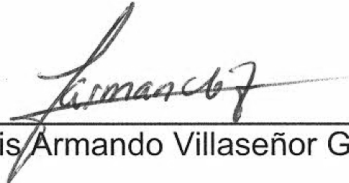


TESIS DEFENDIDA POR  
**Joel Ruiz Ibarra**  
Y APROBADA POR EL SIGUIENTE COMITÉ



---

Dr. Luis Armando Villaseñor González  
*Co-Director del Comité*



---

Dr. José Rosario Gallardo López.  
*Co-Director del Comité*

---

Dr. Jaime Sánchez García.  
*Miembro del Comité*

---

Dr. Carlos Alberto Brizuela Rodríguez.  
*Miembro del Comité*



---

Dr. Javier Gómez Castellanos  
*Miembro Externo del Comité*

---

Dr. Roberto Conte Galván.

*Coordinador del programa de  
posgrado en Electrónica y  
Telecomunicaciones.*

---

Dr. David Hilario Covarrubias  
Rosales

*Director de Estudios de Posgrado*

26 de Agosto del 2011.

**CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR  
DE ENSENADA**



---

**PROGRAMA DE POSGRADO EN CIENCIAS  
EN ELECTRÓNICA Y TELECOMUNICACIONES.**

---

**ENFOQUE MULTICAPA PARA LA DEFINICIÓN DE UN PROTOCOLO DE  
ACCESO AL MEDIO CAPAZ DE OFRECER GARANTÍAS DE CALIDAD DE  
SERVICIO EN REDES INALÁMBRICAS DE SENSORES.**

**TESIS**

que para cubrir parcialmente los requisitos necesarios para obtener el grado de  
**DOCTOR EN CIENCIAS**

Presenta:

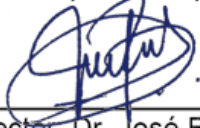
**JOEL RUIZ IBARRA**

Ensenada, Baja California, México, Agosto del 2011.

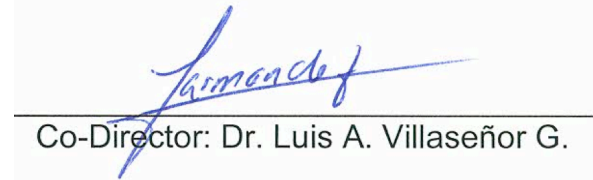
**RESUMEN** de la tesis de Joel Ruiz Ibarra, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS en ELECTRÓNICA Y TELECOMUNICACIONES con orientación en TELECOMUNICACIONES. Ensenada, Baja California. Agosto 2011.

**Enfoque multicapas para la definición de un protocolo de acceso al medio capaz de ofrecer garantías de calidad de servicio en redes inalámbricas de sensores.**

Resumen aprobado por:



Co-Director: Dr. José Rosario. Gallardo L.



Co-Director: Dr. Luis A. Villaseñor G.

Existe una creciente demanda de Redes Inalámbricas de Sensores (WSN) que sean capaces de soportar información en tiempo real. Sin embargo, las tecnologías actuales no son capaces de ofrecer garantías de calidad de servicio (QoS), el cual es necesario para dar servicio a este tipo de aplicaciones. Lograr calidad de servicio en una WSN es un reto, especialmente por su naturaleza multi salto, su poca capacidad de procesamiento, poca memoria y poca disponibilidad de energía. En esta tesis se propone una arquitectura multi-capas en la cual el mecanismo de control de acceso al medio (MAC) y de enrutamiento colaboran para organizar a los nodos en agrupamientos o clusters, y lograr un acceso al medio de transmisión coordinado por compartición de tiempo. El protocolo resultante es llamado QUATTRO: QUALity-of-service-capable clusTer-based Time-shared ROuting-assisted MAC protocol. Los resultados de evaluación del desempeño mostraron que el sobreencabezado del protocolo, mostrado en términos de tiempo de configuración, mensajes de control transmitidos y energía consumida es muy razonable y que no sólo se logra ofrecer garantías de QoS sino además ahorro de energía, eliminando colisiones y reduciendo considerablemente la escucha oscura.

**Palabras Clave:** Protocolos multi-capa, control de acceso al medio, calidad de servicio, enrutamiento, redes inalámbricas de sensores.

**ABSTRACT** of the thesis presented by Joel Ruiz Ibarra as a partial requirement to obtain the DOCTOR OF SCIENCE degree in ELECTRONICS AND TELECOMMUNICATIONS in the area of TELECOMUNICACIONES. Ensenada, Baja California, México. August 2011.

**CROSSLAYER APPROACH FOR THE DEFINITION OF A QUALITY OF SERVICE CAPABLE MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS**

There is increasing demand for wireless sensor networks (WSN) to be able to carry real-time information. However, current WSN technologies are not yet capable of offering quality of service (QoS) guarantees, which are required to support these types of applications. Achieving QoS is especially challenging in WSNs due to their multi-hop nature and their processing-power, memory and energy constraints. In this paper we propose a cross-layer architecture in which the medium-access control (MAC) and routing protocols collaborate to organize nodes into clusters and to achieve a coordinated time-shared access to the transmission medium. The resulting protocol is called QUATTRO: QUAlity-of-service-capable clusTer-based Time-shared ROuting-assisted MAC protocol. Our performance evaluation results show that the protocol overhead observed in terms of configuration time, transmitted control messages and consumed energy is very reasonable and that not only QoS is achieved but also great energy savings by eliminating collisions and considerably reducing idle listening.

**Keywords:** Cross-layer protocols, medium access control, quality of service, routing, wireless sensor networks.

## **Dedicatorias**

Dedico esta tesis a Dios, que “en armonía, bajo su gracia y de manera perfecta”, sea para el beneficio del hombre.

## **Agradecimientos**

Agradezco a mis padres, José Ruiz y Mirna Ibarra, por su ejemplo y constante apoyo en la realización de esta tesis.

Agradezco a mi hermana y mi cuñado, Erica Ruiz y Adolfo Espinoza, por enseñarme el camino del trabajo y la constancia.

Agradezco a José Rosario, por sus enseñanzas que trascienden fuera de los confines de la formación doctoral, como ejemplo de un ser con gran calidad humana, respeto por la vida, la familia y las buenas costumbres. Gracias por la amistad y familia adoptiva que me has brindado.

Agradezco a Betty Olvera, Martha Lucia y Spanish por ser mi familia adoptiva en tierras lejanas y brindarme su amistad incondicional.

Agradezco al Dr. Dimitrios Makrakis y al Dr. Hussein Mouftah por la oportunidad de ser parte de su equipo de trabajo en la universidad de Ottawa.

Agradezco al Dr. Abdallah Shami por permitirme hacer una estancia de investigación en la Universidad del Oeste de Ontario.

Agradezco al Dr. Villaseñor, por su disposición para orientarme en la realización de esta tesis.

Agradezco a los miembros del comité por sus retroalimentaciones e interés por que esta tesis concluya positivamente.

A Aldo, Silvia y Oscar por su amistad y ayuda en la gestión de esta tesis.

A Aurelie, Octavio, Jose Luis, Paul, Benjamín, Mae Abelardo, Crystal, Yuichi, y todos aquellos amigos que hicieron amena esta travesía.

Agradezco al CICESE por brindar los medios físicos y humanos para la realización de esta tesis.

Agradezco al CONACYT por su apoyo a lo largo del programa de doctorado.

## CONTENIDO

	<b>Página</b>
Resumen español .....	i
Resumen inglés.....	ii
Dedicatorias.....	iii
Agradecimientos.....	iv
Contenido.....	v
Lista de Figuras.....	vi
Capítulo I. Introducción.....	1
Capítulo II. Estado del arte.....	7
Capítulo III. Metodología.....	11
Capítulo IV. Simulación y resultados.....	38
IV.1 Modelo de simulación.....	38
IV.2 Evaluación de fase de configuración.....	42
IV.3 Desempeño en transmisión de datos.....	47
IV.4 Escenario “ROW”.....	51
IV.5 Costo de configuración .....	56
IV.6 Costo de las diferentes opciones de reconfiguración.....	57
Capítulo V. Conclusiones.....	63
Bibliografía.....	65

## LISTA DE FIGURAS

<i>Figura</i>		<b>Página</b>
1	Formato de paquete, descubrimiento de rutas.	<b>13</b>
2	Formato de paquete, mensajes de prueba.	<b>13</b>
3	Resumen de enrutamiento y asignación de pesos.	<b>15</b>
4	Formato de paquete, mensajes de reservación-asociación.	<b>16</b>
5	Resumen del proceso de reservación de recursos.	<b>20</b>
6	Formato de paquete, recolección de interferentes.	<b>22</b>
7	Resumen del proceso de recolección de interferentes.	<b>23</b>
8	Estructura de supertrama.	<b>24</b>
9	Estructura de un ciclo de trama.	<b>24</b>
10	Formato de paquete, notificación de ventanas de actividad.	<b>29</b>
11	Resumen de notificación de ventanas de actividad.	<b>29</b>
12	Formato de paquete, reconfiguración.	<b>34</b>
13	WSN con despliegue aleatorio y fase de configuración terminada.	<b>38</b>
14	Modelo de nodo.	<b>41</b>
15	Secuencia de ejecución de pipelines.	<b>41</b>
16	Modelo de proceso. Etapa de configuración.	<b>42</b>
17	Modelo de proceso. Etapa de transmisión de datos.	<b>42</b>
18	Tiempo de configuración, escenario: área variable.	<b>44</b>
19	Número de paquetes de configuración, escenario: área variable.	<b>44</b>
20	Tiempo de configuración, escenario: área fija.	<b>45</b>
21	Número de paquetes de configuración, escenario: área fija.	<b>45</b>
22	Energía consumida durante el proceso de configuración, escenario: área fija.	<b>47</b>
23	Caudal eficaz promedio, escenario: área fija.	<b>49</b>
24	Retraso promedio, escenario: área fija.	<b>49</b>



25	Energía promedio, escenario: área fija.	<b>51</b>
26	Porcentaje de ciclo encendido, escenario: área fija.	<b>51</b>
27	Escenario "ROW".	<b>52</b>
28	Caudal eficaz promedio, escenario: ROW	<b>53</b>
29	Retraso promedio, escenario: ROW.	<b>53</b>
30	Energía promedio, escenario: ROW.	<b>54</b>
31	Porcentaje de ciclo encendido, escenario: ROW.	<b>55</b>
32	Energía por bit, escenario: área fija, 70n/Ac	<b>56</b>
33	Tiempo de recuperación, escenario: área fija 70n/Ac	<b>59</b>
34	Umbral de nodos desconectados para diferentes tipos de recuperación, escenario: área fija.	<b>59</b>
35	Tiempo de recuperación y porcentaje de fallas necesarias para desconectar una red, en función de la densidad de nodos.	<b>61</b>

# Capítulo I

---

## Introducción

---

Una red inalámbrica de sensores (WSN) es una red de área local autoconfigurable (WLAN) de la cual sus terminales están equipadas con algún tipo de sensor y transmiten los parámetros medidos a un conjunto predeterminado de receptáculos de información, llamados "Sink". La principal diferencia entre una WLAN y una WSN es que, en general, esta última requiere protocolos más eficientes en consumo de energía, ya que sus nodos están alimentados generalmente por baterías no recargables. Debido a estas restricciones, los protocolos para WSN deben ser eficientes en energía, tales como enrutamiento dinámico que no agote la energía de un nodo común para diferentes rutas, o protocolos de control de acceso al medio (MAC) que les permitan a los nodos apagarse por cierto tiempo y así ahorrar energía.

Existe actualmente un gran interés, tanto tecnológico como económico, en el desarrollo de soluciones para WSN. Esta necesidad creciente de soluciones, fomenta el usar con mayor eficiencia los recursos de la red, incluyendo protocolos que permitan acceder al medio de comunicaciones de una forma ordenada, así como nuevos métodos de descubrimiento de rutas debido a la naturaleza ad-hoc de este tipo de redes.

Conforme la tecnología de las WSN evoluciona hacia tasas de transmisión más altas, tiene sentido considerar el sensado de información de tiempo real como audio y video. Estas aplicaciones requieren que el sistema sea capaz de ofrecer garantías de calidad de servicio (QoS). Es tal la necesidad de este tipo de

protocolos que se maneja ahora el término Red de Sensores de Video Inalámbrica, por sus siglas en inglés (WVSN). Fallahi, Hossain, (2009); Chow., (2009); Giirses et al (2009) Este es un campo de investigación que ha recibido una gran atención recientemente como se explicará en el Capítulo II (Wang, et al. (2006); Zhao. et al. (2006); Egorova-Forster y Murphy (2007); Small y Haas (2007).

Propuestas recientes para WSN han adoptado con éxito la costumbre de intercambiar información entre capas del modelo tradicional de protocolos, OSI. Este intercambio de información entre capas ha resultado en un mejor desempeño que el diseño por capas, a pesar de restarle modularidad al protocolo. Esta tendencia en el diseño de protocolos ha recibido el nombre de “cross-layering”.

Las WSN son ya una realidad, pero aun se requiere del diseño de protocolos que mejoren su desempeño mediante la optimización del uso de los recursos. Esta propuesta se enfoca principalmente en proveer un protocolo que sea eficiente en energía y capaz de garantizar determinada QoS.

El objetivo de este trabajo es proponer un protocolo “cross-layer” en el cual la capa de enrutamiento y la sub-capas MAC sean capaces de colaborar en conjunto para lograr un mecanismo eficiente en energía y proveer de garantías de QoS para redes de sensores inalámbricas. El protocolo resultante de esta propuesta recibe el nombre de QUATTRO (QUALity of service capable clusTer based Time shared ROuting assisted MAC protocol).

Las tareas realizadas por el protocolo de enrutamiento incluyen descubrimiento de rutas, reservación de recursos, formación de clústeres y recopilación de información como interferencias entre clústeres. El protocolo MAC emplea estos datos y crea un calendario de ventanas de actividad para los clústeres o

agrupamientos, cuidando de asignarlas de modo que se den transmisiones libres de colisiones entre los clústeres.

El mecanismo propuesto ahorra energía gracias a que evita colisiones y permite a los nodos apagar sus radios transmisores de una forma segura, sin el riesgo de perder datos relevantes. Este mecanismo, además, asegura QoS gracias a que se realiza un procedimiento de reservación de recursos previo, en donde cada nodo conoce la cantidad de ancho de banda que necesitará para desahogar tanto su tráfico como el de aquellos nodos que lo elijan como ruta hacia el Sink. El concepto de dependencia de ancho de banda, introducido por Dong. y Makrakis (2006), significa que las transmisiones de un nodo afectarán y serán afectadas por las transmisiones de nodos vecinos a uno y dos saltos, sin importar si pertenecen o no a la misma ruta.

## **I.1 Planteamiento del problema.**

Los protocolos para WSN no están diseñados para situaciones de respuesta crítica. Las redes de sensores inalámbricos carecen de un mecanismo que garantice la entrega de paquetes con un límite de retardo, proporcionando con esto calidad de servicio (QoS) y a su vez considerar la fuerte restricción en energía presente en este tipo de redes. El creciente mejoramiento de hardware nos permite considerar sensores con más capacidades, tales como compresión de audio y video o tasas de transmisión más elevadas, sin invadir el campo de acción de redes de más capacidad, como la familia 802.11.

## **I.2 Justificación.**

Algunas aplicaciones requieren que la información que manejan tenga garantías tanto de entrega como de tiempo de retardo. Ofrecer estas garantías en un ambiente con colisiones puede representar un consumo inaceptable de energía para dispositivos con este recurso tan limitado como en las WSN, acortando el tiempo de vida de la red. Un esquema basado en compartición de tiempo puede ayudar a conservar mejor la energía. Esta ganancia se paga con la limitación de ser aplicable solo a redes con movilidad limitada y con una configuración centralizada, lo que disminuye la escalabilidad. Sin embargo, un esquema MAC basado en acceso múltiple por división de tiempo (TDMA) con variables multicapa compartidas con la capa de enrutamiento puede ofrecer garantías de entrega de paquetes con límite de tiempo como parámetro de QoS, consumiendo menor energía, en múltiples aplicaciones donde sus nodos no son móviles y la escalabilidad no es un factor radicalmente importante.

Se pueden considerar aplicaciones tales como cámaras de vigilancia controladas por sensores de movimiento, en donde el tráfico no es constante pero necesita ser reservado, o bien, interfaces humanas para juegos de video o equipos de radio control, como vehículos no tripulados. En sí, esta propuesta sería imprescindible para toda aplicación que tenga requerimientos de garantías de entrega de paquetes y limitaciones de tiempo, y quieran ser funcionales sobre una red de sensores con restricciones en energía.

## **I.3 Objetivos.**

### *1.3.1 Objetivos Generales.*

Proponer un protocolo multicapa para el descubrimiento de topología y selección de rutas entre los nodos de una red de sensores y su sumidero, así como la propuesta de un mecanismo de acceso al medio capaz de proporcionar calidad de

servicio (garantías de retardo y entrega de paquetes), sin descuidar el consumo de energía.

### *1.3.2 Objetivos Particulares.*

Adaptar un mecanismo de enrutamiento para que, además de definir las trayectorias que seguirán los paquetes, pueda recopilar información necesaria para el protocolo de acceso al medio (confirmación de clústeres, interferencia entre clústeres, necesidades de ancho de banda en cada clúster, etc.).

Definir un mecanismo de reservación de recursos.

Implementar un mecanismo de asignación de ventanas de actividad para los diferentes clústeres tomando en cuenta requerimientos de QoS, evitando colisiones entre clústeres y minimizando retardos de Tx de extremo a extremo.

Definir un mecanismo de acceso al medio intra-cluster basado en encuestas, para que el nodo cabeza de cluster (CH) controle las oportunidades de transmisión de los demás nodos del clúster.

Cambiar periódicamente las rutas de los flujos de información en la red, de manera que el consumo de energía sea lo más homogéneo posible.

## **I.4 Delimitaciones y alcances.**

- El mecanismo propuesto es validado sólo por las simulaciones y comparaciones con otro mecanismo similar, no es implementado en ninguna plataforma.
- Se adopta parte del mecanismo propuesto en Gallardo et al (2007) como protocolo de descubrimiento de rutas y asignación de pesos a las mismas

- Se divide el trabajo de la sub-capa de acceso al medio (MAC) en Intra-Clúster e Inter-Clúster, con el fin de definir el método de acceso de cada nodo dentro de un clúster y de cada clúster dentro de la red completa.
- Se diseña un mecanismo de reservación de recursos, recolección de información de interferencias y asignación de ventanas de actividad, el cual es ejecutado después de un descubrimiento de rutas y asignación de pesos, adoptado de Gallardo et al (2007). El mecanismo diseñado define ventanas de actividad para cada clúster, asegurando transmisiones sin colisiones ni interferencias de otros clústeres, empleándose como mecanismo MAC inter-clústeres.
- Se propone un mecanismo de acceso al medio intra-clúster basado en encuestas, igualando las oportunidades de transmisión dentro de las ranuras de actividad de cada clúster.
- Se propone un mecanismo de reservación de recursos a lo largo de las rutas, que sea compatible con el esquema de compartición de tiempo en el que se basa la propuesta global.
- Se define un mecanismo de descubrimiento y reporte de clústeres interferentes, por ser parte de los datos necesarios para el cálculo de ventanas de actividad que ofrezcan garantías de QoS.
- Se simularon en forma conjunta los mecanismos adoptados y propuestos en el paquete de simulación de redes OPNet para su validación conjunta. OPNET Technologies, (2009).
- Se conceptualizó un mecanismo de sincronización interna de la red; su implementación en el simulador esta sujeta a disponibilidad de tiempo.
- Se propone un mecanismo de reconfiguración de la red con opción de no interrumpir la transmisión de datos así como la recuperación de la red tras la falla de un nodo o ruta.

## Capítulo II

---

### Estado del arte.

---

Recientemente ha habido un creciente número de artículos que tratan del tema de calidad de servicio en redes inalámbricas de sensores, los cuales son un signo de la importancia de resolver este problema dada su amplia aplicabilidad. En este apartado discutiremos los beneficios y desventajas de algunas de esas propuestas.

Algunos protocolos trabajan con bajos ciclos de actividad, dado que uno de los objetivos principales en una WSN es diseñar protocolos eficientes en energía. Estos ciclos cortos de actividad incluyen períodos en los que los transmisores están en modo dormido. SMAC fue uno de los primeros protocolos diseñados con eficiencia energética en mente, y trabaja por medio de tiempos coordinados de dormir y despertar entre nodos vecinos, esto en combinación con un esquema de transmisión basado en contienda, Wei Ye et. al. (2004). Debido a este rol pionero, SMAC se ha convertido en un protocolo de referencia para WSN. Otros ejemplos de protocolos que usan cortos ciclos de actividad y coordinación de actividades son T-MAC, Tijs van Dam y Koen Langendoen, (2003), R-MAC, Shu Du et. al. (2007), y DW-MAC, Yanjun Sun et. al. (2008). También se han explorado propuestas distintas, algunas basadas en transmisiones asíncronas, en donde los nodos, en vez de acceder al medio durante tiempos definidos de actividad, intercambian mensajes indicando cuándo pueden ocurrir las transmisiones. Algunos ejemplos de protocolos que usan ciclos de transmisión asíncronos son B-MAC, Polastre et. al. (2004), X-MAC, Buettner et. al. (2006) y Wise-MAC, Amre El-Hoiydi y Jean-Dominique Decotignie (2004).



Existen varias propuestas que consideran TDMA como una manera de evitar colisiones, como ejemplo están los trabajos de Cano-Tinoco y Fapojuwo (2007); Su y Zhang (2008); Ergen y Varaiya (2006). TDMA es conocido por ser costoso tanto en sobre encabezado como en complejidad computacional, además de requerir una fina sincronización entre nodos. Desafortunadamente, estas propuestas no describen a detalle cómo y qué tipos de paquetes intercambian los nodos para especificar un calendario TDMA, lo que hace difícil estimar el sobreencabezado (tanto en mensajes de control como en tiempo de configuración). Pedamacs, Ergen y Varaiya (2006), por ejemplo, supone que el Sink tiene una potencia de transmisión ilimitada, que le permite comunicarse directamente con todos los nodos en la WSN, lo que es una condición que difícilmente puede ser satisfecha en sistemas reales.

Dado que uno de los principales objetivos en el diseño de protocolos para WSN es la eficiencia en energía, algunos de ellos incluyen en su diseño el uso de períodos cortos de actividad y permiten a sus nodos cambiar a un estado de dormir en donde apagan sus radios transmisores por ciertos períodos. SMAC es un protocolo bien conocido diseñado específicamente para WSN, en el cual, el principal objetivo es reducir el consumo de energía y lograr buenos niveles de escalabilidad y evasión de colisiones, empleando un esquema combinado de calendarización y contienda, Wei Y. et al. (2004).

Otras propuestas, como las de Boukerche et al. (2007), Peng et al. (2008) y Chen, Nasser (2008) y Felemban et. Al (2006), se enfocan en proporcionar cierta diferenciación de calidad servicio y contemplan el uso de múltiples trayectorias, lo que consiste en darle a un nodo la posibilidad de usar una de varias rutas para enviar un mensaje a un destino en particular a un tiempo dado, como método para que mensajes sensibles a retardo eviten rutas congestionadas. Boukerche et al. (2007) propone que el Sink clasifique las rutas de acuerdo a su retardo,

confiabilidad y consumo de energía, de modo que los nodos puedan enviar datos sensibles al retardo a través de las rutas más rápidas, datos sensibles a errores por las rutas más confiables y datos sin limitantes por las rutas con menor consumo de energía. Este es un buen método de diferenciación de servicios, pero es difícil ofrecer garantías de QoS, dado que por ejemplo, la ruta más rápida puede no ser lo suficientemente rápida bajo ciertas condiciones. Peng et al. (2008) proponen una técnica adaptiva basada en un algoritmo de colonia de hormigas en que cual las rutas son descubiertas cuando se necesitan (algoritmo reactivo), enviando un mensaje de prueba. Cuando este mensaje alcanza el destino deseado a través de múltiples rutas, se envían mensajes por las rutas en reversa. Este mensaje marca los nodos visitados de acuerdo a la calidad de la ruta (ancho de banda disponible, retardo esperado y tasa de pérdidas). Posteriormente, los nodos vecinos comparten sus etiquetas de calidad, de modo que cada nodo es entonces capaz de seleccionar la mejor opción cuando se necesita reenviar una trama de datos. Este algoritmo presenta un problema de escalabilidad dado que tiene que ser ejecutado por cada par de nodos fuente-destino. Además, al igual que en el caso mencionado anteriormente, seleccionar la mejor opción localmente no garantiza un desempeño aceptable extremo a extremo. Chen y Nasser (2008) por otro lado, hacen énfasis en la confiabilidad de las rutas, por lo que este protocolo no puede ser aplicable para tratar con datos en tiempo real. Finalmente, con MMSPEED, Felemban et. al (2006), realizan una estimación local de la confiabilidad y retraso que pueden experimentar los paquetes a lo largo de las diferentes rutas disponibles. Cada paquete puede elegir la mejor combinación de opciones de servicios, dependiendo de sus requerimientos. Dado que tanto el retraso y la confiabilidad dependen del patrón de tráfico y de las decisiones de reenvío de tráfico, basadas a su vez en los mismos retrasos y confiabilidad, se genera una retroalimentación que puede generar una alta variabilidad en la dinámica del tráfico, provocando una necesidad de re-calcular, lo antes posible, las decisiones de tráfico. Aunado a esto, los nodos requieren tener registro constante de los límites de tiempo y el tiempo transcurrido.

Este algoritmo, MMSPEED, requiere grandes recursos computacionales en cada nodo.

Otros autores, tales como Aalsalem et al (2008) y Gelenbe y Ngai (2008), proponen introducir prioridades, como una manera de reducir la latencia de mensajes sensibles al tiempo. Establecer prioridades para diferentes tipos de tráfico es de nuevo un mecanismo de diferenciación de servicios, lo que significa que no es posible ofrecer garantías de QoS, dado que aunque el servicio para tráfico de alta prioridad será mejor que el de baja prioridad, el primero puede no ser suficiente para garantizar retardos o hasta la entrega misma de paquetes.

Una alternativa es el uso de técnicas de control de congestión como un modo reactivo de detectar un bajo desempeño y tomar acciones para solucionarlo, Wang et al. (2007), Rahman et al. (2008), Shirazi et al. (2008). Estos algoritmos se basan en una continua evaluación de los niveles de congestión para decidir en qué momento actuar. Cuando uno de estos mecanismos decide que se necesita tomar acciones, implica generalmente, descartar tráfico de baja prioridad o bien notificar a las fuentes de tráfico que existe una congestión, para que éstas bajen su tasa de transmisión. En el primer caso, se supone implícitamente que el tipo de tráfico tolera las pérdidas de paquetes de baja prioridad, lo cual no siempre está justificado. En el segundo caso, se supone que las fuentes de tráfico son capaces de reducir su tasa de generación de paquetes, lo que nuevamente puede no ser tolerado por las fuentes. Aunado a estas limitantes, el constante monitoreo del estado de congestión de la red puede incurrir en porcentajes muy altos de sobre encabezados, teniendo siempre el riesgo de perder datos importantes cuando las condiciones de una congestión se presenten más rápido de lo que el sistema lo pueda manejar.

## Capítulo III

---

### Metodología.

---

En este capítulo se describe a detalle la propuesta y su implementación en la herramienta de simulación OPNET, OPNET Technologies, (2009). Nuestra propuesta está basada en reservar suficiente recurso que garantice el desempeño de QoS requerido, Ruiz et. al. (2009).

Se consideró un conjunto de redes homogéneas de sensores, las cuales cuentan con un nodo sumidero o “Sink” de mayores capacidades, al que todos los nodos de la red reportan sus mediciones por medio inalámbrico de múltiples saltos (Ad-Hoc). QUATTRO, como hemos llamado a esta propuesta, consta de las siguientes fases:

1. Descubrimiento de rutas y asignación de pesos.
2. Selección y reservación de rutas.
3. Recolección de información de interferencias entre clústeres.
4. Asignación de ventanas de actividad a los clústeres. (Al terminar esta fase cada nodo tiene suficiente ancho de banda asignado para garantizar la QoS).
5. Fase de operación normal.
6. Reconfiguración y recuperación tras fallas de nodos.

Las fases del uno al cuatro conforman la configuración de la red, ejecutándose secuencialmente al ser desplegados los nodos para establecer las rutas, reservaciones de recursos y ventanas de actividad necesarias para la transmisión de datos. Una vez configurada la red, inicia la fase de operación normal en la que

se dan las transmisiones de datos. La fase de operación normal se mantiene activa hasta que ocurra una reconfiguración programada o la pérdida de algún nodo, desencadenando la fase 6 del protocolo. La idea de tener reconfiguraciones periódicas tiene dos objetivos. Por un lado asegurarse de que la información de enrutamiento se mantiene actualizada, y por otro que se distribuya de manera más equitativa el consumo de energía en los nodos. Esta idea fue tomada del protocolo LEACH, de Heinzelman et. al. (2000).

### **III.1 Descubrimiento de rutas y asignación de pesos.**

La primera fase de QUATTRO consiste en descubrir un conjunto de rutas disjuntas (para no consumir recurso en rutas repetidas) desde cada nodo sensor hacia el nodo Sink, asignando un peso a cada una proporcional a los cuellos de botella en carga y energía en dichas rutas. Este procedimiento se adoptó del mecanismo propuesto en Gallardo (2007) que a su vez es una generalización del trabajo de Lou (2005) y consiste en cuatro tareas.

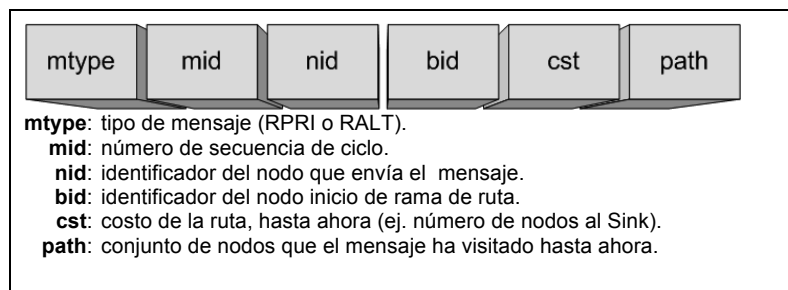
La primera tarea para descubrir rutas se inicia con el Sink difundiendo un paquete de actualización de rutas, llamado RPRI, el cual tiene como objetivo dar a conocer a cada nodo sensor su ruta primaria hacia el Sink. El formato de este mensaje se muestra en la Figura 1. Cada nodo, cuando recibe este paquete por primera vez, marca al nodo del que lo recibió como su padre y difunde de nuevo el mensaje. Este proceso continúa recursivamente hasta que cada nodo en la red ha recibido una copia de este paquete e identificado a su nodo padre, quien es el siguiente salto en la ruta primaria hacia el Sink.

La segunda tarea del algoritmo de descubrimiento de rutas consiste en transmitir mensajes del tipo RALT, empleados por cada nodo para compartir con sus vecinos el conjunto de rutas alternativas descubiertas, ya sea por escuchar el paquete RPRI de un nodo diferente al marcado como padre o por paquetes RALT

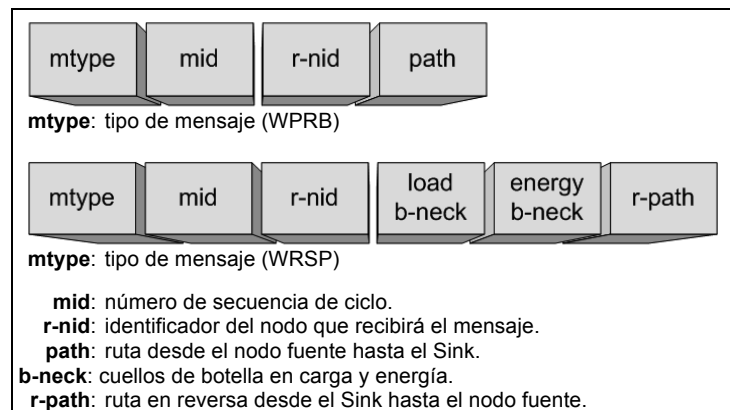
recibidos de otros vecinos. Cuando esta segunda tarea termina, los nodos han descubierto todas las rutas disjuntas disponibles con mínimo costo disponibles. En nuestro caso el costo está dado por el número de saltos y cantidad de tráfico en cada ruta, aunque se pueden incluir otras métricas como tasa de error de bit (BER) o nivel de conglomeración de nodos (tasa anticipada de colisiones).

En la tercera tarea cada nodo sensor envía un paquete WPRB (de prueba) a través de cada una de las rutas descubiertas, incluyendo la ruta primaria y alternativas. La Figura 2 muestra la lista de campos incluidos en este mensaje. Cuando un nodo recibe un paquete de este tipo, incrementa el contador de rutas que potencialmente pasarán a través de él (*num\_routes*) y reenvía el paquete camino al Sink.

Finalmente, la cuarta tarea consiste en que el Sink envía paquetes de respuesta WRSP a través de cada ruta por las que recibió los paquetes de prueba WPRB.



**Figura 1. Formato de paquete, descubrimiento de rutas.**



**Figura 2. Formato de paquete, mensajes de prueba.**

Cuando un nodo recibe un WRSP, lo reenvía al siguiente nodo en la ruta después de etiquetar el paquete con sus propios valores de cuellos de botella en carga y energía según los siguientes criterios. Si el valor almacenado en el paquete como cuello de botella en carga, es menor a la carga esperada en ese nodo ( $num\_routes$ ), se sustituye el valor en el paquete por el valor de  $num\_routes$  del nodo. De manera similar, si el valor de cuello de botella en energía, almacenado en el paquete es mayor a la energía restante en el nodo, la etiqueta del paquete será reemplazada por la del nodo. Este paquete llegará eventualmente al nodo al final de la ruta en reversa. Cuando todos los paquetes WRSP hayan llegado a sus nodos destino, o que haya expirado un temporizador, cada nodo asignará un peso  $p_i$  a cada una de sus respectivas  $N$  rutas, en función de la siguiente expresión.

$$p_i = \frac{\varepsilon_i}{\lambda_i \cdot (h_i)^\beta}, \quad \text{para } i = 1, 2, \dots, N \quad (1)$$

Donde:

- $\varepsilon_i =$  Cuello de botella en energía de la ruta  $i$  recibido en el paquete WRSP de esa ruta (o 0 si no se recibió el paquete).
- $\lambda_i =$  Cuello de botella en carga de la ruta  $i$  recibido en el paquete WRSP de esa ruta (o 1 si no se recibió el paquete).
- $h_i =$  Número de saltos en la ruta  $i$ .

$\beta \in (0, 1) =$  Factor que define el impacto deseado del número de saltos en el peso de las rutas.

La figura 3 muestra un resumen del proceso de descubrimiento de rutas y asignación de pesos.

### III.2 Selección y reservación de rutas.

La segunda fase de QUATTRO consiste en reservar recursos por una de las rutas existentes entre cada nodo sensor y el Sink. Cada nodo generará tráfico durante la fase de operación normal que será detallada más adelante. La ruta seleccionada debe ser capaz de satisfacer los requerimientos de QoS del tráfico que será generado para esa ruta. Estas reservaciones se realizan salto a salto. Dos nodos se convierten en miembros de un clúster cuando se logra exitosamente un intento de reservación entre ellos. El nodo que acepta una reservación se convierte en cabeza de clúster (CH) del nodo hijo que la solicita, y se compromete a reenviar todo el tráfico generado por los nodos que ha aceptado. Los clústeres crecerán

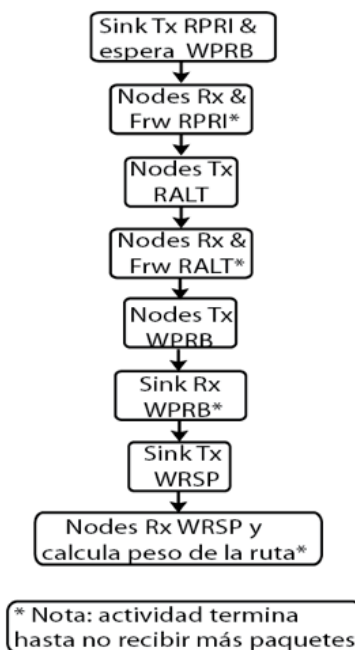
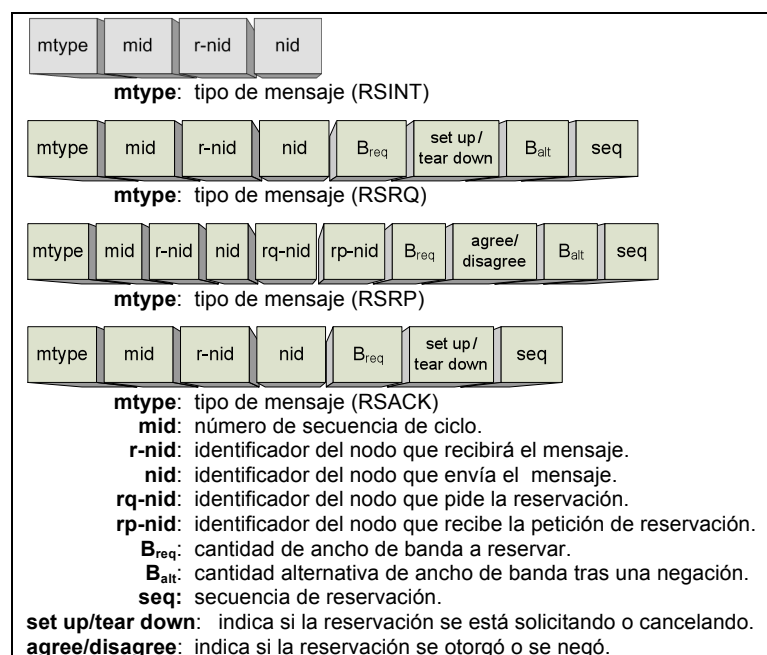


Figura 3. Resumen de enrutamiento y asignación de pesos.



conforme el CH acepte más peticiones de reservación. Como podemos ver, en esta propuesta, clúster es definido como un conjunto de nodos vecinos a un salto que necesitan comunicarse directamente, y el objetivo es coordinar los patrones de tiempos durante los cuales, estos clústeres permanecen dormidos o despiertos. Cada clúster adoptará el identificador de su CH como identificador del clúster. El tráfico convencional se genera en los nodos sensores y tiene como destino el Sink, sin embargo, puede existir tráfico en sentido opuesto (desde el Sink hacia los nodos sensores), pero este será en general menos intenso y no de tiempo real, por lo que no es necesario definir reservaciones para este tipo de tráfico.

Es importante notar que, cuando un nodo accede a reenviar tráfico de otros nodos, se convierte en miembro de dos clústeres, uno en el que es cabeza (a cargo de recolectar los datos de sus miembros) y otro en el que es solo otro miembro del clúster al que reenviará tanto su propio tráfico como el recolectado desde sus hijos.



**Figura 4. Formato de paquete, mensajes de reservación-  
asociación.**

Como podemos ver, la reservación de recursos se realiza salto por salto. Con el fin de coordinar el proceso de reservación, el Sink lo inicia enviando un paquete de intención de reservación (RSINT) en modo difusión. Cuando los vecinos a un salto del Sink escuchan este mensaje envían un mensaje RSINT propio con destino al Sink. Como se muestra en la figura 4, el mensaje RSINT contiene los identificadores del que lo envía y del que lo debe recibir. El enviar un paquete RSINT significa que el nodo que lo envía ha elegido al nodo receptor como siguiente nodo potencial en ruta hacia el Sink, indicándole la intención de iniciar posteriormente un proceso de reservación de recursos. Cuando un nodo diferente al Sink o a sus vecinos a un salto escuchan que hay un paquete RSINT transmitiéndose en el vecindario, almacenan en una lista el identificador de los nodos transmisores de estos paquetes y lanzan o reinician un temporizador. Cuando este temporizador expira, el nodo selecciona a uno de los nodos de la lista antes mencionada como CH en potencia, en base al peso de la ruta a la que esos nodos pertenecen, enviándole un paquete RSINT. Este proceso excluirá a las rutas de las cuales no se haya escuchado un paquete RSINT. El objetivo de este proceso ordenado es evitar rutas cicladas, debidas a la naturaleza salto a salto del proceso de reservación.

Tras mandar el paquete RSINT cada nodo inicia un contador. Cuando este contador expira se inicia la gestión de reservaciones en los nodos hoja de rama de ruta, aquellos que no recibieron mensajes RSINT. La gestión de reservación se inicia enviando un paquete de petición de reservación o RSRQ al nodo seleccionado como CH en potencia. Si un nodo recibió uno o más mensajes de intención (RSINT), este nodo se convierte en un CH potencial, lo cual es útil para impedir que estos inicien la gestión de reservación hacia el siguiente salto al Sink, antes de recibir las peticiones de reservación de todos sus nodos hijos potenciales (aquellos de los que recibió un mensaje RSINT). Una vez que un CH potencial recibe las peticiones de reservación (RSRQ) de todos sus hijos potenciales, este

se consolida como CH de aquellos nodos a los que acepte, y envía un paquete RSRQ a su propio CH requiriendo recurso suficiente para él y todos sus hijos.

Gallardo (2007) explica que garantizando cierta cantidad de ancho de banda es suficiente para asegurar QoS, como por ejemplo caudal eficaz, retrasos de paquetes y pérdidas por desborde de búfer, que serían incluidas en las garantías como consecuencia. Es necesario garantizar una velocidad de transmisión mayor a la velocidad con la que el nodo genera el mayor de sus tráficos.

Cuando inicia el proceso de reservación, todos los nodos empiezan con un ancho de banda disponible igual a la tasa de datos efectiva del canal  $R$ , lo que depende de la capa física y las políticas de acceso al medio empleadas. Por ejemplo, en un sistema basado en encuestas como el que se propone en esta tesis, tiene un caudal eficaz del 85% de la tasa efectiva del canal, Gallardo (2007). La ecuación 2 establece las proporciones en las que consiste el ancho de banda disponible en cada nodo.

$$B_{avail} = R - (2 B_{committed} + B_{own} + B_{overheard}) \geq 0 \quad (2)$$

En la ecuación anterior,  $B_{avail}$  corresponde al ancho de banda disponible para nuevas reservaciones,  $B_{committed}$  es el total del tráfico que el nodo ha comprometido a reenviar en nombre de otros nodos,  $B_{own}$  es el tráfico que el propio nodo generará, y  $B_{overheard}$  es la cantidad de ancho de banda de reservaciones escuchadas en el vecindario de las que el nodo no es parte de la ruta.  $B_{committed}$  se cuenta doble dado que se emplea el mismo canal tanto para recibir esa cantidad de tráfico como para reenviarla al siguiente salto hacia el Sink.

Es importante definir a  $B_{req}$  como la cantidad de ancho de banda que es reservada por cada nodo. Primero, cada nodo hoja envía un paquete de petición de reservación (RSRQ) a su CH.  $B_{req}$  en este caso será igual a  $B_{own}$  debido a que los

nodos hoja no necesitan reenviar ningún dato pues no tienen hijos. Posteriormente, cada nodo que no es hoja, por consiguiente CH, esperará a recibir las peticiones de reservación RSRQ de todos sus nodos hijos para enviar una petición de reservación a su CH. Desde este punto, toda  $B_{req}$  será igual a  $B_{committed} + B_{own}$ , dado que cada CH necesita reenviar al siguiente CH los datos que recibió de sus hijos además de los datos propios generados. Los CH que reciben un mensaje de petición de reservación confirmarán si su ancho de banda disponible  $B_{avail}$  es al menos una, dos o tres veces  $B_{req}$ , dependiendo de si se trata del Sink, de un CH a un salto del Sink o de un CH a más de un salto del Sink, respectivamente. Esto es para anticiparse a las reservaciones que él como CH ha otorgado, que él como nodo tiene que hacer con su CH y que su CH mismo tiene que reservar con el siguiente salto hacia el Sink. Todos los nodos que pueden escuchar el mensaje de reservación (y no pertenecen al par de nodos reservando) también confirmarán si su ancho de banda disponible  $B_{avail}$  es al menos igual al que se está reservando en el vecindario  $B_{req}$ . Si no hay ancho de banda disponible suficiente  $B_{avail}$ , ya sea por peticiones entrantes o por peticiones escuchadas en el vecindario, el nodo sin el recurso suficiente enviará un mensaje de respuesta (RSRP) indicando que esa nueva reservación no es posible.

Si la reservación es exitosa, satisfaciendo la ecuación 2, el nodo CH que recibe la petición enviará una respuesta positiva al nodo que la solicita, empleando un paquete RSRP. Si todos sus nodos hijos han completado el proceso de reservación, el CH reiniciará este proceso como un nodo miembro de otro clúster con su propio CH, en el siguiente salto hacia el Sink. Si un nodo no escuchó el paquete RSRQ pero escucha el paquete RSRP también verifica si tiene recurso disponible para alojar dicha reservación del vecindario. Si esta reservación representa problema debido a una falta de recursos, este nodo envía una respuesta impidiendo o cancelando esa reservación, tal como sucede cuando se escucha un paquete RSRP que sobrepasa el recurso disponible. Cuando un nodo envía un paquete de respuesta negativo, incluirá el cambio  $B_{alt}$  con la cantidad de

$B_{avail}$ , con el propósito de que los nodos en la reservación negada se ajusten al recurso disponible.

Si una reservación no es aceptada, el nodo solicitante intenta la misma reservación empleando otra ruta. Si todas las rutas fallan reducirá la cantidad de ancho de banda solicitado, cancelando reservaciones previamente concedidas a algunos de sus hijos, si tiene alguno, e intenta reservar de nuevo con la ruta que tenga mayor recurso disponible. Cuando un nodo selecciona una segunda ruta, debido a que la primera negó la reservación, este mandará de nuevo un mensaje de intención antes de enviar el paquete RSRQ. Si el nodo que recibe esa intención ya gestionó su propia reservación, reenviará el paquete de intención entrante al siguiente nodo hacia el Sink. La intención con esto es que, aun cuando algunas reservaciones tomen más tiempo de lo esperado, el Sink tenga conocimiento de que se siguen gestionando reservaciones, y espere un poco más antes de iniciar la siguiente fase del proceso de configuración, que es la recopilación de información de interferentes.

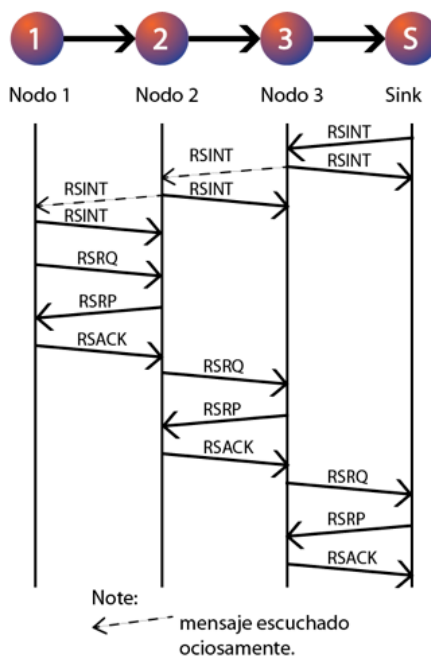


Figura 5. Resumen del proceso de reservación de recursos.

En resumen, se emplean tres paquetes de control para reservar ancho de banda: RSRQ, RSRP y RSACK. El paquete RSRQ es empleado para solicitar ancho de banda. El paquete RSRP se usa para enviar respuestas positivas o negativas de reservación. Hay que hacer notar que un mensaje RSRP puede ser enviado no solo en respuesta a un RSRQ recibido, sino además por una reservación escuchada en el vecindario para la cual no se tenga suficiente ancho de banda disponible para compartir. Finalmente el paquete RSACK es usado por los nodos que han recibido una respuesta a su solicitud de reservación, positiva o negativa, y se emplea para acusar de recibida dicha respuesta. Los mensajes intercambiados para establecer una nueva reservación tienen el objetivo de alcanzar a todo el conjunto de nodos del posible vecindario interferente. Es posible que dos nodos cercanos, aunque no se puedan comunicar, si se causen interferencia uno al otro ya que aun que el nivel de energía recibido no sea suficiente para decodificar el mensaje, esa energía recibida en las antenas es interferencia. Para tomar esto en cuenta, los paquetes de reservación pueden ser transmitidos con una mayor potencia que cualquier otro mensaje, incluyendo aquellos en las otras etapas de configuración y los paquetes de datos transmitidos en la fase de operación normal. De esta manera, aun aquellos nodos que están lo suficientemente lejos para no comunicarse adecuadamente, pero si para causar interferencias, serán capaces de reconocerse como interferentes potenciales al momento de hacer las reservaciones de recursos. La Figura 4 muestra el formato de los paquetes de reservación. Como se puede ver, todos ellos contienen un número de secuencia con el fin de evitar el procesar peticiones de reservación repetidos. La Figura 5 muestra en resumen el proceso de reservación de recursos.

Los paquetes RSRP y RSACK no solo tienen el objetivo de mantener la cuenta del ancho de banda reservado en el vecindario, sino además de conocer y enlistar aquellos clústeres con los que potencialmente se tendrá interferencia. Esta lista es necesaria para la siguiente fase del protocolo.

### III.3 Recolección de información de interferencias entre clústeres.

Es necesario recolectar información sobre aquellos nodos que son interferentes potenciales en cada vecindario de clúster, de modo que el Sink sea capaz de calcular un calendario de transmisiones sin interferencias. Esta recopilación se realiza al final de la fase de reservación de recursos, mediante un paquete CISTART (inicio de recolección de interferencias de clúster) difundido desde el Sink hasta inundar todos los nodos de la red. Este mensaje contiene solo dos campos, “*mtype*” definiendo el tipo de paquete con la etiqueta CISTART y el campo “*mid*” definiendo el número de ciclo de configuración.

Para recolectar información de interferentes se emplea un paquete llamado CIINFO, partiendo de los nodos hoja, que además ayuda a calcular la profundidad de cada clúster. Profundidad de clúster es definida como la distancia en número de saltos hasta el nodo más alejado del Sink, perteneciente a la misma rama de ruta del clúster o alguna de sus bifurcaciones. Los nodos que no son CH, a los que hemos llamado nodos hoja por ser el final de cada rama de ruta hacia el Sink, serán los que inicien la recolección de información de interferentes. Cada nodo hoja enviará un mensaje CIINFO a su respectivo cabeza de clúster, conteniendo en el campo “*local interf*”, la lista de clústeres interferentes que él ha escuchado a

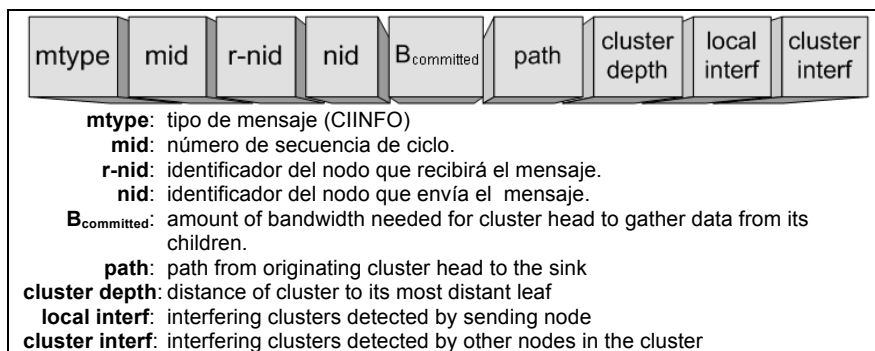


Figura 6. Formato de paquete, recolección de interferentes.

su alrededor (no nodos individuales), y cero en el campo “*cluster depth*”. En este mensaje inicial, los campos “*B<sub>committed</sub>*”, “*path*” y “*cluster interf*”, mostrados y descritos en la figura 6, estarán vacíos indicando al receptor que el que lo envía no es un CH, por consiguiente ese mensaje no debe ser reenviado hasta el Sink. Cuando un CH recibe un CIINFO de un miembro del clúster en el cual el campo “*path*” está vacío, o contiene el mismo valor que el campo “*nid*” (el paquete fue generado por el mismo nodo hijo), el CH receptor actualizará su lista de clústeres interferentes detectados por los miembros del clúster, basándose en el campo “*local interf*” del paquete recibido, y actualizará su cuenta de profundidad a la mayor de las profundidades reportada por los miembros del clúster más uno. Todo paquete CIINFO que llegue a un CH, y que haya sido generado por otro CH (campo “*cluster depth*” diferente de cero), será reenviado al siguiente CH después de agregar su identificador a la lista contenida en el campo “*path*”. Cuando un CH ha recibido los paquetes CIINFO de todos sus nodos hijos, llenará los campos de su propio paquete CIINFO y lo enviará al siguiente CH en ruta al Sink. Este paquete, contendrá en el campo “*path*” y “*nid*” el identificador del CH que lo genera, el campo “*r-nid*” al identificador del siguiente CH en ruta al Sink, el campo “*cluster interf*” a los identificadores de clúster reportados como “*local interf*” por sus nodos hijos, el campo “*local interf*” contendrá los identificadores de aquellos clústeres que él mismo es capaz de escuchar en el vecindario, y no estén ya incluidos en el campo “*cluster interf*” llenado anteriormente, el campo “*cluster depth*” la profundidad

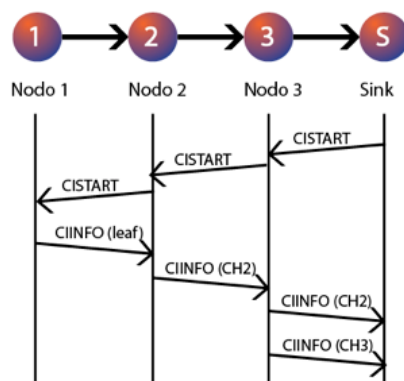


Figura 7. Resumen del proceso recolección de interferentes.



calculada del clúster y finalmente el campo " $B_{committed}$ " contendrá la suma de ancho de banda concedida a sus hijos.

Una vez que el Sink reciba los paquetes CIINFO de cada uno de los clústeres de la red, creará un arreglo temporal llamado " $temp\_sched$ ", el cual contendrá los clústeres ordenados de menor a mayor profundidad. Cada elemento de este arreglo incluirá el identificador de clúster, profundidad, lista de nodos de ruta, ancho de banda requerido por sus miembros y la lista de clústeres interferentes, resultado de la unión de los campos " $local\ interf$ " y " $cluster\ interf$ " del paquete CIINFO del clúster en cuestión. La figura 7 muestra un resumen del proceso de recolección de interferentes.

### III. 4 Asignación de ventanas de actividad a los clústeres.

Al terminar el proceso descrito en la sección anterior, el Sink tendrá suficiente información para asignar ventanas de actividad suficientemente grandes a los

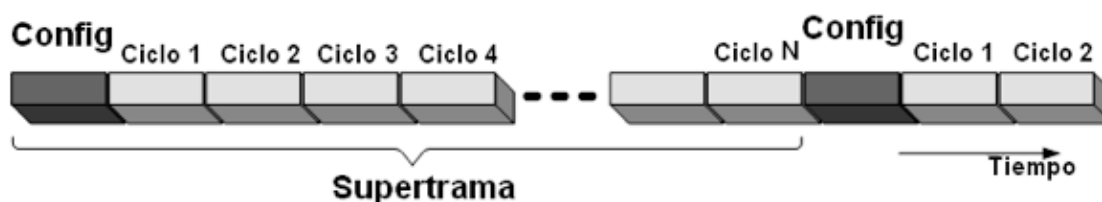


Figura 8. Estructura de supertrama.

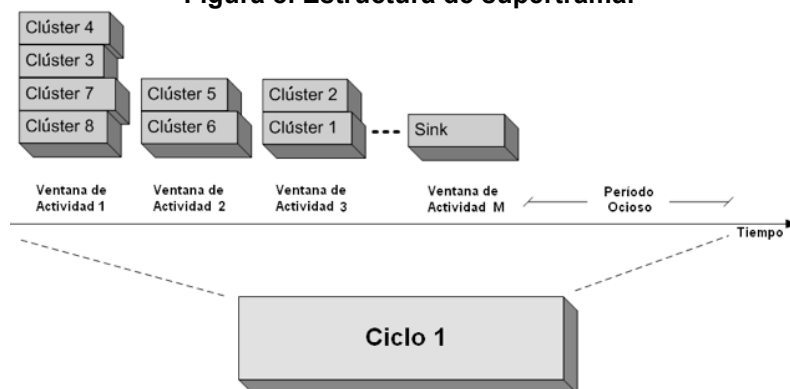


Figura 9. Estructura de un ciclo de trama.

diferentes clústeres, de tal manera que no interfieran entre sí. La asignación de ventanas de actividad a los clústeres es la última etapa de lo que se refiere al proceso de configuración. Es importante mencionar que el proceso de configuración será repetido periódicamente, con relativamente poca frecuencia (posiblemente del orden de horas o días), la cual depende del tamaño y densidad de la red, energía residual de los nodos, intensidad del tráfico y/u otros factores, con el objetivo de reasignar responsabilidades de reenvío de tramas, redistribuyendo el consumo de energía para incrementar el tiempo de vida de la red. Las ventanas de actividad calculadas por el Sink se repetirán periódicamente, dando a los miembros de cada clúster suficiente tiempo para reenviar al CH toda la información acumulada durante el tiempo que permanecieron inactivos. El intervalo de tiempo que incluye una ventana de actividad para cada uno de los clústeres de la red es definido como ciclo, tal como se muestra en las figuras 8 y 9. Dependiendo de la disponibilidad de ancho de banda y de la tolerancia de retraso de paquetes de la aplicación, un ciclo puede incluir períodos ociosos, en los cuales ningún clúster se encuentre activo con el propósito de reducir la duración del ciclo y ahorrar más energía. Las fases que conforman la configuración aunada a una secuencia de varios ciclos de transmisión de datos, es llamada súper trama. Durante una ventana de actividad podrían permanecer encendidos uno o varios clústeres siempre y cuando no sean interferentes entre si, como se muestra en la figura 9. El algoritmo acomoda las ventanas de actividad de acuerdo a la profundidad de cada clúster, de tal modo que la información recopilada pueda moverse en cada ventana de actividad, hacia clústeres que no han tenido oportunidad de transmitir en el mismo ciclo y se encuentran más cercanos al Sink. Con este mecanismo, toda la información presente en los búferes de los nodos al principio de cada ciclo, tendrá garantizado que alcanzará al Sink al final del mismo. Ésta es la razón principal por la que se considera que garantizando suficiente ancho de banda para vaciar todos los búferes dentro de un ciclo, se puede asegurar satisfacer las métricas de QoS (caudal eficaz, retardo, pérdidas por desborde de búferes).

El proceso de asignación de ventanas de actividad se inicia una vez que el Sink tiene la información de interferentes de todos los nodos de la red, o bien expira un temporizador definido para iniciar esta tarea. El Sink crea una estructura bidimensional llamada “*cluster\_sched*”, en la cual se alojan clústeres no interferentes entre si y de la misma densidad en la misma columna. Esto significa que las ventanas de actividad de dichos clústeres podrán ser calendarizadas simultáneamente. De la misma manera, los clústeres alojados en columnas izquierdas tendrán su ventana de actividad calendarizada antes de aquellos clústeres calendarizados en columnas derechas. Para lograr esto, el Sink remueve clústeres uno a uno del arreglo temporal “*temp\_sched*” mencionado anteriormente. Si la columna de la estructura “*cluster\_sched*” en la que se está escribiendo no contiene clústeres de la misma profundidad del siguiente clúster a escribir (removido del vector “*temp\_sched*”), éste se escribe en una columna nueva vacía. Si por otro lado, ya existe en la estructura “*cluster\_sched*” al menos una columna alojando clústeres de la misma profundidad del siguiente clúster a escribir, el Sink intentará alojarlo en una de esas columnas siempre y cuando no interfiera con alguno de los clústeres ya alojados en la misma. Si se encuentra mas de una columna en la estructura “*cluster\_sched*” con la misma profundidad del clúster a escribir, sin interferir con los clústeres alojados, se elegirá aquella columna en la que la diferencia de ancho de banda requerido por el clúster a alojar y el máximo requerido por los clústeres ya alojados sea negativa ( $Clúster\_Nuevo\_B_{req} < Clústers\_columna\_maxB_{req}$ ) o la menor posible. Si el clúster a escribir no puede ser acomodado en ninguna de las columnas existentes con nodos de la misma profundidad debido a interferencias con ellos, el clúster a escribir se alojará en una nueva columna vacía.

Una vez que todos los clústeres han sido acomodados en la estructura “*cluster\_sched*”, el Sink debe calcular la duración de las  $M$  ventanas de actividad resultantes de las columnas de dicha estructura. Con el fin de calcular estas

ventanas de actividad de cada columna, se calcula primero el tiempo de actividad mínimo necesario por cada clúster, siguiendo la ecuación 3.

$$T_{clust}(j) = \frac{B_{committed}(j)}{R} \cdot T_{cycle} \quad (3)$$

De la ecuación 3,  $R$  representa la tasa efectiva de datos del canal,  $B_{committed}(j)$  es el ancho de banda necesario por el CH( $j$ ) para recolectar todos los datos de sus hijos y  $T_{cycle}$  es la duración de cada ciclo.  $T_{cycle}$  debe ser lo suficientemente corto para que el máximo retardo experimentado por la trama de datos, desde que se genera hasta que llega al Sink esté dentro de valores aceptados por la aplicación, y suficientemente largo para alojar las transmisiones necesarias de los miembros de cada clúster. Partiendo del tiempo necesario por cada clúster, podemos calcular el tiempo de cada columna de la estructura “*cluster\_sched*” empleando la ecuación 4.

$$T_{win}(i) = \max_{j \in S_i} T_{clust}(j), i \in \{1, 2, \dots, M\} \quad (4)$$

Donde  $S_i$  es el conjunto de clústeres a ser calendarizados dentro de la  $i$ -ésima ventana de actividad. El tamaño de las ventanas de actividad debe ser incrementado en una pequeña proporción, con el fin de tomar en cuenta el flujo de tráfico en dirección opuesta al Sink. Es necesario hacer notar que si la ecuación 5 se cumple, significa que no hay suficiente ancho de banda en la red para calendarizar ventanas de actividad no traslapadas.

$$\sum_{i=1}^M T_{win}(i) > T_{cycle} \quad (5)$$

Una posible solución en este caso es intentar iniciar un nuevo ciclo antes de que termine el ciclo anterior, siempre manteniendo la condición de evitar interferencias entre clústeres traslapados. Para realizar esto, el Sink examinará las consecuencias de traslapar ciclos consecutivos por una cantidad de tiempo que llamaremos  $T_{overlap}$ , de longitud igual a la primer ventana de actividad. Si se encuentra que no hay interferencia entre los clústeres de las ventanas traslapadas, el Sink analizará la posibilidad de traslapar ciclos consecutivos el tiempo de duración de las primeras dos ventanas de actividad, y así sucesivamente mientras no haya interferencias entre los clústeres de las ventanas traslapadas. Si este proceso termina y se sigue manteniendo la desigualdad expresada en la ecuación 6, significa que el sistema no tiene suficiente ancho de banda para satisfacer los requerimientos de QoS de sus nodos.

$$\sum_{i=1}^M T_{win}(i) - T_{overlap} > T_{cycle} \quad (6)$$

Cuando esto pasa se debe correr un mecanismo de control de topología como los mostrados en Krishnamachari (2005) para decidir si reduciendo el número de nodos (reduciendo la cantidad de tráfico) es suficiente para satisfacer los requerimientos de conectividad y cobertura de la red.

Nótese que el procedimiento de reservación está a cargo de la primera valoración de si el sistema tiene suficientes recursos para satisfacer los requerimientos de QoS. Sin embargo, alojar diferentes ventanas de actividad a los clústeres, significa que cuando un nodo dado esté transmitiendo, todos los miembros de cada clúster vecino deben permanecer en silencio, y no sólo aquellos que pueden experimentar una colisión con el nodo en cuestión. En otras palabras, alojar ventanas de actividad es más demandante de recursos que la reservación local, lo que significa

que mientras la reservación de recursos pudo haber sido exitosa, la asignación de ventanas de actividad aún puede fallar.

Una vez que el Sink ha terminado de calcular las ventanas de actividad de los clústeres, tiene que informar a cada uno de ellos sus respectivas ventanas. Para esto, el Sink enviará un mensaje AWN (notificación de ventana de actividad) a cada uno de los cabezas de clúster, conteniendo toda la información necesaria, como se muestra en la figura 10. Una vez que cada CH recibe su respectivo AWN, enviará un paquete AWLN (notificación local de ventana de actividad) a cada uno

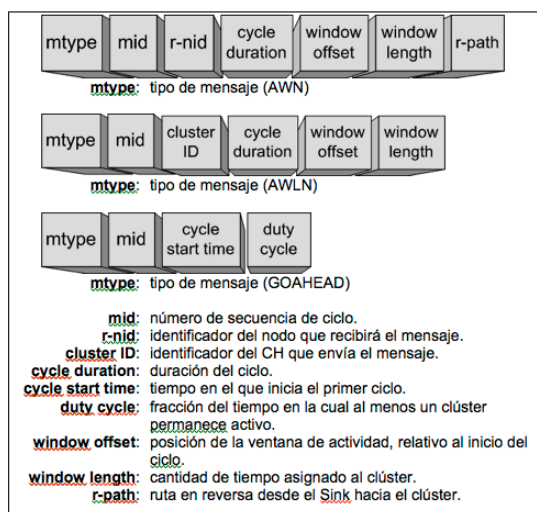


Figura 10. Formato de paquete, notificación de ventanas de actividad.

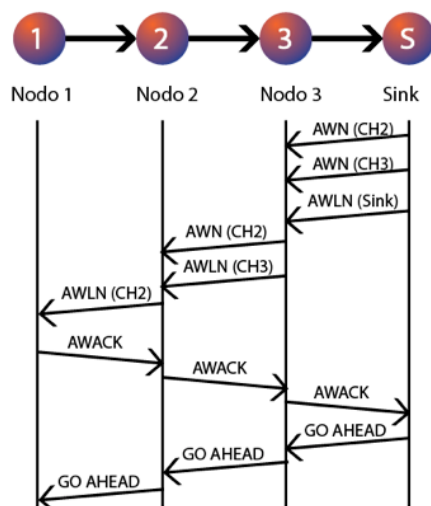


Figura 11. Resumen de notificación de ventanas de actividad.

de los miembros de su clúster.

Los nodos hoja acusarán de recibido el paquete AWLN mediante un paquete AWACK. Los nodos CH esperarán a recibir todos los acuses de recibo AWACK de sus nodos hijos y el AWLN de su cabeza de clúster para enviar su propio paquete AWACK a su CH. Este proceso continúa hasta que eventualmente el Sink recibe todos los acuses de recibo AWACK correspondientes a los miembros de su propio clúster. La figura 11 muestra en forma resumida la fase de notificación de ventanas de actividad.

El sistema está ahora listo para iniciar transmisiones en un ambiente libre de colisiones, con períodos seguros de dormir y con garantías de QoS. Para iniciar este período, el Sink difunde un paquete llamado GOAHEAD que inunda la red informando a cada nodo. Este paquete contiene el tiempo en el que se iniciará el primer ciclo de transmisiones de datos, como se muestra en la figura 10, el cual usarán los nodos para calcular sus tiempos de despertar y dormir. Además se incluye el tiempo de duración del ciclo, de modo que los nodos conocerán la fracción del tiempo de ciclo en la cual permanecerán dormidos todos los nodos, lo cual es útil al momento de reconfigurar la red, como será explicado en el apartado III.7.

Hasta este punto, las comunicaciones entre nodos se dan mediante un protocolo de acceso al medio (MAC) basado en contienda, tal como CSMA/CA en las versiones de los estándares IEEE 802.15.4 o 802.11. Si se emplea 802.11 (DCF) es conveniente forzar a los nodos a realizar una retracción antes de cada transmisión, aún cuando el canal esté desocupado inicialmente, para reducir colisiones.

### **III. 5 Fase de operación normal.**

Durante la fase de operación normal, los únicos nodos activos serán aquellos miembros de clústeres a los que corresponda el tiempo de sus ventanas de actividad. El acceso al canal ya no es mediante contienda, sino que los nodos miembros de un clúster, deben esperar a recibir una encuesta de su CH indicando que tienen oportunidad de transmitir sus datos. Esta encuesta permitirá la transmisión de una trama a la vez, o de todas las tramas correspondientes a un paquete de capas superiores (MSDU), o tantas tramas como sean posibles durante un tiempo pre-especificado (TXOP), proporcional a la cantidad de ancho de banda solicitado durante el proceso de reservación. Para nuestras simulaciones adoptamos el mecanismo de encuesta TXOP.

Un nodo que reciba una encuesta responderá con una trama de datos si tiene información para enviar, indicando con los bits *“More\_Frag”* y *“More\_Data”* si tiene más tramas del mismo paquete o más paquetes para reenviar respectivamente. Si el nodo ya no tiene información para enviar responderá con una trama nula. De cualquier manera, el CH puede detectar cuando un nodo ha terminado de enviar sus datos y deja de enviarle encuestas durante la ventana de actividad de ese ciclo. En este momento, ese nodo puede irse a dormir ahorrando energía. Si todos los nodos se van a dormir antes de que termine la ventana de actividad, el CH puede hacer lo mismo. Si un nodo falla repetidamente en responder la encuesta de su CH, se genera un mensaje de error en el CH y se envía al Sink para alertar de esta falla.

### **III. 6 Sincronización de la red.**

La sincronización de nodos en protocolos de acceso basados en compartición de tiempo, tales como el presentado en esta tesis, es muy importante para prevenir



traslapes en las ventanas de actividad, debidos a desajustes de reloj. Para lograr la sincronización de la red se pueden adoptar, en un principio, mecanismos como los descritos en Mills (1991) o en Sundararaman et al (2005). Una vez que los nodos identifiquen un nodo padre, éstos pueden intercambiar mensajes de sincronización. Esto causará que eventualmente todos los nodos tengan sincronización con el reloj del Sink. Después del proceso de reservación, cada nodo puede intercambiar mensajes de sincronización con sus CH. Este proceso puede continuar durante la fase de transmisión de datos para evitar desajustes considerables. Para esto se pueden emplear tramas de *encuesta*, *datos* y *nulas* para anunciar estampas de tiempo que sincronicen a los nodos con sus CH.

### **III. 7 Reconfiguración y recuperación tras fallas de nodos.**

Ser cabeza de clúster puede ser muy desgastante en cuestiones de energía. Esto debido a que el CH tiene que despertar durante dos ventanas de actividad, una para recolectar datos de sus hijos y otra para transmitirlos, además de incluir los datos propios generados. Es por esto que los nodos deben alternarse el rol de CH a través de repetir periódicamente el proceso de configuración. La redistribución del consumo de energía recae en el hecho de que los nodos seleccionan rutas basándose, entre otros aspectos, en la energía disponible en el nodo más débil de la ruta, como se explica en los subcapítulos III.1 y III.2. Si solo existe un Sink, la redistribución de la energía podría no ser tan efectiva como se desearía, dado que aquellos nodos más cercanos al Sink siempre tendrán que retransmitir más tráfico que aquellos más alejados. Por lo tanto, se recomienda tener varios Sink y alternar sus actividades.

Con el fin de iniciar un nuevo procedimiento de configuración, el Sink actual difundirá un paquete RCWK (despertar), después de cierto tiempo trabajando en fase de operación normal. Este paquete RCWK será difundido repetidamente por

los CH, al inicio de cada ciclo subsecuente, con el propósito de alcanzar eventualmente al siguiente Sink que estará activo. Cuando el siguiente Sink recibe este mensaje RCWK, enviará inmediatamente un paquete RCWRP (respuesta de despertar) para confirmar que está listo para entrar en actividad. Cada CH que escuche este paquete RCWRP lo reenviará hasta que alcance al Sink activo. Está claro que los Sinks que permanecen dormidos, deberán encender sus receptores periódicamente para ser capaces de escuchar los mensajes de despertar. Es recomendable, de igual forma, que estos nodos Sink dormidos se mantengan escuchando promiscuamente paquetes de otro Sink activo para conservar la sincronización con el mismo.

Si después de un tiempo de haber enviado un paquete de despertar RCWK, no se recibe un paquete de respuesta RCWRP, el Sink activo intentará despertar a otro Sink o, si ninguno responde, el mismo se mantendrá activo durante otro ciclo de trabajo. El siguiente paso es que el nuevo Sink activo difunda un paquete RCN (notificación de reconfiguración), indicando que se iniciará de nuevo una etapa de configuración. Cuando un CH recibe un paquete RCN éste le indica que debe detener las retransmisiones de paquetes RCWK. Si solo hay un Sink en toda la red, no se usarán los paquetes RCWK y RCWRP, sino que el Sink solamente enviará un paquete RCN, indicando una nueva fase de configuración.

Cada CH reenviará a los miembros de su clúster (durante su ventana de actividad correspondiente), los paquetes RCN recibidos, anunciando la siguiente fase de configuración. Estos miembros a su vez, harán lo mismo si son CH de otro clúster, y así sucesivamente hasta que todos los nodos de la red tengan copia del paquete RCN. Este paquete contendrá un campo llamado "*time-to-setup*", como se muestra en la Figura 12, el cual indicará el número de ciclos restantes hasta iniciar el proceso de configuración.

Este campo es llenado inicialmente por el Sink con su valor de profundidad. Cada CH reducirá este valor de uno en uno antes de retransmitirlo, de modo que los mensajes de configuración se empezarán a transmitir un ciclo después de que el nodo hoja más alejado reciba el paquete RCN. En ese momento cada nodo en el sistema estará al tanto y listo para iniciar el nuevo proceso de configuración, ya sea permaneciendo despierto durante el tiempo en el que el sistema solía estar dormido o deteniendo temporalmente la fase de transmisión de datos, dependiendo de si el campo *reconfig\_type* contiene el valor *normal* o *exclusive*. En otras palabras, en una reconfiguración normal los nodos pueden evitar el detener la transmisión de datos para reconfigurarse usando la porción de tiempo de dormir de cada ciclo, suponiendo que la duración de ciclo es suficientemente menor que 1. Si este es el caso, los nodos continuarán transmitiendo información al Sink activo anteriormente hasta un instante en el que sea tiempo de enviar un mensaje AWACK, en el que se acusa de recibido a la nueva ventana de actividad a usar con el nuevo Sink. Si el tiempo de duración de un ciclo está muy cercano a 1, se necesitará una reconfiguración exclusiva, en la cual los nodos tendrán que detener la transmisión de datos cuando el campo *time-to-setup* llegue a cero en las retransmisiones del paquete RCN iniciando el nuevo proceso de configuración.

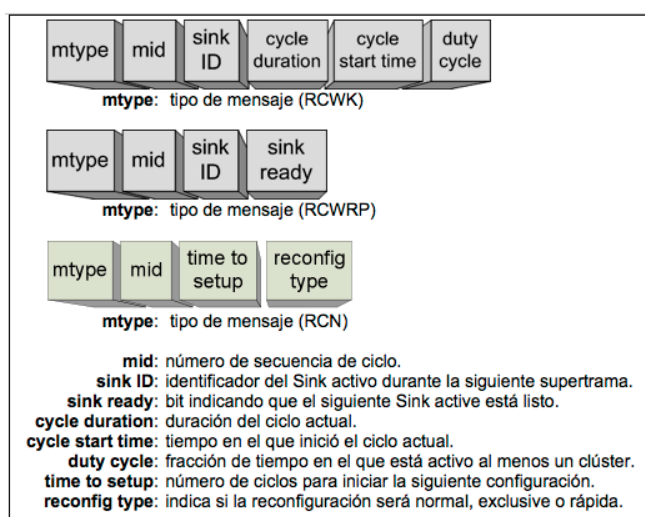


Figura 12. Formato de paquete, reconfiguración.

El proceso de reconfiguración puede ser iniciado también si se detecta un problema severo en el sistema, tal como la pérdida de un CH, la cual aislaría a un conjunto de nodos. Si esto pasa, se generará un mensaje de error por el nodo que ha detectado el problema; este será reenviado hasta que llegue al Sink, como se mencionó en la sección III.5. Dependiendo de las dimensiones del daño (eg. el número de nodos que han sido desconectados tras la falla) el Sink puede decidir entre si es necesario realizar un procedimiento completo de configuración o sólo una recuperación rápida. Esto será indicado de nuevo en el campo *reconf\_type* del paquete RCN transmitido por el Sink. En una reconfiguración rápida los nodos no realizan un nuevo descubrimiento de rutas y asignación de pesos, descrito en la sección III.1, sino que usan la información recolectada durante el último proceso de configuración e inician directamente el proceso de reservación descrito en la sección III.2. De hecho, aquellos nodos que no han sido desconectados tras la falla, podrían enviar su solicitud de reservación RSRQ al mismo nodo CH que tenían anteriormente, el cual tendría mayor probabilidad de aceptar dicha reservación, de modo que se ahorra tiempo y energía en este proceso. Dependiendo de si el tipo de reconfiguración es completo o rápido, puede ser realizado usando la porción de tiempo dormido de la red (normal) o deteniendo la transmisión de datos para usar todo el ancho de banda (exclusivo).

Un nodo tendrá conocimiento de si ha quedado aislado del resto de la red si no se reciben encuestas de su CH durante algunos ciclos consecutivos. Cuando esto pase, si el nodo en cuestión es a su vez un CH, detendrá la transmisión de encuestas a sus miembros de clúster, indicándoles que sus datos ya no pueden ser reenviados hasta el Sink. Para permitir la posibilidad de ser reconectado al resto de la red, los nodos aislados permanecerán despiertos, (durante el tiempo que el sistema está dormido, si el ciclo de trabajo es suficientemente menor a 1, o constantemente, si no) esperando por el mensaje RCN que iniciará un proceso de reconfiguración, permitiéndoles seleccionar un nuevo CH.

### **III. 8 Aplicabilidad de la propuesta.**

Lo siguiente es un escenario típico en el que el protocolo propuesto podría ser de gran utilidad. Considerando un sistema de vigilancia, ya sea para casa u oficina, edificios comerciales o de gobierno. Además considerando nodos sensores que están equipados con al menos dos tipos de sensores, uno que detecta la presencia de intrusos (sensores de movimiento) y otro que recolecte video una vez que se identifica una actividad sospechosa. Los datos recolectados por los sensores de movimientos podrían no ser transmitidos continuamente hacia el Sink, pero pueden ser usados localmente para activar los sensores de video y enviar esa información al Sink. Si algún intruso entra al edificio, la actividad sospechosa será detectada por varios sensores localizados en ciertas regiones comunes. Aún cuando no todos los sensores se activen al mismo tiempo, el hecho de que un conjunto de nodos vecinos se activen al mismo tiempo hace necesaria y pertinente la reservación de recursos, como la presentada en esta propuesta.

Un escenario similar podría ser un sistema de vigilancia para un campo de batalla. Los nodos pueden estar equipados con sensores que detecten la presencia de fuerzas enemigas, los cuales una vez identificadas dichas fuerzas, iniciarán la recolección de video para que el personal militar pueda analizarlo en el Sink en tiempo real, estimando el número de tropas enemigas y el tipo de equipamiento que poseen.

Nuestra propuesta puede trabajar con diferentes tipos de tráfico. Si se conoce de antemano, la máxima cantidad de datos que puede ser generado por un nodo durante un ciclo, tal como en una fuente de tasa de bit constante o una cubeta agujerada limitada, Gallardo et al (2007), se pueden ofrecer garantías determinísticas (duras) de QoS. Si por otro lado, el tráfico presenta alta variabilidad, de modo que no es práctico reservar recursos para procesar la

máxima cantidad de datos que puede ser generada en un ciclo, entonces se pueden ofrecer garantías estocásticas (suaves), de modo que se pueden violar algunos requerimientos de QoS pero con una probabilidad muy baja. El diseño de un método para calcular la cantidad necesaria de ancho de banda por reservar, con el fin de garantizar cotas máximas de violaciones a garantías de QoS, está fuera del alcance de este trabajo.

## Capítulo IV

---

### Simulación y Resultados.

---

En este capítulo, se presenta el modelo de simulación de la propuesta empleando OPNET, así como los cambios pertinentes a los modelos empleados, de modo que se facilite la reproducción de estas simulaciones, y finalmente se presentan los resultados obtenidos de las mismas.

#### IV.1 Modelo de simulación.

Se evaluó el desempeño de la propuesta mediante simulaciones, empleando la versión 14.0 de la herramienta OPNET (OPNET Technologies, Inc, 2009), sus características serán descritas brevemente en esta sección. La figura 13 muestra un ejemplo de la topología empleada.

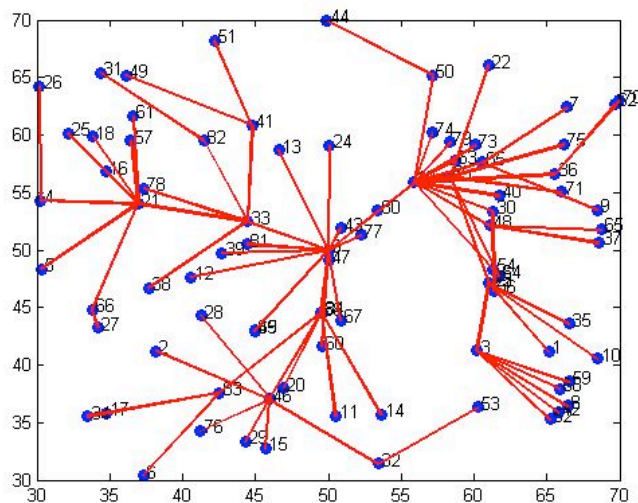


Figura 13. WSN con despliegue aleatorio y fase de configuración terminada.

Se consideró una capa física compatible con el estándar IEEE 802.11 (CSMA/CA) con una tasa de transmisión de 1Mbps. El rango de transmisión de los nodos fue supuesto de 10m. Los paquetes de configuración constan de 100Bits y los paquetes de datos de 1Kbit. Se evaluó la propuesta sobre redes de nodos sensores con posiciones aleatorias, contando con un nodo sumidero al cual llamaremos "Sink" posicionado siempre en el centro de la topología de red. El nodo Sink se encargó de recopilar la información de todos los nodos de la red.

La Figura 14 muestra el modelo de nodo empleado por la propuesta. Este modelo se divide en tres secciones en función de sus tareas:

- Configuración.
  - *Módulo de cola, "configuration"*: Este módulo aloja el modelo de proceso encargado de generar y procesar los paquetes de configuración, ya sean descubrimiento de rutas, asignación de pesos, reservación de recursos, recolección de información de interferentes y cálculo de ventanas de actividad. Sería la capa de aplicación del proceso de configuración.
  - *Módulo de proceso, "wlan\_mac\_intf"*: Este módulo aloja el modelo de proceso encargado de agregar información de control (dirección destino, control de transmisión por difusión o punto a punto) a los paquetes generados por la capa de aplicación, para ser enviados posteriormente a la capa MAC, y de recibir paquetes de la misma MAC y enviarlos a la capa de aplicación.
  - *Módulo de proceso, "wireless\_lan\_mac"*: Este módulo se encarga de invocar el modelo MAC nativo de OPNET, el cual procesa el estado de las transmisiones, colisiones, recepción de paquetes, retracciones entre otras tareas definidas por el estándar 802.11.
- Transmisión de datos en ventanas de actividad.



- Módulo “*schedl\_mac*”: Este módulo aloja las tablas de enrutamiento y ventanas de actividad generadas por el proceso de configuración. Despierta al nodo en su ventana de actividad encuestando a los nodos hijos, si se trata de un CH, o permanece en escucha y espera de una encuesta, si es el tiempo de despertar como nodo miembro de un clúster. Controla la cola de datos generados por la aplicación.
- Módulo “*source*”: Este módulo genera el tráfico deseado de la capa de aplicación, insertándolo en la cola del módulo SCHDL\_MAC para su transmisión una vez que se encuentre dentro de su ventana de actividad.
- Módulo “*sink*”: Este módulo genera las estadísticas correspondientes en función de los paquetes recibidos y enviados.
- Selección del tipo de MAC en función de la fase actual del protocolo.
  - Módulo “*wlan\_port\_rx/tx*”: Estos dos módulos son las interfaces inalámbricas del nodo. Su trabajo es invocar los procesos de ejecución de instrucciones necesarios para imitar de la manera más realista posible el comportamiento del canal inalámbrico en cada transmisión. Entre sus tareas están el realizar las modulaciones pertinentes, análisis de relación señal a ruido, cargar los modelos de terreno, grupo de receptores, resoluciones de enlaces, entre otras tareas referentes a la capa física. La Figura 15 muestra la secuencia de ejecución de instrucciones de un radio transceptor en una transmisión, invocadas por este módulo.

- Módulo “MAC\_SWITCH”: Este módulo se encarga de conmutar la conexión a la interfaz inalámbrica de un mecanismo MAC al otro, conectando inicialmente la interfaz inalámbrica con el mecanismo MAC de configuración por contienda, y posteriormente cambiar al mecanismo basado en calendarización, una vez que la configuración ha terminado y se procede a transmitir paquetes de datos en la red.

El módulo MAC nativo de OPNET, invocado por el módulo de proceso wireless\_lan\_mac, fue modificado para permitir a los nodos escuchar ociosamente los paquetes transmitidos en su vecindario, de modo que sean capaces de identificar posibles nodos interferentes.

El proceso de ejecución de instrucciones número dos, mostrado en la Figura 15, está encargado de la resolución de si un enlace entre dos nodos es o no válido, y tiene un valor por omisión que le permite a todos los nodos estar al alcance uno del otro. Este valor por omisión fue modificado para darle a cada nodo un radio de

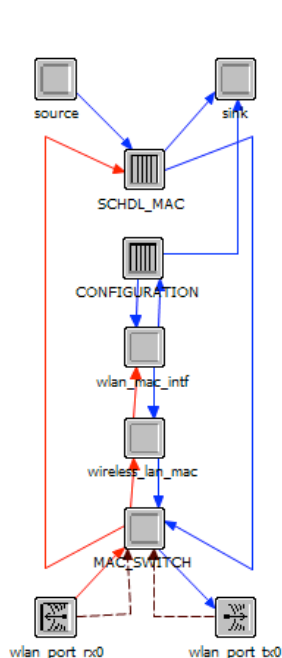


Figura 14. Modelo de Nodo.

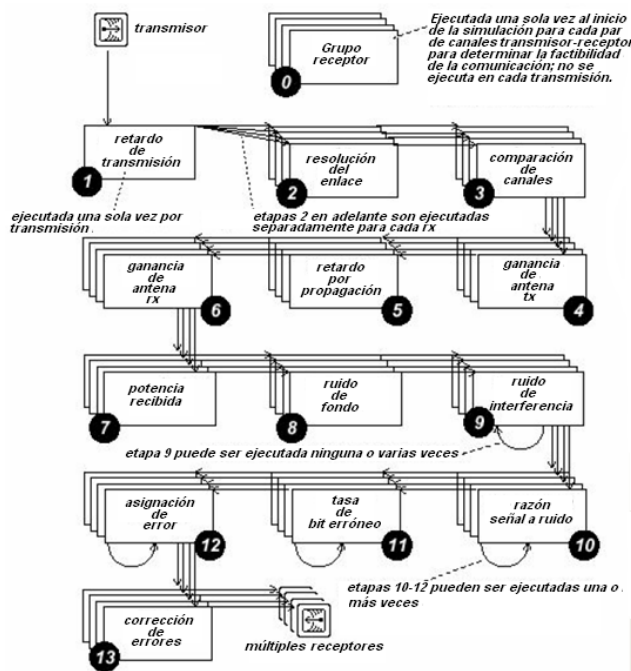


Figura 15. Secuencia de ejecución de “pipelines”

cobertura de 10m, permitiendo la simulación de redes con múltiples saltos.

## IV.2 Evaluación de fase de configuración.

La primera evaluación se realizó sobre la fase de configuración del protocolo. Esta evaluación puede ser considerada como el precio a pagar en términos de sobre encabezado, tiempo de configuración y energía consumida, con el fin de lograr una red con garantías de QoS sin colisiones y con el mínimo tiempo de escuchas ociosas, lo que se resume en un alto nivel de ahorro de energía.

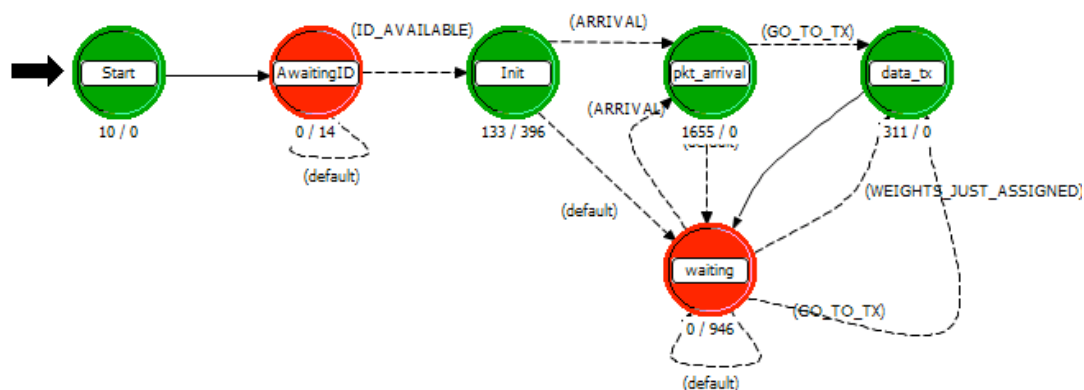


Figura 16. Modelo de proceso. Etapa de configuración.

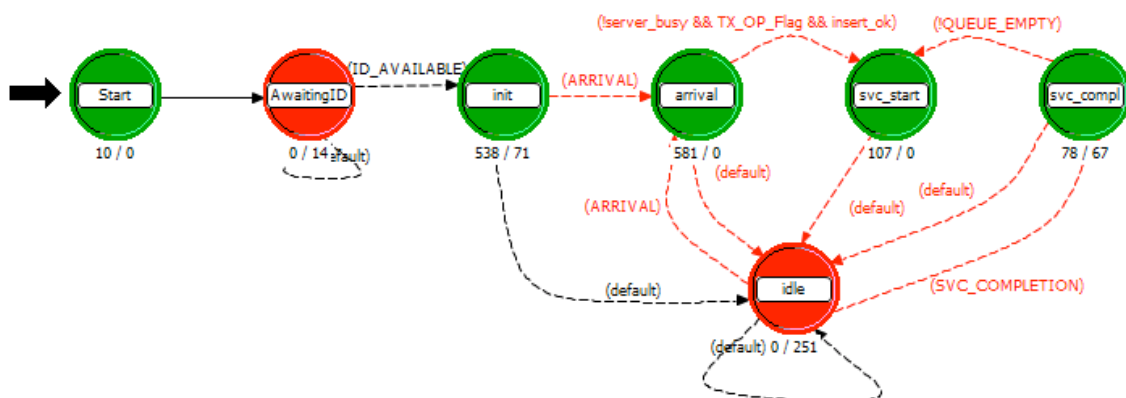


Figura 17. Modelo de proceso. Etapa de transmisión de datos.

OPNet simula un nodo en base a una máquina de estados. Las Figuras 16 y 17 muestran el modelo empleado durante la etapa de configuración y transmisión de datos, respectivamente.

La fase de configuración, como fue descrita anteriormente, consiste en el descubrimiento del vecindario, de las rutas disponibles y la reservación de recursos por una de estas rutas desde cada nodo sensor hacia el Sink.

Existen dos factores clave que afectan la fase de configuración: la densidad de nodos y el máximo número de saltos hacia el nodo Sink, considerando el nodo más alejado del mismo. Debido a lo anterior se realizaron dos conjuntos de simulaciones para evaluar la fase de configuración:

1. Una en la que el número de nodos es fijo a 100 y el área total de despliegue de nodos es variable para lograr diferentes densidades de nodos.
2. Otra en la que el área de despliegue de nodos es fija a  $25 \times 25 \text{m}^2$  y el número de nodos varía para lograr las diferentes densidades de nodos deseadas.

El objetivo en ambos conjuntos de simulaciones es analizar el sistema bajo diferentes densidades de nodos. La diferencia entre ellos, radica en el hecho de que para aumentar la densidad de nodos en el primer conjunto de simulaciones, se reduce el área de despliegue de nodos, reduciendo eventualmente el máximo número de saltos hacia el Sink a un solo salto, lo que no pasa en el segundo conjunto de simulaciones. En el segundo conjunto de simulaciones el número máximo de saltos hacia el Sink se mantiene constante, lo que varía es el número de nodos en la red.

Se corrieron diez simulaciones por cada una de las nueve densidades de nodos abordadas, cada una con semilla diferente para posicionamiento aleatorio de

nodos. Los resultados que se muestran corresponden a los valores promedio con un intervalo de confianza de 95%. Los resultados se encuentran en función de la densidad de nodos por área de cobertura ( $n/Ac$ ).

Con respecto al primer conjunto de simulaciones, la Figura 18 muestra el tiempo necesario para descubrir rutas, asignar pesos a esas rutas y negociar todas las reservaciones de la red. La menor densidad de nodos por área de cobertura corresponde a un área de despliegue de  $65 \times 65 m^2$ , mientras que la mayor densidad corresponde a un área de  $18 \times 18 m^2$ . Sobre los mismos escenarios mostrados en la Figura 18, la Figura 19 muestra el número promedio por nodo de mensajes de control, necesarios para lograr el mismo descubrimiento de rutas, asignación de pesos y reservación de recursos.

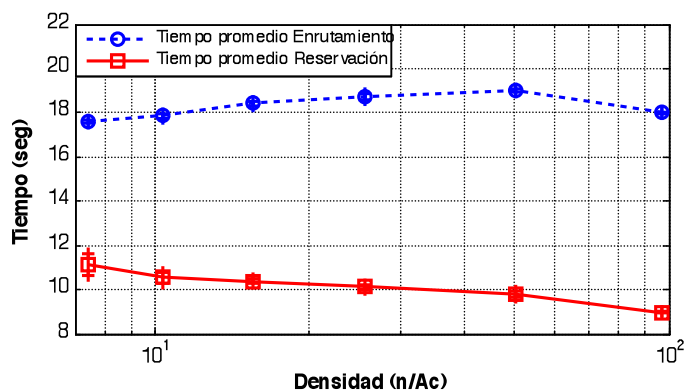


Figura 18. Tiempo de configuración, escenario: área variable.

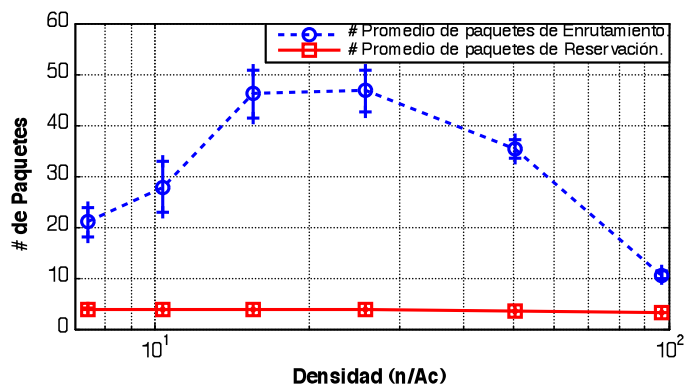


Figura 19. Número de paquetes de configuración, escenario: área variable.

Tanto en la Figura 18 como en la Figura 19 podemos ver que el tiempo de configuración y la cantidad de paquetes necesarios para la misma va en incremento conforme la densidad de la red aumenta, hasta llegar a cierto punto en el que ambos parámetros decaen conforme sigue aumentando la densidad de la red. Este comportamiento se debe al hecho de que cuando el área total de despliegue se reduce por debajo del área de cobertura de los nodos, la red deja de ser de múltiples saltos y se convierte en una red donde todos los nodos se localizan a un salto del Sink; esto simplifica enormemente el proceso de configuración del protocolo, reduciendo tiempo y paquetes necesarios para la misma.

En el segundo conjunto de simulaciones, las menores y mayores densidades

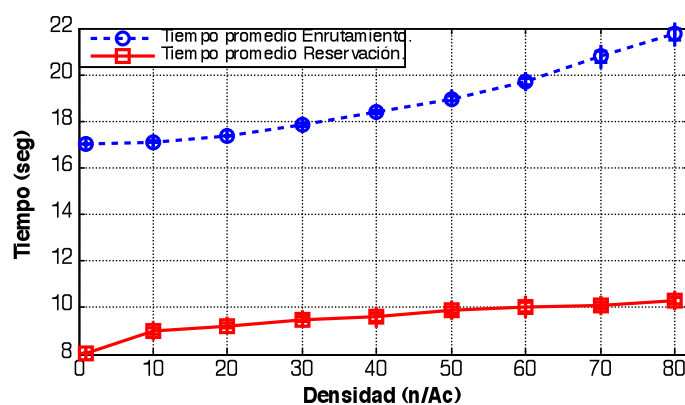


Figura 20. Tiempo de configuración, escenario: área fija.

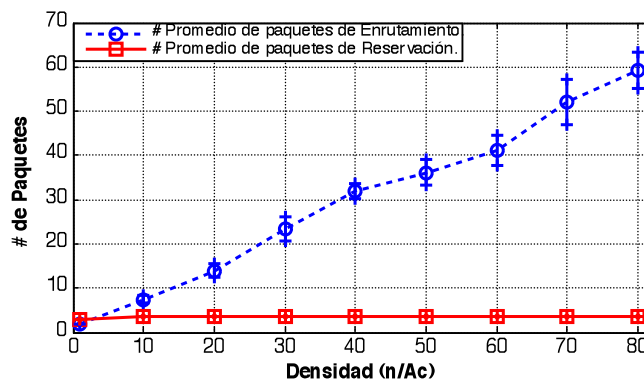


Figura 21. Número de paquetes de configuración, escenario: área fija.

corresponden a 2 y 159 nodos, desplegados en un área de  $25 \times 25 \text{m}^2$  para obtener densidades de 1 a 80 n/Ac. Al tener un área de cobertura por nodo de  $10 \text{m}^2$ , y el Sink posicionado en el centro de la topología, los nodos se encuentran desplegados en uno y dos saltos hacia el Sink. Las Figuras 20 y 21 muestran que para este segundo conjunto de simulaciones, el tiempo y número de paquetes de configuración se mantiene en aumento conforme la densidad aumenta, sin presentar el decaimiento a altas densidades, (observado en el primer conjunto de simulaciones), debido a que esta vez se mantiene el mismo número de saltos hacia el Sink.

En este segundo conjunto de simulaciones podemos observar que el tiempo necesario para la fase de configuración es menor a un minuto. Además se puede ver que el precio a pagar en términos de número de paquetes de configuración es alrededor de 70 para el peor caso, cuando la densidad de nodos es mayor. El sobre encabezado incurrido por la fase de configuración de esta propuesta es bastante razonable en términos de tiempo y número de paquetes, considerando que con esto se logra un sistema de transmisión de datos libre de colisiones y menores escuchas ociosas.

En la Figura 22 podemos observar la energía consumida por el proceso de configuración para el segundo conjunto de simulaciones, basándonos en el modelo de energía Atheros para productos WLAN (Atheros white paper 2003). El consumo de la capa física WLAN usado es el siguiente: 2W transmisión, 0.9W recepción y 0.8W escucha ociosa. Estos parámetros se incluyen con el objetivo de tener una referencia cualitativa (aunque los valores cuantitativos puedan cambiar de aplicación en aplicación) y para propósitos de comparación con el protocolo de referencia.

### IV.3 Desempeño durante la transmisión de datos.

Para evaluar el desempeño de la propuesta durante la transmisión de datos se reprodujo el segundo conjunto de simulaciones presentado en la sección IV.1.

Los paquetes de datos empleados tienen un tamaño de 1Kbit. Para los nodos en el estado de censado, se considera una fuente de datos “leaky bucket constrained”, con una tasa máxima de ráfaga de 4 paquetes/seg. Esto nos lleva a que cada nodo genera 4Kb/s en el peor de los casos. En la etapa de configuración para este escenario, se reservaron recursos para el peor de los casos con el objetivo de proporcionar garantías duras de QoS. Se consideró un requerimiento de retardo de 0.25s por paquete. Cada paquete contiene una estampa de tiempo cuando es generado que permite el cálculo del retraso promedio. Los escenarios simulados permanecen transmitiendo datos por 60 segundos hacia el Sink. Se corrieron diez simulaciones por cada densidad de red, contando con ocho densidades diferentes, que van desde 1 hasta 70n/Ac. Se calcularon valores promedio de caudal eficaz, retraso y consumo de energía con un intervalo de confianza del 95%.

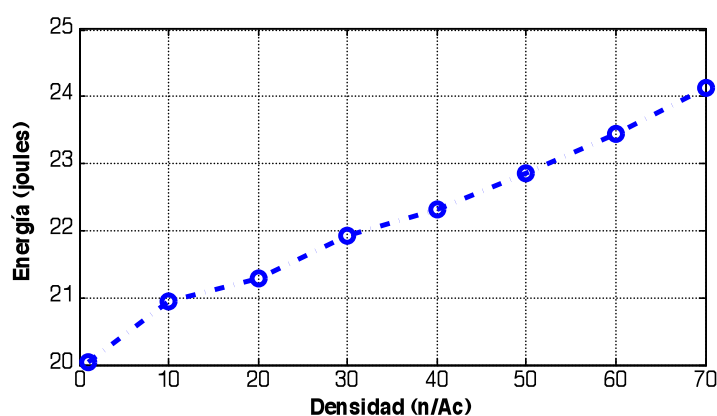


Figura 22. Energía consumida durante el proceso de configuración, escenario: área fija.



Esta propuesta se comparó con el protocolo de acceso al medio para WSN, SMAC, debido a que igual que nosotros, su principal objetivo es ahorrar energía mediante ciclos calendarizados de actividad y períodos largos de nodos dormidos evitando escucha ociosa (Wei Ye, et al. 2004). Para dicha comparación se emplearon exactamente los mismos escenarios, tasas de transmisión y rutas encontradas hacia el Sink. Se definió una duración variable del ciclo de trabajo para SMAC, la cual inició con un 10% y se incrementó conforme se tenían más paquetes a transmitir o se seguían escuchando paquetes en el vecindario. SMAC, al ser un protocolo basado en contienda, define una cota máxima de intentos de retransmisión de un paquete tras una transmisión fallida. En las simulaciones presentadas se consideraron 7 y 255 como cotas máximas de intentos de retransmisión para SMAC.

Se emplearon las siguientes métricas de evaluación y comparación de desempeño:

- *Retardo promedio de paquete*: Diferencia de tiempo desde que el paquete fue creado en la fuente del nodo sensor hasta llegar al Sink.
- *Energía promedio consumida por nodo*: Energía promedio consumida por un nodo durante el tiempo de simulación.
- *Caudal eficaz promedio*: Razón promedio porcentual de paquetes recibidos por el Sink entre el número de paquetes transmitidos por nodo durante el tiempo de simulación.
- *Porcentaje promedio de ciclo encendido*: Razón promedio porcentual de tiempo que los nodos permanecieron encendidos en cualquiera de las funciones de la red (transmitir, recibir o escuchar), entre el tiempo total de simulación.

La Figura 23 muestra el caudal eficaz promedio para diferentes densidades, tanto de la propuesta (QUATTRO) como del modelo de referencia (SMAC), empleando los dos límites de retransmisión antes mencionados para este último. Se puede

observar que SMAC logra un 100% del caudal eficaz sólo cuando se tiene 255 como límite de retransmisiones y antes de llegar a una densidad de 20n/Ac. En cambio, QUATTRO logra un 100% del caudal eficaz sobre todas las densidades de red. Esto es debido a que QUATTRO reserva recursos y calendariza transmisiones sin interferencias para asegurar ese caudal eficaz completo.

En la Figura 24 se presenta el retraso promedio de paquetes para la propuesta y el modelo de referencia. SMAC alcanza los 12 y 2 segundos de retraso para 255 y 7 como límite de retransmisión respectivamente, mientras que QUATTRO mantiene un retraso máximo de 0.23 segundos, el cual está por debajo del requerimiento de la red (0.25s).

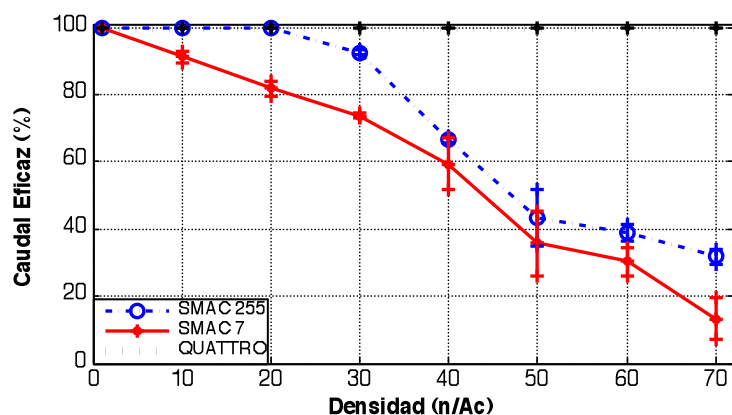


Figura 23. Caudal eficaz promedio, escenario: área fija.

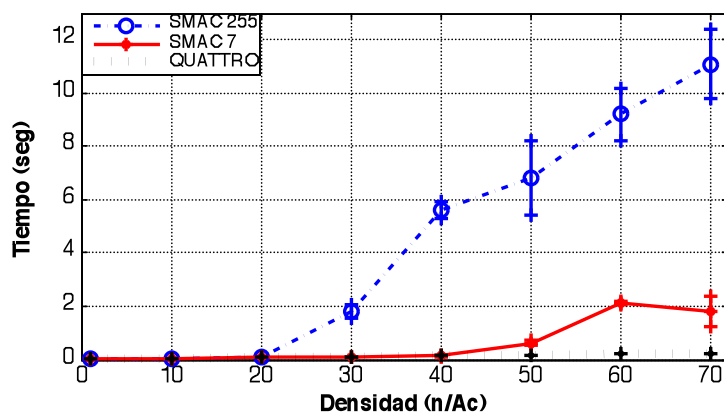


Figura 24. Retraso promedio, escenario: área fija.

Con la información proporcionada por las Figuras 23 y 24, y considerando que se genera un nuevo paquete cada 0.25s, SMAC es capaz de entregar todo el tráfico generado al Sink sólo cuando emplea 255 como límite de retransmisión y con densidades de red por debajo de 20n/Ac. QUATTRO, en cambio, es capaz de mantener un retraso debajo del requerido por la red (0.25s) con un caudal eficaz completo (100%) en todas las densidades por área de cobertura simuladas (Redes desde 2 hasta 140 nodos).

La Figura 25 muestra la energía consumida por SMAC y QUATTRO al final de cada simulación durante la transmisión de datos. SMAC aparentemente converge a cerca de 80 Joules con densidades mayores a 30n/Ac, en contraste con el constante incremento de consumo presente en QUATTRO. Este comportamiento convergente es debido a que SMAC termina descartando paquetes después de los intentos de retransmisión no exitosa. SMAC degrada el caudal eficaz conforme se incrementa la densidad de la red, como vimos en la figura 23, lo que produce un consumo promedio de energía casi constante a partir de 30n/Ac; mientras que QUATTRO incrementa su consumo de energía conforme incrementa la densidad de la red, debido a que no hay descarte de paquetes. A pesar de que QUATTRO mantiene el total de paquetes dentro de la red consumiendo energía, está por debajo del presentado por SMAC para todas las densidades simuladas.

El promedio del porcentaje de ciclo encendido para la propuesta y el modelo de referencia se presentan en la Figura 26. Las colisiones e interferencias ocurridas en escenarios a partir de 20n/Ac para SMAC, mantienen a los nodos escuchando paquetes y colisiones constantemente. Dado que los nodos con SMAC no pueden saber si habrá más paquetes para reenviar en el presente ciclo, necesitan mantenerse encendidos mientras sigan escuchando tráfico en el vecindario, es por esto que su ciclo activo se ve incrementado, alcanzando casi el 100% a partir de 20n/Ac.

En el caso de QUATTRO, el ciclo activo presenta un incremento lineal, proporcional al tráfico generado por la inserción de nuevos nodos para incrementar la densidad de la red, sin llegar al 80% en la mayor densidad analizada. Esto se debe a que QUATTRO administra todas las transmisiones en un calendario, permitiendo a los nodos dormir por determinado tiempo, evitando colisiones e interferentes, manteniendo las garantías de QoS y ahorrando energía.

#### IV.4 Escenario “ROW”.

Un esquema basado en calendarización, como lo es QUATTRO, tiene la característica principal de controlar las oportunidades de transmisión, lo que permite comunicarse entre sí solo a dos nodos en el mismo vecindario, al mismo tiempo. Al tener una estructura topológica de árbol, los nodos padre deben esperar

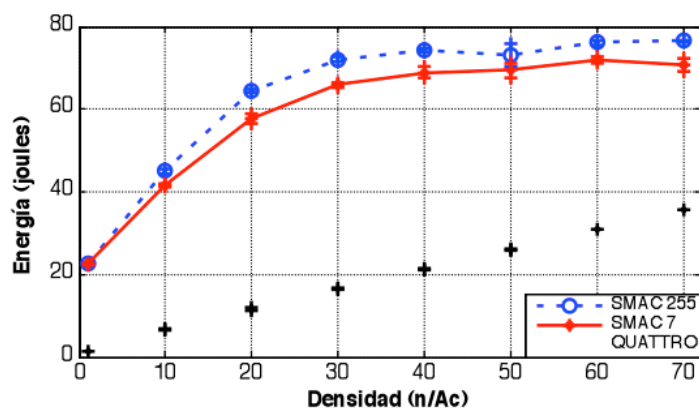


Figura 25. Energía promedio, escenario: área fija.

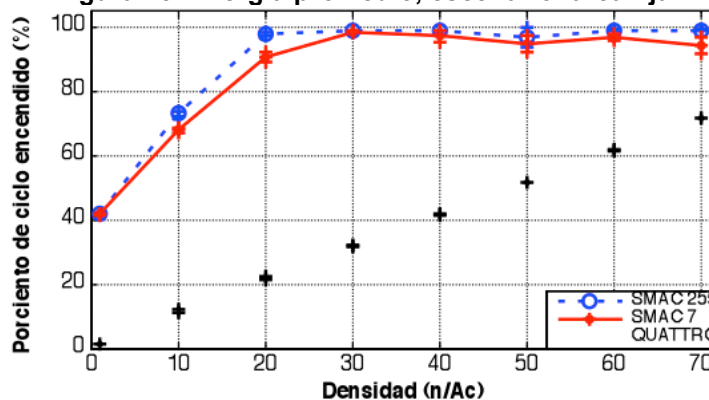
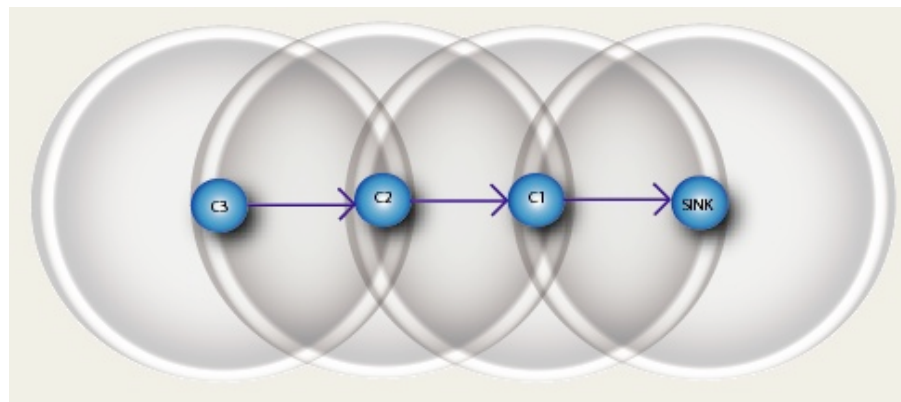


Figura 26. Porcentaje de ciclo encendido, escenario: área fija.

a recibir toda la información de sus nodos hijo, para posteriormente reenviarla junto con los datos propios hacia el siguiente nodo en ruta hacia el Sink. Esta característica es tomada en cuenta al momento de calcular las ventanas de actividad en QUATTRO, lo que nos lleva a abordar el escenario presentado en este apartado. Hemos llamado escenario "ROW" a una topología lineal en el que los nodos tienen conectividad solo con otros dos nodos, su hijo y su padre. Para evaluar el desempeño de la propuesta y el modelo de referencia sobre este escenario, se corrieron diez simulaciones con nueve diferentes números de nodos, donde cada nuevo nodo en la red se colocó un salto más alejado del Sink, teniendo conectividad con sólo un nodo en la red como se puede ver en la Figura 27. El tamaño de la red fue de 2 a 18 nodos con incrementos de 2.

La Figura 28 muestra el caudal eficaz para SMAC con 255 y 7 como límites de retracción, así como para QUATTRO en el escenario "ROW". Se puede ver que ambos mecanismos alcanzan el 100% del caudal eficaz en todas las densidades simuladas. SMAC es capaz de lograr un caudal eficaz completo al igual que QUATTRO, debido a que este escenario presenta bajas interferencias, ya que cada nodo solo escucha a su nodo padre y a su nodo hijo.



**Figura 27. Escenario "ROW".**

Como se puede ver en la Figura 29, QUATTRO presenta mayores tiempos de retraso que SMAC conforme se agregan nodos a la topología lineal. QUATTRO calendariza no solo transmisiones no interferentes, sino que además son secuenciales sobre cada rama de ruta. Esta característica hace que las transmisiones de datos de un ciclo se inicien en los nodos fin de ruta. Cada nodo en la ruta espera a recibir toda la información de sus hijos para posteriormente transmitirla a su cabeza de clúster. Ningún nodo reinicia transmisiones hasta que la información completa de la red (generada en un ciclo) llega al Sink, ocasionando el retraso que se observa en la Figura 29 para QUATTRO. SMAC, en cambio, permite a los nodos intentar transmitir inmediatamente después de tener paquetes en la cola de servicio, ya sea tráfico propio o de los hijos. Debido a lo

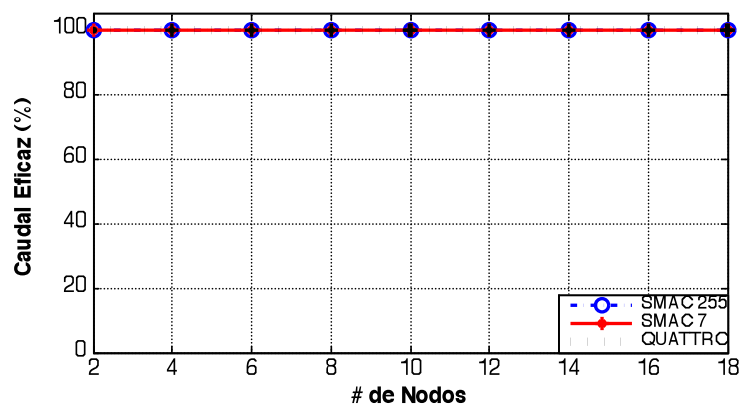


Figura 28. Caudal eficaz promedio, escenario: ROW

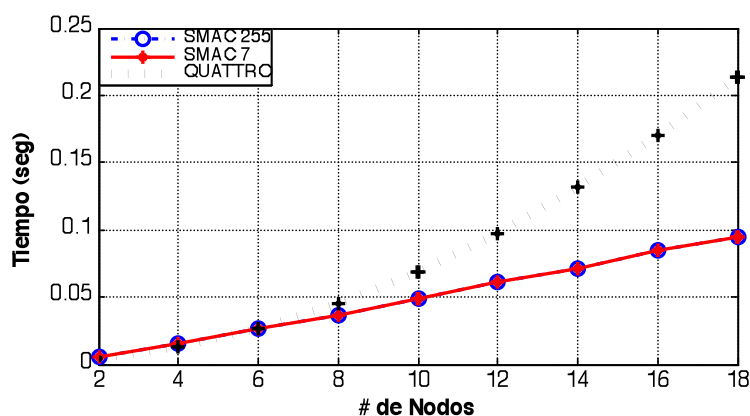


Figura 29. Retraso promedio, escenario: ROW.

anterior y a la baja interferencia presente en este escenario, SMAC hace llegar los paquetes al Sink con un menor retraso que QUATTRO.

Para mejorar este desempeño en QUATTRO, proponemos como trabajo futuro la simulación del procedimiento “sobre-calendarización”. Este procedimiento consiste en permitir al calendario calculado reiniciar transmisiones antes de que el ciclo activo de recolección de datos termine, siempre y cuando no se interfiera con las transmisiones en curso. Esto permitiría a nodos transmitir o recibir datos al mismo tiempo que lo hacen nodos a más de tres saltos en la misma ruta, reduciendo los tiempos de retraso una vez que los datos del primer ciclo lleguen al Sink.

La Figura 30 muestra la energía consumida por el escenario “ROW”. Aquí es donde se puede ver la ganancia de QUATTRO en comparación con SMAC para este escenario. Dado que todos los nodos tienen transmisiones calendarizadas en QUATTRO, no se desperdicia energía resolviendo colisiones producidas por nodos que intentan transmitir a su cabeza de clúster, cuando éstos aún se encuentran transmitiendo al siguiente nodo en ruta al Sink, que es lo que pasa en SMAC.

En la Figura 31 se puede ver que los nodos bajo el protocolo SMAC se mantienen encendidos más tiempo que aquellos bajo la propuesta QUATTRO. Los nodos con

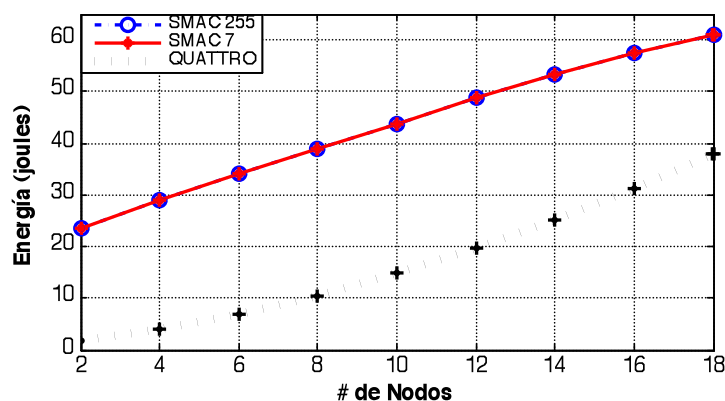


Figura 30. Energía promedio, escenario: ROW.

SMAC no tienen registro de cuáles transmisiones necesitan ser llevadas a través de ellos, por lo que si se mantienen escuchando transmisiones o colisiones en el vecindario, incrementan su ciclo encendido. En contraste, los nodos bajo la propuesta QUATTRO contienen un calendario de ventanas de actividad. Este calendario de ventanas de actividad les permite a ciertos nodos dormir a pesar de haber transmisiones en el vecindario, dado que esas transmisiones no serán reenviadas a través de ellos, sino por los correspondientes nodos activos en la calendarización.

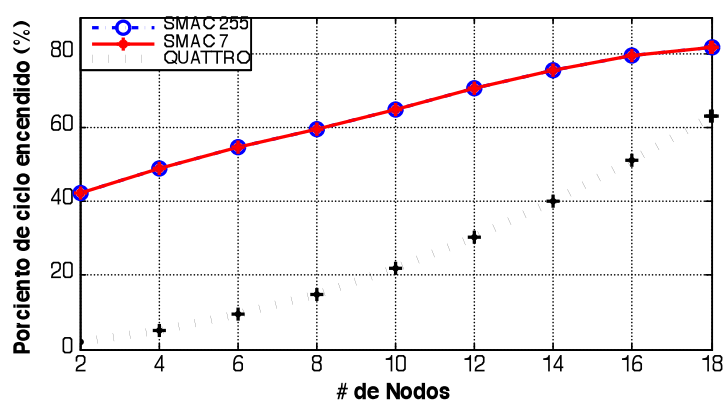


Figura 31. Porcentaje de ciclo encendido, escenario: ROW.



## IV.5 Costo de configuración.

La energía consumida durante la etapa de configuración puede ser considerada como una inversión, que dará ciertas ganancias, entre ellas, el beneficio de ahorrar energía durante la fase de operación normal. En esta sección, se analiza que tanto tiempo toma recuperar la inversión en energía, consumida durante el proceso de configuración del protocolo propuesto.

La siguiente ecuación muestra la cantidad de energía consumida para transmitir bits de datos desde su origen hasta el Sink, en función del tiempo.

$$E_b(t) = \frac{E_0 + E_{nb} \cdot N_b(t)}{N_b(t)} \quad (7)$$

En donde,  $E_0$  es la energía promedio consumida por nodo durante el período de configuración,  $E_{nb}$  es la energía promedio consumida para transmitir cada bit de datos desde su origen hasta el Sink, durante la fase de operación normal del protocolo, y  $N_b(t)$  es el número de bits de datos que se han recibido en el Sink en el tiempo  $t$ . Está claro que, para valores grandes de  $t$ ,  $E_b(t)$  se aproximará a  $E_{nb}$ , lo que significa que la inversión inicial  $E_0$  ha sido diluida a un punto insignificante.

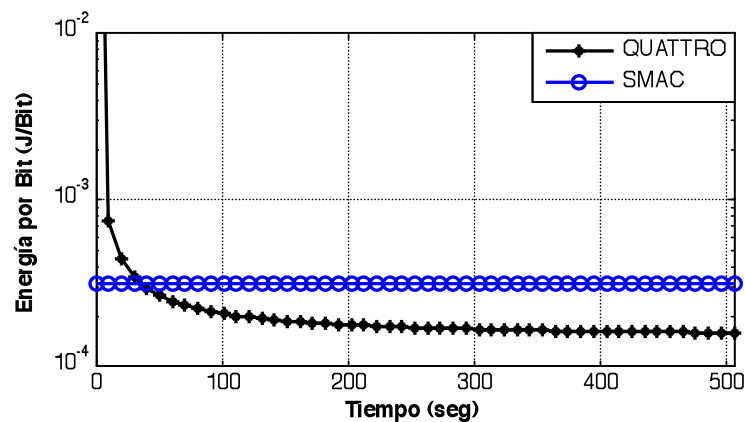


Figura 32. Energía por bit, escenario: área fija, 70n/Ac.

Para este análisis se consideró el conjunto de simulaciones descritas en el subcapítulo IV.3, para el caso específico en el cual la densidad de nodos es de  $70n/Ac$ .

La Figura 32 muestra el desempeño de  $E_b(t)$  y lo compara con el consumo de energía en SMAC. Nótese que se está despreciando la energía consumida por la fase de configuración de SMAC y se considera que el consumo promedio por bit de datos en SMAC no varía con el tiempo. De las simulaciones realizadas se obtuvo que  $E_0=24.1J$ ,  $E_{nb}=1.49 \times 10^{-4}$  J/bit y que la energía promedio consumida por SMAC para transmitir cada bit de datos desde su origen hasta el Sink es  $E_{nb,SMAC}=3.19 \times 10^{-4}$  J/bit. En la Figura 32 se puede ver que la cantidad de energía, consumida para transmitir cada bit de datos desde su origen hasta el Sink mediante QUATTRO, empieza a ser menor que el consumido mediante SMAC, después de aproximadamente 37.5 segundos o 150 ciclos. En otras palabras, en términos de energía, la inversión consumida por el proceso de configuración de QUATTRO, empieza a dar ganancias después de 37.5 segundos de transmisión de datos para el escenario en cuestión. Se puede ver también que  $E_b(t)$  converge rápidamente a su valor asintótico  $E_{nb}$ , por ejemplo, después de quince minutos (no mostrado en la figura), la diferencia es menor al 5%, y después de una hora, esta es de aproximadamente 1%.

## IV.6 Costo de las diferentes opciones de reconfiguración.

Tal como fue mencionado en la sección III.7, los nodos pueden emplear el tiempo ocioso cuando se requiere una reconfiguración (*normal*) para intercambiar los mensajes de control, o bien, detener temporalmente la transmisión de datos (*exclusivo*) para enviar los mensajes de control usando todo el tiempo, no sólo el ocioso.

También se explicó que el proceso de reconfiguración no era necesario que fuese *completo*, sino que se puede apresurar el proceso, empleando la información de enrutamiento que se recolectó previamente, pasando directamente a la fase de reservación de recursos (*rápida*). Dados los tiempos necesarios para los diferentes tipos de reconfiguración y la cantidad de datos que se perderían si dichas reconfiguraciones son debidas a fallas de nodos, se realizó un análisis de las condiciones bajo las cuales es mejor realizar una reconfiguración completa o una rápida, usando el tiempo ocioso o deteniendo la transmisión de datos para usar el tiempo completo en el proceso de recuperación.

La Figura 33 muestra el tiempo necesario para reconfigurar la red usando las tres opciones siguientes:

- Reconfiguración rápida, empleando sólo el tiempo ocioso (etiquetada: *short*).
- Reconfiguración completa empleando solo el tiempo ocioso (etiquetada: *long*).
- Reconfiguración completa empleando todo el tiempo del ciclo (etiquetada: *exclusive*).

Para recuperarse de fallas en nodos, supone que sólo un par de nodos han fallado simultáneamente (ej. 1 o 2), se puede ver que tiene sentido correr reconfiguraciones rápidas mientras la densidad de nodos esté por debajo de  $50 n/Ac$ , dado que es más rápido que correr una recuperación completa exclusiva, y no hay necesidad de detener la transmisión de datos de aquellos nodos que permanecen conectados.

De la misma forma, tiene sentido correr reconfiguraciones rápidas para densidades mayores de nodos si el porcentaje de nodos desconectados se encuentra por debajo de cierto umbral. El criterio empleado para definir este umbral es la cantidad de datos que se perdería como consecuencia del tiempo

que toma realizar una reconfiguración completa. En otras palabras, si un número pequeño de nodos se desconectan durante un tiempo relativamente largo, necesario para realizar una reconfiguración rápida normal, se perderían menos datos que si se detienen las transmisiones de datos de todos los nodos durante el período más corto, necesario para realizar una reconfiguración completa exclusiva.

Si fallan varios nodos a la vez, (ej. 3 o más), podría no ser conveniente correr una reconfiguración rápida, dado que se necesitaría actualizar la información de enrutamiento almacenada en los nodos sensores, para definir la nueva topología. En este caso, se recomienda una reconfiguración completa, pero el Sink aún tiene

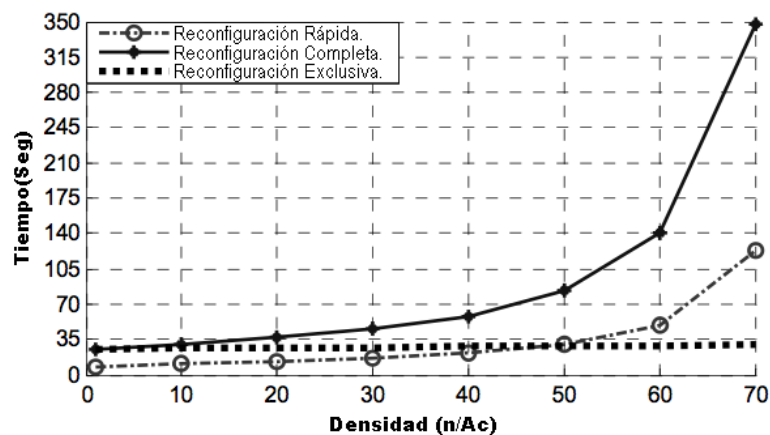


Figura 33. Tiempo de recuperación, escenario: área fija, 70n/Ac.

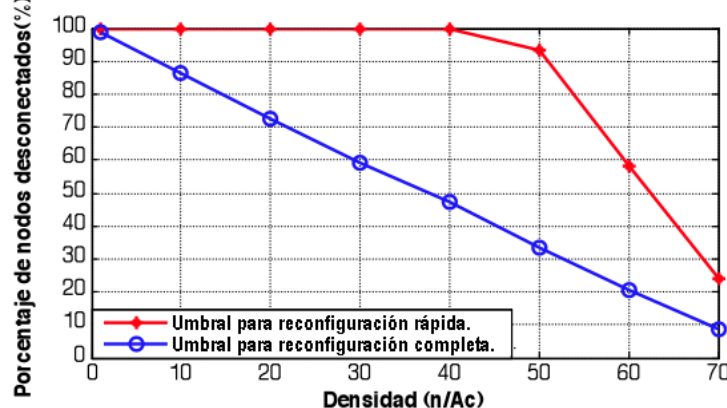


Figura 34. Umbral de nodos desconectados para diferentes tipos de recuperaciones, escenario: área fija.

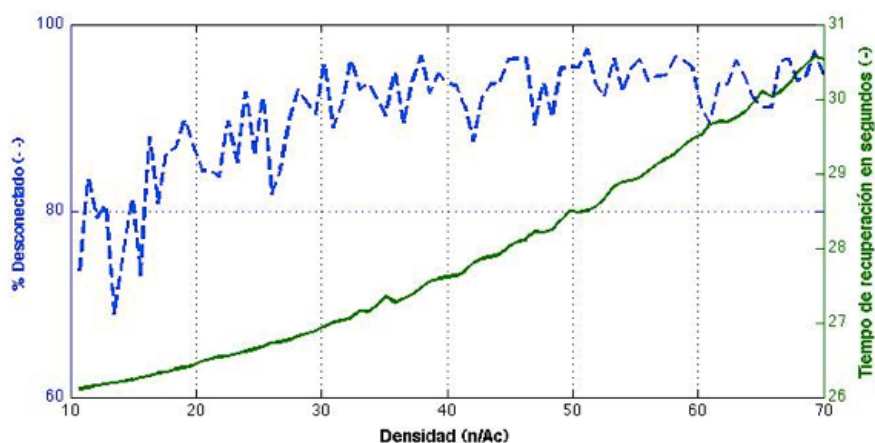
que decidir si, el detener la transmisión de datos de todos los nodos durante el tiempo de una reconfiguración exclusiva provocaría una mayor o menor pérdida de datos que el correr una reconfiguración normal, perdiendo sólo los datos generados por los nodos desconectados por un período de tiempo más largo.

Basándonos en la discusión anterior y en los resultados mostrados en la Figura 33, tenemos la Figura 34 que muestra los umbrales que definen cuándo escoger una reconfiguración rápida o una completa en vez de una recuperación exclusiva, que es el peor de los casos, cuando se necesita recuperar la topología de red en el menor tiempo posible. Estos dos tipos de recuperaciones alternativas a la exclusiva, mostradas en la figura 34, se proponen dado que pueden fallar pocos nodos al mismo tiempo sin necesidad de reconfigurar toda la red. Una recuperación es etiquetada como umbral de reconfiguración rápida y otra como reconfiguración completa. Para ilustrar la interpretación del umbral de reconfiguración rápida, por ejemplo, si se tiene una red con una densidad de 50  $n/A_c$ , cuando el porcentaje de nodos desconectados es del 93% o menor, sin mayores cambios topológicos, es mejor optar por una reconfiguración rápida en vez de una completa. Similarmente, para interpretar el umbral de reconfiguración normal, si se tiene por ejemplo una red con densidad de 60  $n/A_c$ , y el porcentaje de nodos desconectados tras una falla es del 20% o menor, y es necesaria una actualización tipológica, es mejor optar por una reconfiguración completa en vez de una exclusiva, que se realiza cuando en el peor de los casos hay que recuperar la red en el menor tiempo posible.

Cuando el objetivo de una reconfiguración es solamente la redistribución del consumo de energía, mas no una falla en un nodo, es recomendable optar por una reconfiguración completa normal. La razón de optar por una reconfiguración normal en vez de exclusiva es que es mejor no detener la transmisión de datos, y reconfiguración completa por el hecho de que no hay la urgencia de terminar el proceso de reconfiguración pues no se están perdiendo datos debido a fallas.

En la Figura 35 se muestra con una línea sólida, la cantidad de tiempo que le toma a QUATTRO reconfigurarse (encontrar rutas desde cada nodo hacia el sink y hacer las reservaciones correspondientes), para diferentes densidades de red, una vez que el 50% de los nodos han muerto. Contrario a lo que se muestra en la Figura 33, en donde se supone que solo uno o dos nodos fallan al mismo tiempo. En la figura 35 se muestra la situación en la que fallan muchos nodos a la vez, por lo que se necesita una reconfiguración exclusiva. Esto se debe a que la información de rutas debe ser actualizada antes de realizar cualquier reservación; a su vez, cuando varios nodos fallan, es muy probable que se haya perdido mucha información, lo que hace necesario usar el ancho de banda completo para mensajes de control y así acelerar el proceso de reconfiguración. Como podemos ver en la Figura 35, en todos los casos de recuperación se toma de 26 a 31 segundos.

Cuando algunos nodos fallan, es posible que algunos otros se desconecten del resto de la red. La Figura 35 también muestra, con una línea discontinua, el porcentaje de nodos que necesitan estar desconectados en promedio para que exista una sección desconectada de la red, en función de la densidad de nodos. Es de esperarse que las redes con menor densidad resulten desconectadas con



**Figura 35. Tiempo de recuperación y porcentaje de fallas necesarias para desconectar una red, en función de la densidad de nodos.**

un menor número de nodos fallidos, dado que no son capaces de proveer suficientes rutas alternas. Sin embargo, aún para densidades pequeñas se puede ver que cuando el 75% de los nodos fallan se sigue teniendo conectividad en la red.

---

## Capítulo V

---

### Conclusiones.

---

Se ha propuesto una arquitectura en la cual los protocolos MAC y enrutamiento colaboran para descubrir y reservar rutas, organizar nodos en clústeres y calendarizar el acceso al medio de transmisión en un modo de tiempo compartido. Se evaluó la propuesta mediante simulaciones, mostrando el efecto del sobre encabezado en el que incurre el protocolo, en términos de tiempo de configuración, mensajes de configuración y energía consumida. Se cuantificó el desempeño de QoS en cuanto a garantías de entrega y de límite de tiempo de entrega de paquetes. También se evaluó el desempeño de QUATTRO durante transmisión de datos bajo diferentes densidades de nodos y topologías.

Como consecuencia, la propuesta es capaz de ofrecer garantías de QoS además de lograr grandes ahorros de energía al eliminar colisiones y reducir considerablemente la escucha ociosa.

El encabezado en el que incurre la propuesta, le permite superar evidentemente el desempeño de la propuesta de referencia, SMAC, en términos de tiempo de configuración, mensajes de configuración y energía consumida.

En cuanto al desempeño de QoS de QUATTRO, éste logró mantener las garantías de retardo y caudal eficaz ofrecidas al principio de cada simulación. Se pudo ver que la inversión realizada en el proceso de configuración de la propuesta, fue recuperada inmediatamente en términos de QoS y en un corto tiempo en términos de eficiencia en energía.



En cuanto al desempeño de QUATTRO en la transmisión de datos, se concluyó que se satisficieron las garantías de QoS ofrecidas durante el proceso de configuración de cada simulación, en términos de retraso y caudal eficaz.

Se comparó el desempeño de QUATTRO con el de SMAC, el cual es uno de los primeros protocolos diseñados específicamente para WSN y sus restricciones en energía. Dado que fue uno de los primeros protocolos para WSN, SMAC ha sido frecuentemente empleado como protocolo de referencia en redes de sensores.

Los resultados muestran que QUATTRO mejora el desempeño de SMAC tanto en consumo de energía como en la habilidad extra de QUATTRO de proporcionar garantías de QoS.

La relación entre el consumo de energía en los estados de escucha ociosa, recepción y transmisión son muy similares al de varios sistemas, consumiendo más energía en transmisión que en los otros estados, con un igualmente comparable en los otros estados. Por esto, el emplear un modelo Atheros de referencia es útil para obtener datos cualitativos, aunque los valores cuantitativos cambien de aplicación en aplicación.

Las simulaciones se realizaron considerando un canal de comunicaciones ideal, donde no hubo pérdidas de paquetes debidos a fenómenos como desvanecimientos multitrayectorias, ruido e interferencia de capa física. Sin embargo se tomaron en cuenta los accesos al medio simultáneos como colisiones interferentes.

## Bibliografía

- Aalsalem M. Y., Taheri J., Iftikhar M., Zomaya A. Y., 2008, *Providing QoS guarantees to multiple classes of traffic in wireless sensor networks*, Conferencia Internacional en Sistemas Computacionales y Aplicaciones IEEE/ACS (AICCSA 2008), Doha, Qatar, del 31 de Marzo al 04 Abril del 2008.
- Atheros white paper, 2009, *Power Consumption and Energy Efficiency Comparisons of WLAN Products*. <http://www.atheros.com/pt/papers.html>. Consultado en Noviembre del 2009.
- Amre El-Hoiydi y Jean-Dominique Decotignie., 2004, *WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks*. Primer Taller Internacional en Aspectos Algorítmicos de Redes Inalámbricas de Sensores. (ALGOSENSORS 2004), LNCS 3121, pages 18–31, Turku, Finlandia, 16 de Julio del 2004.
- Boukerche A., Araujo R. B., Villas L., 2007, *A Novel QoS Based Routing Protocol for Wireless Actor and Sensor Networks*, Conferencia Global de Telecomunicaciones IEEE (GLOBECOM 2007), Washington DC, Estados Unidos, Noviembre del 2007.
- Buettner M., Yee G. V., Anderson, E. and Han R., 2006, *X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks*. Cuarto Congreso Internacional en Sistemas de Redes de Sensores Empotradas, Nueva York, Estados Unidos, Abril 2006.
- Cano-Tinoco, A. Fapojuwo O., 2007, *Performance Analysis of a QoS-Enabled Routing Protocol for Wireless Sensor Networks*, 18 Simposio Internacional en Comunicaciones Personales, de Interiores y Radio Móvil. (PIMRC), Atenas, Grecia, Febrero del 2007.
- Chen Y., Nasser N., 2008, *Enabling QoS Multipath Routing Protocol for Wireless Sensor Networks*, Conferencia Internacional en Comunicaciones IEEE (ICC 2008). Beijing, China, Mayo del 2008.
- Chow B.-S., 2009, *A Limited Resources-Based Approach to Coding for Wireless Video Sensor Networks*, IEEE Sensors Journal, 9(9): 1118 – 1124.
- Dong Y., Makrakis D., 2006, *Effective Bandwidth Calculation for QoS Routing in IEEE 802.11 DCF Ad hoc Networks*, Ad Hoc & Sensor Wireless Networks, 2(3): 245-280.

- Egorova-Forster A., Murphy A. L., 2007, *Exploring Non Uniform Quality of Service for Extending WSN Lifetime*, 5 Conferencia Anual Internacional IEEE en Cómputo Previsivo y Talleres en Comunicaciones (PerCom 2007), White Plains, Nueva York, Estados Unidos, Marzo del 2007.
- Ergen S. C., y Varaiya P., 2006, *PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks*, IEEE Trans. Mobile Comput., vol. 5, pp. 920-930.
- Fallahi A., Hossain E., 2009, *A Dynamic Programming Approach for QoS-Aware Power Management in Wireless Video Sensor Networks*, IEEE Transactions on Vehicular Technology, Vol. 58, No. 2 (Feb.), pp. 843 – 854.
- Felemban E., Lee C.-G., and Ekici E., 2006, *MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks*, IEEE Trans. Mobile Computing., vol. 5, pp. 738-754.
- Gallardo J. R., Gonzalez A., Villasenor-Gonzalez L. A., Sanchez J., 2007, *Multipath routing using generalized load sharing for wireless sensor networks*, Conferencia Internacional en Comunicaciones Ópticas e Inalámbricas (IASTED), Montreal, Canadá, Mayo del 2007.
- Gallardo J. R., Medina P., Zhuang W., 2007, *QoS Mechanisms for the MAC Protocol of IEEE 802.11 WLANs*, Wireless Networks, 13(3): 335-349 p.
- Gelenbe E., Ngai E., 2008, *Adaptive QoS routing for significant events in wireless sensor networks*, 5 Conferencia Internacional en Sistemas Ad-Hoc y Sensores Móviles. (MASS 2008), Atlanta, Georgia, Estados Unidos, Septiembre del 2008.
- Giirses E, Yuan L, Boutaba R, 2009, *Distributed Quality-Lifetime Maximization in Wireless Video Sensor Networks*, 2009 Conferencia Internacional IEEE en Comunicaciones. (ICC '09), Dresden, Alemania, Junio 2009.
- Heinzelman W. R., Chandrakasan A., Balakrishnan H., 2000, *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*, Conferencia Internacional Hawaiana en Sistemas. Maui, Hawaii, Enero del 2000.
- Krishnamachari B., 2005, *Networking Wireless Sensors*,; New York: Cambridge University Press, Cambridge, UK.

- Lou W., 2005, *An Efficient N-to-1 Multipath Routing Protocol in Wireless Sensor Networks*. Conferencia Internacional IEEE en Sistemas Móviles Ad-Hoc y de Sensores. (MASS 2005), Washington, Estados Unidos, Noviembre del 2005.
- Mills D. L., 1991, *Internet Time Synchronization: the Network Time Protocol*, IEEE Transactions of Communications, 39(10): 1482–1493 p.
- OPNET Technologies, Inc., 2009, “OPNET Modeler”, <http://www.opnet.com/> Consultado en Enero del 2009.
- Peng S., Yang S. X., Gregori S., Tian F., 2008, *An Adaptive QoS and Energy-Aware Routing Algorithm for Wireless Sensor Networks*, Conferencia Internacional en Información y Automatización. (ICIA 2008), Hunan, China, Junio del 2008.
- Polastre J., Hill J., and Culler D, 2004, *Versatile Low Power Media Access for Wireless Sensor Networks*. Segunda Conferencia Internacional en Sistemas de Redes de Sensores Empotrados. (SenSys 2004), Baltimore, MD, Estados Unidos, Noviembre del 2004.
- Rahman M. O., Monowar M. M., Hong C. S., 2008, *A QoS Adaptive Congestion Control in Wireless Sensor Network*, 10 Conferencia Internacional en Tecnología de Comunicaciones Avanzadas. (ICACT 2008), Phoenix Arizona, Estados Unidos, Febrero del 2008.
- Ruiz J, Gallardo J. R., Villasenor-Gonzalez L., Makrakis D., Mouftah H.T., 2009, *QUATTRO: QoS-capable cross-layer MAC protocol for wireless sensor networks*, Conferencia Global en Comunicaciones IEEE (GLOBECOM 2009), Honolulu, HI. Noviembre del 2009.
- Shirazi G. N., Wang P., Dong X., Eu Z. A., Tham C.-K., 2008, *A QoS network architecture for multi-hop, multi-sink target tracking WSNs*. 11 Conferencia IEEE Internacional de Singapur en Sistemas de Comunicaciones (ICCS 2008), Kraków, Polonia, Junio del 2008.
- Shu Du, Amit Kumar Saha, and David B. Johnson, 2007, *RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks*. 26 Conferencia Anual IEEE en Comunicaciones Computacionales. (INFOCOM 2007), Alaska, Mayo del 2007.
- Small T., Haas Z. J., 2007, *Quality of Service and Capacity in Constrained Intermittent-Connectivity Networks*, IEEE Transactions on Mobile Computing, 6(7).

- Su H., Zhang X., 2008, *Battery-Aware TDMA Scheduling Schemes for Wireless Sensor Networks*, Conferencia Global de Telecomunicaciones IEEE (GLOBECOM 2008), Nueva Orleans, Estados Unidos, Noviembre del 2008.
- Sundararaman B., Buy U., Kshemkalyani A. D., 2005, *Clock synchronization for wireless sensor networks: a survey*, Ad Hoc Networks, 3: 281–323 p.
- Tijs van Dam y Koen Langendoen, 2003, *An Adaptive Energy- Efficient MAC Protocol for Wireless Sensor Networks*. Primer Conferencia Internacional en Sistemas de Redes de Sensores Empotrados (SenSys 2003), Los Angeles, California, Estados Unidos, Noviembre del 2003.
- Wang Y., Henning I., Li X., Hunter D., 2006, *SOTP: A Self-Organized TDMA Protocol for Wireless Sensor Networks*, Conferencia Canadiense IEEE en Ingeniería Eléctrica y Computacional. (CCECE2006), Ottawa Ontario, Canadá, Mayo del 2006.
- Wang C., Li B., Sohraby K., Daneshmand M., Hu Y., 2007, *Upstream congestion control in wireless sensor networks through cross-layer optimization*, Revista IEEE de Áreas Selectas en Comunicaciones. 25(4): 786 – 795 p.
- Wei Ye, Heidemann J, Estrin D. 2004, *Medium access control with coordinated adaptive sleeping for wireless sensor networks*. Networking, IEEE/ACM Transactions on, 12(3): 493-506 p.
- Yanjun Sun, Shu Du, Omer Gurewitz, and David B. Johnson, 2008,. *DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks*. Noveno Simposio Internacional ACM en Cómputo y Redes Móviles Ad-Hoc. (MobiHoc 2008), Hong Kong, China, Mayo del 2008.
- Zhao L., Xu C., Xu Y., Li X., 2006, *Energy-Aware QoS Control for Wireless Sensor Network*, Conferencia IEEE en Electrónica Industrial y Aplicaciones, (ICIEA 2006) Singapur, Mayo del 2006.