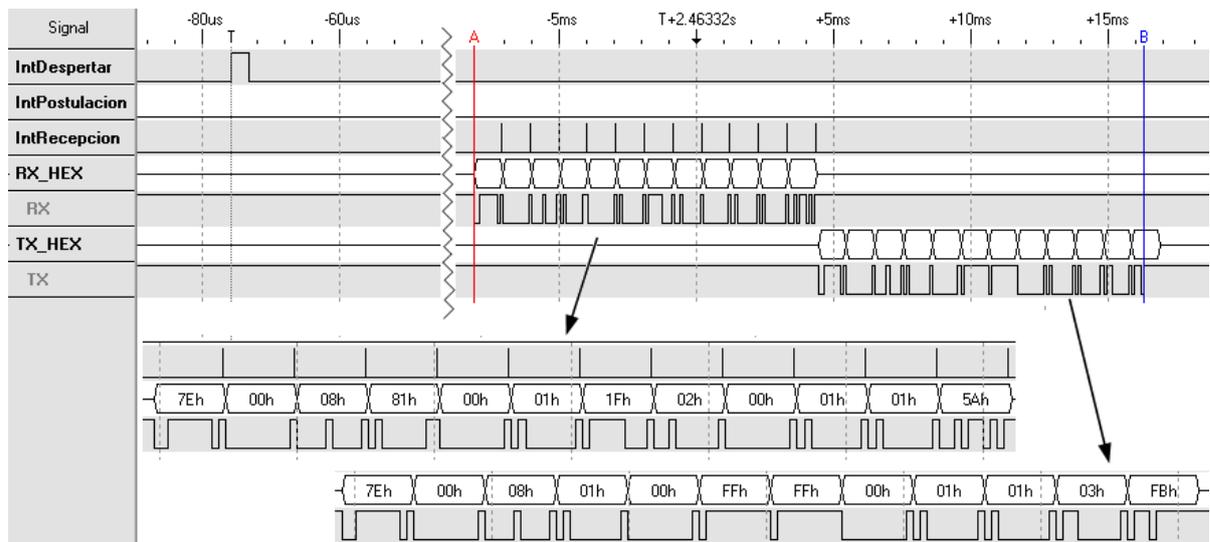


Figura 35. Visualización de señales: Recepción de la postulación del Nodo 1 por parte del Nodo 3

La figura 35, muestra la interrupción generada por el módulo IP_Act del nodo 3 que lanza el temporizador T_{DIC} . Antes de que se cumpla dicho tiempo, el nodo 3 recibe el mensaje de postulación del nodo 1. Como respuesta, el nodo 3 envía un mensaje anunciando que adopta al nodo 1 como su enlace en el *cluster*, declinando realizar su postulación a CH. En la figura pueden verse las interrupciones producidas por el módulo UART del nodo cada vez que ingresa un byte enviado por el nodo 1 (IntRecepción). Los mensajes intercambiados son transmitidos por los módulos en su formato API (*Application Programming Interface*) [21], y se muestran en las líneas RX y TX de la figura 35.

La figura 36 muestra otra situación en el nodo 3. La interrupción del módulo IP_Act despierta el nodo, que en este caso no recibe ningún mensaje de su vecino. Por lo tanto, una vez cumplidos los tiempos, se produce la interrupción generada por el temporizador de postulación (IntPostulación) y el nodo 3 envía (TX) un mensaje con el cual se propone como cabecera de *cluster*.

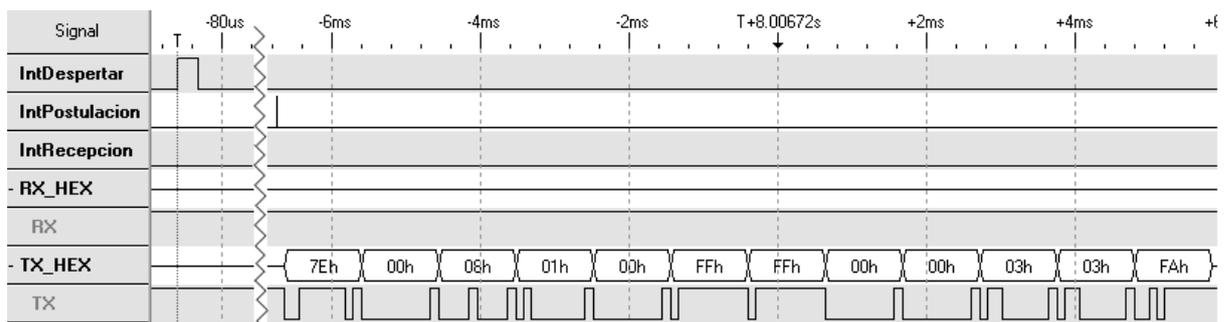


Figura 36. Visualización de señales: Postulación del Nodo 3

Una verificación más detallada de la definición de *clusters* y las pruebas asociadas con el establecimiento del árbol de cabeceras y las fases de envío de datos, exigen conformar una red con mayor número de dispositivos sensores. Sin embargo, se consideró demostrada la viabilidad de la implementación de la arquitectura propuesta en la sección 6.2 a partir de las pruebas descritas. Esta afirmación se fundamenta en que los módulos definidos para los ensayos plasman las ideas básicas de funcionamiento de CLUDITEM, y en que para abarcar su funcionalidad completa se deben incorporar otros bloques del mismo tipo que los ya implementados.

Las pruebas realizadas se enfocaron en el buen funcionamiento del algoritmo en las situaciones planteadas, sin establecer resultados respecto del consumo de energía de los nodos. Como ya se ha comentado, la plataforma reconfigurable utilizada no constituye una implementación definitiva y además, está basada en dispositivos FPGA, muy adecuados para los objetivos planteados, pero que la hacen poco propensa para un estudio del consumo asociado con su funcionalidad.

Finalmente es importante destacar, que la plataforma de prueba demostró ser sumamente flexible para realizar los cambios y correcciones que se llevaron adelante al ensayar el

comportamiento de los nodos. Sus características facilitaron la evaluación de las alternativas descritas y permitieron elegir la más conveniente en base a los resultados obtenidos.

Conclusiones

El análisis y desarrollo de las redes inalámbricas de sensores suele abordarse desde la perspectiva de las redes de comunicaciones y los protocolos utilizados, áreas que han realizado un importantísimo aporte a esta tecnología. Sin embargo, la implementación de una RISI plantea muchos desafíos ya que el logro de un diseño exitoso implica balancear un conjunto de objetivos, muchas veces conflictivos entre sí, abordando el desarrollo y/o adaptación de módulos *software* y de la plataforma *hardware* asociada.

Los profesionales deben trabajar no sólo con los requerimientos funcionales de las aplicaciones, sino también con restricciones muy estrictas originadas por las limitaciones en recursos de los dispositivos sensores, y por las características de los ambientes de trabajo, en ocasiones inhóspitos y muchas veces impredecibles. Por estos motivos, la Ingeniería del Software puede realizar un aporte significativo, apoyando la mejora de la metodología de trabajo y proveyendo soporte para todo el ciclo de vida de implementaciones capaces de alcanzar un buen desempeño en los ambientes en los que se las despliega. El desarrollo de CLUDITEM adoptó este último enfoque, evitando la utilización de especificaciones demasiado genéricas al establecer como punto de partida un análisis detallado del espacio de diseño de las aplicaciones de interés. Los resultados obtenidos con esta perspectiva permitieron definir lineamientos de trabajo fundamentados en requerimientos concretos, que guiaron las decisiones referidas a control de topología, régimen de trabajo, consumo de energía y calidad de servicio.

Las decisiones tomadas sustentaron la elección de estrategias orientadas a plasmar las características deseadas y la definición de métricas para evaluar el desempeño del algoritmo. Además se incorporaron parámetros que dotaron a CLUDITEM de un margen de flexibilidad que le permite adecuarse a variaciones acotadas en los requerimientos de las aplicaciones seleccionadas.

Los resultados de la simulación del funcionamiento del algoritmo se analizaron sobre la base de las métricas definidas, y permitieron realizar ajustes y correcciones. En este sentido se verificó que las estrategias implementadas resultaron eficaces para alcanzar los objetivos

de gestionar la carga de trabajo de los *clusters*, distribuir los CH en el área supervisada, y cumplir con la calidad de servicio durante un período compatible con los requerimientos de las aplicaciones. El mecanismo de rotación de roles realizado periódicamente en base al cálculo del momento óptimo de aplicación, permitió reducir el consumo de energía global en la red balanceando los diferentes períodos de encendido de los dispositivos sensores, acordes con su rol, en distintas rondas de medición.

La incorporación de esquemas TDMA para organizar el envío de mensajes fue necesaria para reducir las colisiones y acotar la pérdida de información. Las pruebas de simulación resultaron fundamentales para definir las características de los esquemas utilizados, que demostraron ser eficaces para garantizar que la red cumpla con la calidad de servicio requerida.

Para verificar la viabilidad funcional de CLUDITEM se realizó una implementación del algoritmo sobre una plataforma de hardware reconfigurable utilizando el procesador embebido Microblaze de Xilinx, sobre la placa de desarrollo Spartan3 Starter Kit Board. La construcción de dos nodos con el algoritmo CLUDITEM permitió verificar la correcta funcionalidad del mismo. La plataforma utilizada demostró ser lo suficientemente flexible como para permitir las pruebas relacionadas con distintas implementaciones de la arquitectura propuesta, y seleccionar la más adecuada.

Finalmente se puede afirmar que la metodología aplicada y las pruebas de funcionamiento llevadas a cabo, permitieron lograr una versión refinada de CLUDITEM capaz de trabajar en escenarios compatibles con las aplicaciones de interés, alcanzando un desempeño coherente con los requerimientos definidos.

Trabajo futuro

Los resultados obtenidos en las pruebas de funcionamiento de CLUDITEM demuestran que el algoritmo es capaz de cumplir con las especificaciones establecidas. Sin embargo se pueden plantear varias líneas de trabajo futuro con el objetivo de mejorar su desempeño, y ampliar su campo de aplicación.

La sincronización de los relojes de los nodos quedó fuera del alcance de esta tesis, pero es importante para poder garantizar un desfase máximo entre los dispositivos y reducir los períodos de actividad. Por lo tanto queda pendiente la implementación de un esquema de sincronización que se elegirá a partir de un cuidadoso análisis de las distintas opciones, ya que su incorporación traerá aparejado un costo en complejidad de funcionamiento y consumo de energía.

Las variables a medir están fuertemente correlacionadas en tiempo y espacio, por lo que sería interesante trabajar con especialistas del dominio para elaborar funciones de agregación más económicas que la utilizada en las pruebas realizadas. El trabajo podría estar orientado no sólo a reducir el tamaño del agregado, sino incluso a decidir sobre el envío de las mediciones dependiendo de cuánto se diferencien respecto de los valores colectados con anterioridad. La reducción de transmisiones es importante para disminuir el consumo, pero también para acotar la probabilidad de colisiones.

La posibilidad de falla en los dispositivos puede afectar seriamente el funcionamiento de la red. Si esta situación se presenta en una cabecera de *cluster*, todas las mediciones del grupo de nodos que coordina se perderán hasta la siguiente reconfiguración, y considerando que el tiempo de vida del sistema se fija en base a la QoS, la red dejará de ser útil para los usuarios en forma prematura. Esta situación puede corregirse decidiendo realizar una reconfiguración del encaminamiento cuando se detecta el problema. La redefinición del árbol de encaminamiento tiene su costo ya que para realizarla los nodos deben permanecer activos más tiempo. Sin embargo, no es el gasto adicional el aspecto más importante, ya que la reconfiguración periódica cada X rondas es la estrategia elegida para balancear el consumo de los nodos, por lo que realizar rotación de roles en otro momento degradará esta

importante característica. Queda entonces planteado realizar un análisis futuro, respecto de la conveniencia de tomar acciones como la descrita en base a la probabilidad de fallas, y también la evaluación de su impacto en la distribución de la carga de trabajo en la red.

Por otro lado, se plantea analizar la posibilidad de incorporar la detección de eventos específicos en el área de trabajo. CLUDITEM está orientado hacia aplicaciones de supervisión ambiental con medición periódica de las variables de interés. Por lo tanto, incrementar su funcionalidad transformándolo en un algoritmo híbrido, no es una tarea simple. La detección de situaciones problemáticas en el ambiente requiere en la mayoría de los casos velocidad de respuesta, y por lo tanto, no sería suficiente despertar los dispositivos cada vez que transcurre un período T del orden de 10 o 15 minutos, como el utilizado para las aplicaciones seleccionadas. Se deberá analizar qué dispositivos se despertarán más a menudo, con qué régimen de trabajo lo harán, y que estructura de encaminamiento resulta adecuada para llevar rápidamente hasta la estación base la información relacionada con la presencia de eventos de importancia en el ambiente.

El trabajo realizado sobre la plataforma HW también abre varias posibilidades futuras, que se relacionan con los ítems ligados a la mejora del desempeño y ampliación de funcionalidad de CLUDITEM antes mencionados. Para comenzar, sería muy interesante poder disponer de recursos para realizar pruebas del funcionamiento completo del algoritmo a partir de un mayor número de nodos sensores. La incorporación de algún esquema de sincronización de relojes implicará, además del estudio de su costo en consumo y complejidad, analizar las distintas posibilidades de su implementación en la plataforma HW. Por otro lado, el trabajo sobre agregación puede conducir a implementar un pre-procesamiento de las mediciones en cada ronda para decidir sobre el envío de la información hacia la estación base. Esta tarea puede ser asumida por el procesador, o por un bloque de HW cableado que evite, de ser posible, el encendido del módulo SW.

Finalmente, sería de interés el estudio respecto de una plataforma HW definitiva, que implemente en su versión final los bloques de la arquitectura, con el objetivo de desplegar una red CLUDITEM en un escenario real. Esto implica un arduo trabajo de ingeniería ligado al diseño de *hardware* bajo fuertes restricciones de consumo y con todos los problemas y desafíos inherentes al mundo físico.

Publicaciones relacionadas con la tesis

Los resultados obtenidos en las distintas etapas del desarrollo de la tesis, se presentaron en congresos nacionales e internacionales. En este sentido se pueden enumerar una serie de publicaciones, correspondientes a los anales de dichos eventos, realizadas en colaboración con los integrantes del Grupo de Redes Inalámbricas de Sensores Inteligentes del cual la autora participa desde 2006.

- CORTI R., D'AGOSTINO E., GIANDOMÉNICO E., MARTÍNEZ R. Algoritmo de encaminamiento para redes de sensores inalámbricas utilizando técnicas de agregación. En: CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN (XIII, 2007, Corrientes, Argentina). CACIC 2007. XIII Congreso Argentino de Ciencias de la Computación. Libro de trabajos. 2007. pp. 1-11.
- CORTI R., MARTÍNEZ R., D'AGOSTINO E., GIANDOMÉNICO E. Redes inalámbricas de sensores inteligentes: Implementación con FPGA de un algoritmo de establecimiento del árbol de encaminamiento. In: SOUTHERN CONFERENCE ON PROGRAMMABLE LOGIC (III, 2008, Bariloche, Argentina). SPL 2008. The III Southern Conference on Programmable Logic. Proceedings. 2008. pp. 13-18.
- CORTI R., D'AGOSTINO E., GIANDOMÉNICO E., BELMONTE J., MARTÍNEZ R. Clustering dinámico para tiempo de encendido mínimo en redes inalámbricas de sensores (CLUDITEM). Análisis de la definición de clusters. En: CONFERENCIA LATINOAMERICANA DE INFORMÁTICA (34, 2008, Santa Fé, Argentina). CLEI 2008. 34° Conferencia Latinoamericana de Informática. Libro de trabajos. 2008. pp. 1219-1228.
- D'AGOSTINO E., CORTI R., GIANDOMÉNICO E., BELMONTE J., MARTÍNEZ R. Clustering dinámico para tiempo de encendido mínimo en redes inalámbricas de sensores (CLUDITEM). Definición del árbol de encaminamiento. En: CONGRESO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN (XIV, 2008, Chilecito, Argentina). CACIC 2008. El XIV Congreso Argentino de Ciencias de la Computación. Libro de trabajos. 2008.

-
- CORTI R., MARTÍNEZ R., D'AGOSTINO E., BELMONTE J., GIANDOMÉNICO E. Arquitectura de un nodo con tiempo de encendido mínimo para una red de sensores inalámbrica. Prototipo basado en FPGA. En: JORNADAS DE COMPUTACIÓN RECONFIGURABLE Y APLICACIONES (IX, 2009, Alcalá de Henares, España). JCRA 2009. IX Jornadas de Computación Reconfigurable y Aplicaciones. Libro de Trabajos. 2009. pp. 203-212.
 - CORTI R., D'AGOSTINO E., GIANDOMÉNICO E., MARTÍNEZ R., BELMONTE J. Clustering Dinámico para Tiempo de Encendido Mínimo en Redes Inalámbricas de Sensores (CLUDITEM). Análisis de las fases de envío de datos. En: Conferencia Latinoamericana de Informática (36, 2010, Asunción, Paraguay). CLEI 2010. 36° Conferencia Latinoamericana de Informática. Libro de Resúmenes. 2010. pp. 49-50. Trabajo completo en el CD del evento.

Nomenclatura

Término	Significado
agregado	Mensaje que representa las mediciones de un conjunto de nodos
agregación	Proceso que permite al nodo responsable obtener el agregado
beacon	Baliza o mensaje que envía un coordinador para sincronizar sus hijos
beaconless	Comunicación sin habilitar beacons o balizas
CH	Nodo cabecera o coordinador de cluster
chip	Circuito integrado en un soporte de silicio
cluster	Conjunto de nodos con un coordinador o cabecera
clustering	Proceso de definición de los clusters de una red
comunicación multisalto	Nodos intermedios retransmiten el mensaje para que llegue a destino
CPLD	Dispositivo complejo de lógica programable
CSMA/CA	Acceso múltiple por detección de portadora c/prevención de colisiones
GPIO	Módulo de entrada / salida de propósitos generales
encaminamiento	Proceso de definición de rutas para llevar los datos hasta el sink
estándar RS-232	Interfaz que soporta una norma p/ intercambio serie de datos binarios
intracluster	Entre nodos miembros de un cluster
intercluster	Entre nodos cabecera de cluster
Esquema TDMA	Acceso al canal utilizando un esquema basado en slots de tiempo
formato API	Estructura del frame de datos de la UART al trabajar en modo API
FPGA	Arreglo de puertas programable en campo
HW	Hardware
ID	Identificador de un nodo
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
IEEE 802.15.4	Estándar que define las capas física y MAC para LR-WPAN
IP	Propiedad intelectual

Término	Significado
LR-WPAN	Red inalámbrica de área personal con baja tasa de transmisión de datos
MAC	Capa de acceso al medio
Mensaje EC	Mensaje necesario para definir la estructura de clusters de la red
Mensaje ACH	Mensaje intercambiado entre los CH para definir el árbol de cabeceras
modo API	Comunicación que simplifica la operatoria con módulos remotos
NC	Nodo común o miembro de un cluster
NDE	Nodo de enlace en la estructura de encaminamiento
OTcl	Lenguaje de herramientas de comando orientado a objetos
QoS	Calidad de servicio
RAM	Memoria de acceso aleatorio
Red inalámbrica Ad-Hoc	Red sin infraestructura predefinida c/ nodos que se mueven libremente
Red plana	Sus nodos tienen asociada la misma funcionalidad
Red jerárquica	Sus nodos asumen roles diferenciados
RISC	Computadora con un conjunto reducido de instrucciones
RISI	Red inalámbrica de sensores inteligentes
script	Archivo de órdenes
sink	Estación base de la red a la que debe llegar la información colectada
slot	Intervalo de tiempo para transmisión de mensajes
SW	Software
topología de una red	Patrón de conexión entre los nodos
transceptor	Dispositivo que se ocupa de la transmisión y recepción de los datos
UART	Unidad asincrónica universal para transmisión y recepción
VHDL	Lenguaje de descripción de HW adoptado como estándar por IEEE

Bibliografía

- [1]. ABBASI A. and YOUNIS M. A survey on clustering algorithms for wireless sensor networks. *Computer Communications (Elsevier)*. 30 (2007): 2826-2941, June 2007.
- [2]. ABOELAZE M. and ALOUL F. Current and Future Trends in Sensor Networks: A Survey. In: IFIP INTERNATIONAL CONFERENCE ON WIRELESS AND OPTICAL COMMUNICATIONS NETWORKS (2, 2005, Missouri, USA). WOCN2005. 2th IFIP International Conference on Wireless and Optical Communications Networks. Proceedings. IEEE Press, 2005, pp.551-555.
- [3]. AL-HADITHI B.M. y SUARDIAZ MURO J. Nuevas tendencias en el diseño electrónico digital: codiseño Hardware/Software. *Tecnología y Desarrollo* [en línea]. Septiembre 2004, II: 3-30 [consulta: 14 de abril de 2011]. Disponible en: http://www.uax.es/publicaciones/archivos/TECELS04_001.pdf.
- [4]. AL-KARAKI J. N. and KAMAL A. E. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*. 11(6): 6-28, December 2004.
- [5]. AL-KARAKI J. N., UI-MUSTAFA R., KAMAL A. E. Data Aggregation in Wireless Sensor Networks - Exact and Approximate Algorithms. In: WORKSHOP ON HIGH PERFORMANCE SWITCHING AND ROUTING (2004, Phoenix, Arizona). HPSR2004. 2004 Workshop on High Performance Switching and Routing. Proceedings. IEEE Press, 2004, pp. 241-245.
- [6]. ALTERA CORPORATION. *Nios II Processor Reference Handbook* [en línea]. San José, USA [consulta: 20 de abril de 2011]. Disponible en http://www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf
- [7]. BORGES, L.M., VELEZ, F.J., LEBRES A.S. Taxonomy for Wireless Sensor Networks Services Characterisation and Classification. In: CONFERENCE ON TELECOMMUNICATIONS (7, 2009, Santa Maria da Feira, Portugal). Conftele' 2009. 7th Conference on Telecommunications. Proceedings. IEEE, 2009, 1: pp. 433-436.
- [8]. BOUKERCHE A., CHENG X., LINUS J. A Performance of a Novel Energy-Aware Data-Centric Routing Algorithm in Wireless Sensor Networks. *Wireless Networks (Springer)*. 11(5): 619-635, September 2005.
- [9]. BURATTI, C., CONTI, A., DARDARI, D., VERDONE, R. An Overview on Wireless Sensor Networks Technology and Evolution. *Sensors* [en línea]. August 2009, 9 (9): 6869-6896 [consulta: 14 de abril de 2011]. Disponible en: <http://www.mdpi.com>.

- [10]. CHALIVENDRA G., SNIRIVASAN R., MURTHY N.S. FPGA Based Re-Configurable Wireless Sensor Network Protocol. *In: INTERNATIONAL CONFERENCE ON ELECTRONIC DESIGN (2008, Penang, Malaysia). 2008 International Conference on Electronic Design. Proceedings. IEEE Press, 2008, pp 1-4.*
- [11]. CHAPMAN K. *200 MHz UART with Internal 16-Byte Buffer* [en línea]. Xilinx Inc. USA, 2008 [consulta: 20 de abril de 2011]. Disponible en: http://www.xilinx.com/support/documentation/application_notes/xapp223.pdf.
- [12]. CHEN H. and MEGERIAN S. Cluster Sizing and Head Selection for Efficient Data Aggregation and Routing in Sensor Networks. *In: WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE (2006, Las Vegas, USA). WCNC 2006. 2006 Wireless Communications and Networking Conference. Proceedings. IEEE Press, 2006, 4: pp 2318-2323.*
- [13]. COMMURI S., TADIGOTLA V., ATIQUZZAMAN M. Reconfigurable Hardware Based Dynamic Data Aggregation in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks (ACM)*. 4 (2): 194-212, April 2008.
- [14]. CROSSBOW TECHNOLOGY INC. *Home Page* [en línea]. [consulta: 20 de abril de 2011]. Disponible en: <http://www.xbow.com>.
- [15]. CULLER D., ESTRIN D., SRIVASTAVA M. Overview of Sensor Networks. *IEEE Computer Magazine*. 37 (8): 41-49, August 2004.
- [16]. CZAPSKI P.P. and SLUZEC A. A Survey on System-Level Techniques for Power Reduction in Field Programmable Gate Array (FPGA)-Based Devices. *In: INTERNATIONAL CONFERENCE ON SENSOR TECHNOLOGIES AND APPLICATIONS (2, 2008, Cap Esterel, France). SENSORCOMM '08. The Second International Conference on Sensor Technologies and Applications. Proceedings. IEEE Press, 2008, pp 319-328.*
- [17]. DEBONO C.J. and BORG N.P. The Implementation of an Adaptive Data Reduction Technique for Wireless Sensor Networks. *In: IEEE INTERNATIONAL SYMPOSIUM ON SIGNAL PROCESSING AND INFORMATION TECHNOLOGY (2008, Sarajevo, Bosnia y Herxegovina). IEEE International Symposium on Signal Processing and Information Technology. Proceedings. IEEE Press, 2008, pp. 402-406.*
- [18]. DECHENE D.J., EL JARDALI A., LUCCINI M., SAUER A. A Survey of Clustering Algorithms for Wireless Sensor Networks [en línea]. Project Report. Department of Electrical and Computer Engineering, University of Western Ontario, Canadá, December 2006. [consulta: 21 de abril de 2011]. Disponible en: http://www.dechene.ca/papers/report_635a.pdf.

- [19]. DELIANG, L. and FEI P. Energy-efficient MAC protocols for Wireless Sensor Networks. *Information and Communications Technologies*. Beihang University. Beijing. 2009 (3): 32-36, March 2009.
- [20]. DEMIRKOL, I., ERSOY, C., ALAGOZ, F. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*. 44 (4): 115-121, May 2006.
- [21]. DIGI INTERNATIONAL INC. *XBee/XBee-PRO OEM RF Modules Product Manual v1.xCx - 802.15.4 Protocol* [en línea]. Minnetonka, 2008 [consulta: 10 de abril de 2011]. Disponible en:
http://ftp1.digi.com/support/documentation/90000982_A.pdf.
- [22]. DUNLAP R. In-Network Aggregation in Wireless Sensor Networks. College of computing [en línea]. Georgia Institute of Technology. USA, 2004. [consulta: 21 de abril de 2011]. Disponible en:
http://www.cc.gatech.edu/~rocky/docs/rocky_netagg.pdf.
- [23]. FAN K.W., LIU S., SINHA P. Structure-Free Data Aggregation in Sensor Networks. *IEEE Transactions on Mobile Computing*. 6 (8): 923-942, August 2007.
- [24]. FASOLO E., ROSSI M., WIDMER J., ZORZI M. In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey. *IEEE Wireless Communication*. 14 (2):70-87, May 2007.
- [25]. FLETCHER B. FPGA Embedded Processors: Revealing True System Performance. In: EMBEDDED SYSTEMS CONFERENCE (2005, San Francisco, USA). The 2005 Embedded Systems Conference. Proceedings. 2005, pp. 2-18.
- [26]. GARCÍA-HERNÁNDEZ C. F., IBARGÜENGOYTIA-GONZÁLEZ P. H., GARCÍA-HERNÁNDEZ J., PÉREZ-DÍAZ J.A. Wireless Sensor Networks and Applications: a Survey. *International Journal of Computer Science and Network Security IJCSNS*. 7 (3): 264-273, March 2007.
- [27]. GARCIA R., GORDON-ROSS A., GEORGE A. D. Exploiting Partially Reconfigurable FPGAs for Situation-Based Reconfiguration in Wireless Sensor Networks. In: IEEE SYMPOSIUM ON FIELD PROGRAMMABLE CUSTOM COMPUTING MACHINES (17, 2009, Napa, USA). The 17th IEEE Symposium on Field Programmable Custom Computing Machines. Proceedings. IEEE Press, 2009, pp.243-246.
- [28]. GUCCIONE S. Microprocessors: The New LUT. In: INTERNATIONAL CONFERENCE ON ENGINEERING OF RECONFIGURABLE SYSTEMS AND ALGORITHMS (2005, Las Vegas, Nevada). ERSA 2005. The 2005 International Conference on Engineering of Reconfigurable Systems and Algorithms. Proceedings. CSREA Press, pp.s.p.
- [29]. HAIGANG G., MING L., XIAOMIN W., LIJUN C., LI X. An Interference Free Cluster-Based TDMA Protocol for Wireless Sensor Networks. In: Wireless

- Algorithms, Systems, and Applications. Lecture Notes in Computer Science, CHENG X., LI W., ZNATI T. (Ed.) Vol. 4138/2006, Springer Berlin, 2006, pp. 217-227.
- [30]. HAKALA I., TIKKAKOSKI M., KIVELÄ I. Wireless Sensor Network in Environmental Monitoring - Case Foxhouse. In: INTERNATIONAL CONFERENCE ON SENSOR TECHNOLOGIES AND APPLICATIONS (2, 2008, Cap Esterel, France). SENSORCOMM '08. The Second International Conference on Sensor Technologies and Applications. Proceedings. IEEE Press, 2008, pp. 202-208.
- [31]. HAQUE R. and NAZNIN M. Monitoring Cost Reduction in Sensor Networks using Proximity Queries. *Journal of Networks (Academy Publisher)*. 6 (1): 4-11, January 2011.
- [32]. HE T., BLUM B.M., STANKOVIC J.A., ABDELZAHER T. AIDA: Adaptive Application-Independent Data Aggregation in Wireless Sensor Networks. *ACM Transactions on Embedded Computing Systems*. 3 (2): 426-457, May 2004.
- [33]. HEINZELMAN, W. B. Application-specific Protocol Architectures for Wireless Networks. PhD Thesis (Doctor of Philosophy). Massachusetts, USA. Massachusetts Institute of Technology, 2000. 154 p.
- [34]. HEMPSTEAD M., LYONS M.J., BROOKS D., WEI G. Survey of Hardware Systems for Wireless Sensor Networks. *Journal of Low Power Electronics (American Scientific Publishers)*. 4 (1): 1-10, April 2008.
- [35]. HILL, J.L. System Architecture for Wireless Sensor Networks. PhD Thesis (Doctor of Philosophy in Computer Science). Berkeley, USA. University of California at Berkeley, 2003. 186 p.
- [36]. HINKELMANN H., REINHARDT A., VARYANI S., GLESNER M. A Reconfigurable Prototyping Platform for Smart Sensor Networks. In: SOUTHERN CONFERENCE ON PROGRAMMABLE LOGIC (4. 2008, San Carlos de Bariloche, Argentina). SPL 2008. The 4th Southern Conference on Programmable Logic. Proceedings. IEEE, pp. 125-130.
- [37]. IBRIQ J. and MAGOUB I. Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges. In: INTERNATIONAL SYMPOSIUM ON PERFORMANCE EVALUATION OF COMPUTER AND TELECOMMUNICATION SYSTEMS (2004, San Diego, California). SPECTS'04. International Symposium on Performance Evaluation of Computer and Telecommunication Systems. Proceedings. The Society for Modeling and Simulation International, 2004, pp. 759-766.
- [38]. IEEE COMPUTER SOCIETY. IEEE *Std 802.15.4™-2006* [en línea]. IEEE, New York, USA, 2006 [consulta: 20 de abril de 2011]. Disponible en: <http://standards.ieee.org/about/get/802/802.15.html>.

- [39]. INTANAGONWIWAT C., GOVINDAN R., ESTRIN D., HEIDEMANN J., SILVA F. Directed Diffusion for Wireless Sensor Networking. *IEEE / ACM Transactions on Networking*. 11 (1): 2-16, February 2003.
- [40]. INTRONIX. *34 Channel LA1034 Logicport Logic Analyzer-Home Page* [en línea]. [consulta: 20 de abril de 2011]. Disponible en: <http://www.pctestinstruments.com/index.htm>.
- [41]. ISRAR N. and AWAN I. Multihop Clustering Algorithm for Load Balancing In Wireless Sensor Networks. *International Journal of Simulation, Systems, Science and Technology*. 8 (1): 13-25, September 2007.
- [42]. KARL H. and WILLIG A. Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, 2005. 497 p.
- [43]. KENT K.B., SERRA M., HORSPOOL N. Hardware/software co-design for virtual machines. *Computers and Digital Techniques*. IEEE. 152 (5): 537 - 548, September 2005.
- [44]. KOTONYA G. and SOMMERVILLE I. Requirements Engineering: Processes and Techniques. New York, John Wiley and Sons, 2001. 282 p.
- [45]. LINDSEY S. and RAGHAVENDRA C.S. PEGASIS: Power-Efficient Gathering in Sensor Information Systems. *In: IEEE AEROSPACE CONFERENCE (2002, Big Sky, Montana, USA)*. IEEE Aerospace Conference. Proceedings. IEEE Press, 2002, 3: pp. 1125-1130.
- [46]. LI M., LIU Y., CHEN L. Nonthreshold-Based Event Detection for 3D Environment Monitoring in Sensor Networks. *IEEE Transactions on Knowledge and Data Engineering*. 20 (12): 1699-1711, 2008.
- [47]. LIU H., MENG Z., CUI S. A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses. *In: IEEE INTERNATIONAL CONFERENCE OF WIRELESS COMMUNICATIONS, NETWORKING AND MOBILE COMPUTING (2007, Shanghai, China)*. WiCom 2007. The 2007 IEEE International Conference of Wireless Communications, Networking and Mobile Computing. Proceedings. IEEE Press, 2007, pp. 2334-2347.
- [48]. MAC RUAIRÍ R., KEANE M.T., COLEMAN G. A Wireless Sensor Network Application Requirements Taxonomy. *In: IEEE INTERNATIONAL CONFERENCE ON SENSOR TECHNOLOGIES AND (2, 2008, Cap Esterel, France)*. SENSORCOMM '08. The Second International Conference on Sensor Technologies and Applications. Proceedings. IEEE Press, 2008, pp. 209-216.
- [49]. MADDEN S., FRANKLIN M.J., HELLERSTEIN J.L., HONG W. TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. *In: SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (5, 2002, Boston,*

- USA). The 5th Symposium on Operating Systems Design and Implementation. Proceedings. USENIX Association, 2002, pp. 131-146.
- [50]. MANJHI A., NATH S., GIBBONS P.B. Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Streams. *In: ACM INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA (2005, Baltimore, Maryland, USA). SIGMOD 2005. ACM, 2005, pp. 287-298.*
- [51]. MHATRE V. and ROSENBERG C. Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation. *Ad Hoc Networks Journal Elsevier Science. 2 (1): 45- 63, 2004.*
- [52]. NACHMAN L., KLING R., ADLER R., HUANG J., HUMMEL V. The Intel Mote Platform: a bluetooth-based sensor network for industrial monitoring. *In: INTERNATIONAL SYMPOSIUM ON INFORMATION PROCESSING IN SENSOR NETWORKS (4, 2005, Piscataway, USA). The 4th International Symposium on Information Processing in Sensor Networks. Proceedings. IEEE Press, 2005, pp. 437-442.*
- [53]. NATH S., GIBBONS P.B., SESHAN S., ANDERSON Z.R. Synopsis Diffusion for Robust Aggregation in Sensor Networks. *In: ACM CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS (2, 2004, Baltimore, Maryland, USA). SenSys 2004. The Second ACM Conference on Embedded Networked Sensor Systems. Proceedings. ACM, 2005, pp. 250-262.*
- [54]. *The Network Simulator - ns-2* [en línea]. [consulta: 20 de abril de 2011]. Disponible en: <http://www.isi.edu/nsnam/ns/>.
- [55]. OLUKOTUN K. and HAMMOND L. The future of microprocessors. *ACM Queue Magazine. 3 (7): 26-34, 2005.*
- [56]. OU J. and PRASANNA V. K. Rapid Energy Estimation for Hardware-software Codesign Using FPGAs. *EURASIP Journal on Embedded Systems. 2006: 1-11, 2006.*
- [57]. PANDEY V., KAUR A., CHAND N. A review on data aggregation techniques in wireless sensor network. *Journal of Electronic and Electrical Engineering. Bioinfo Publications. 1 (2): 1-8, 2010*
- [58]. PAWLOWSKI A., GUZMAN J.L., RODRÍGUEZ F., BERENGUEL M., SÁNCHEZ J., DORMIDO S. Simulation of Greenhouse Climate Monitoring and Control with Wireless Sensor Network and Event-Based Control. *Sensors 2009* [en línea]. January 2009, 9 (1): 232-252 [consulta: 15 abril de 2011]. Disponible en: <http://www.mdpi.com>.
- [59]. PEI Z., DENG Z., YANG B., CHENG X. Application-Oriented Wireless Sensor Network Communication Protocols and Hardware Platforms: a Survey. *In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY (2008,*

- Chengdu, China). The 2008 IEEE International Conference on Industrial Technology. Proceedings. IEEE Press, 2008, pp. 1-6.
- [60]. PORTILLA BERRUECO J. Plataforma modular e interfaces genéricas de transductores para redes de sensores inalámbricas. Tesis Doctoral. Madrid, España. Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, 2010. 224 p.
- [61]. RAHIMI M., BAER R., IROEZI O.I., GARCIA J.C., WARRIOR J., ESTRIN D., SRIVASTAVA M. Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks. *In: INTERNATIONAL CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS (3, 2005, New York, USA). The 3rd International conference on Embedded networked sensor systems. ACM, 2005, pp. 192-204.*
- [62]. RAMACHANDRAN I., DAS A.K., ROY S. Analysis of the contention access period of IEEE 802.15.4 MAC. *ACM Transactions on Sensor Networks (TOSN)*. 3 (1): 1-29, March 2007.
- [63]. RÖMER K. and MATTERN F. The Design Space of Wireless Sensor Networks. *IEEE Wireless Communications*. 11 (6): 54-61, December 2004.
- [64]. SHRIVASTAVA N., BURAGOHAJN C., AGRAWAL D., Medians and Beyond: New aggregation Techniques for Sensor Networks. *In: ACM CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS (2, 2004, Baltimore, Maryland, USA). SenSys 2004. The Second ACM Conference on Embedded Networked Sensor Systems. Proceedings. ACM, 2004, pp.11.*
- [65]. SINGH S.K., SINGH M.P., SINGH D.K., Routing Protocols in Wireless Networks – A Survey. *International Journal of Computer Science & Engineering Survey (IJCSES)*. 1 (2): 63-83, November 2010.
- [66]. SOLIS I. and OBRACZKA K. In-network aggregation trade-offs for data collection in wireless sensor networks. *ACM International Journal of Sensor Networks 2006*. 1 (3/4): 200 - 212, September 2006.
- [67]. SUN M., SUN K., ZOU Y. Analysis and Improvement for 802.15.4 Multi-hop Network. *In: INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND MOBILE COMPUTING (2009, Kunming, Yunnan, China). CMC 2009. The 2009 International Conference on Communications and Mobile Computing. Proceedings. IEEE Computer Society, 2009, pp. 52-56.*
- [68]. TEXAS INSTRUMENTS INC. *CC2420 Data Sheet* [en línea]. Dallas, Texas, 2008 [consulta: 1 de abril de 2011]. Disponible en: <http://www-mtl.mit.edu/Courses/6.111/labkit/datasheets/CC2420.pdf>.

- [69]. TILAK S., ABU-GHAZALEH N. B., HEINZELMAN W. A Taxonomy of Wireless Micro-Sensor Networks Models. *ACM Mobile Computing and Communications Review*. 6 (2): 1-8, April 2002.
- [70]. VIJ V. and MEHRA R. FPGA Based Kalman Filter for Wireless Sensor Networks. *International Journal of Computer Technology and Applications* [en línea]. January/February 2011, 2 (1): 155-159 [consulta: 15 de abril de 2011]. Disponible en: <http://ijcta.com/>.
- [71]. THE VINT PROJECT. *The ns Manual* [en línea]. Fall K., Varadhan K. Editores. USA, 2010 [consulta: 20 de abril de 2011]. Disponible en: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf.
- [72]. VLAJIC N. and XIA D. Wireless Sensor Networks: To Cluster or Not To Cluster? In: INTERNATIONAL SYMPOSIUM ON A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS (2006, New York, USA). WoWMoM'06. The 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks. Proceedings. IEEE Computer Society, 2006, pp. 258-268.
- [73]. VÖLGYESI P., SALLAI J., LÉDECZI A., DUTTA P., MARÓTI M. Software development for a novel WSN platform. In: INTERNATIONAL WORKSHOP ON SOFTWARE ENGINEERING FOR SENSOR NETWORK APPLICATIONS (1, 2010, Cape Town, South Africa). SESENA'10. The First International Workshop on Software Engineering for Sensor Network Applications. Proceedings. ACM Press, 2010, pp. 20-25.
- [74]. WEST B., FLIKKEMA P., SISK T., KOCH G. Wireless sensor networks for dense spatio-temporal monitoring of the environment: A case for integrated circuit, system, and network design. In: IEEE CAS WORKSHOP ON WIRELESS COMMUNICATIONS AND NETWORKING (2001, Notre Dame, Indiana). The 2001 IEEE CAS Workshop on Wireless Communications and Networking. Proceedings. IEEE Press, 2001, pp.s.p.
- [75]. WILDER J., UZELAC V., MILENKOVIC A., JOVANOVIĆ E. Runtime Hardware Reconfiguration in Wireless Sensor Networks. In: SOUTHEASTERN SYMPOSIUM ON SYSTEM THEORY (40, 2008, New Orleans, USA). The 40th Southeastern Symposium on System Theory. Proceedings. IEEE Press, 2008, pp. 154-158.
- [76]. WOLF W. A Decade of Hardware/Software Codesign. *IEEE Computer Magazine*. 36 (4): 38-43, 2003.
- [77]. XILINX INC. *ISE WebPack Design Software* [en línea]. [consulta: 2 de abril de 2011]. Disponible en: <http://www.xilinx.com/tools/webpack.htm>.
- [78]. XILINX INC. *CoreConnect Architecture - On-chip Peripheral Bus* [en línea]. [consulta: 1 de abril de 2011]. Disponible en:

- http://www.xilinx.com/ipcenter/processor_central/coreconnect/coreconnect_opb.htm.
- [79]. XILINX INC. *PicoBlaze 8-bit Embedded Microcontroller User Guide* [en línea]. [consulta: 1 de abril de 2011]. Disponible en:
http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf.
- [80]. XILINX INC. *Spartan-3 Starter Kit Board User Guide* [en línea]. [consulta: 1 de abril de 2011]. Disponible en:
<http://www.digilentinc.com/Data/Products/S3BOARD/S3BOARD-rm.pdf>.
- [81]. XILINX INC. *Embedded System Tools Reference Manual* [en línea]. [consulta: 1 de abril de 2011]. Disponible en:
http://www.xilinx.com/support/documentation/sw_manuals/edk82i_est_rm.pdf
- [82]. XILINX INC. *MicroBlaze Processor Reference Guide* [en línea]. [consulta: 1 de abril de 2011]. Disponible en:
http://www.xilinx.com/support/documentation/sw_manuals/edk82i_mb_ref_guide.pdf
- [83]. YAO Y. and GEHRKE J. The Cougar Approach to In-Network Query Processing in Sensor Networks. *ACM SIGMOD Record*. 31 (3): 9-18, September 2002.
- [84]. YICK J., MUKHERJEE B., GHOSAL D. Wireless sensor network survey. *Computer Networks: The International Journal of Computer and Telecommunications Networking* (Elsevier). 52 (12): 2292-2330, August 2008.
- [85]. YOUNIS O., KRUNZ M., RAMASUBRAMANIAN S. Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges. *IEEE Network Magazine*. 20 (3): 20-25, June 2006.
- [86]. YU M., HALA M., MADJID M. A Survey of Network Management Architecture in Wireless Sensor Network. In: ANNUAL POSTGRADUATE SYMPOSIUM ON THE CONVERGENCE OF TELECOMMUNICATIONS, NETWORKING AND BROADCASTING (7, 2006, Liverpool, UK). PGNNet 2006. The 7th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting. Proceedings. EPSRC, 2006, pp. 201-206.
- [87]. ZHANG S.K., GONG S.R., CUI Z.M., LIU Q., FAN J.X. An Aggregation Tree Approach for Event Detection in Wireless Sensor Networks. *Journal of Software* (Academy Publisher). 4 (8): 899-906, October 2009.
- [88]. ZHENG J. and LEE M.J. A comprehensive performance study of IEEE 802.15. 4. In: Sensor Network Operations, IEEE Press, Wiley Interscience, 2006, Chapter 4, pp. 218-237.
- [89]. ZHOU X. and XU J. IISA: An Inter-cluster and Intra-cluster Scheduling Algorithm Cluster-Based for Wireless Sensor Networks. In: INTERNATIONAL

CONFERENCE ON WIRELESS COMMUNICATIONS, NETWORKING AND MOBILE COMPUTING (4, 2008, Dalian). WiCOM'08. The 4th International Conference on Wireless Communications, Networking and Mobile Computing. Proceedings. IEEE Press, 2008, pp. 1- 6.