

TESIS DEFENDIDA POR
Alberto González Luque
Y APROBADA POR EL SIGUIENTE COMITÉ

Dr. José Rosario Gallardo López

Director del Comité

Dr. Luis Armando Villaseñor González

Miembro del Comité

Dr. Jaime Sánchez García

Miembro del Comité

Dr. Hugo Homero Hidalgo Silva

Miembro del Comité

Dr. Arturo Velázquez Ventura

*Coordinador del programa de
posgrado en Electrónica y
Telecomunicaciones*

Dr. Edgar Gerardo Pavía López

Director de Estudios de Posgrado

06 de Julio del 2007

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



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

**REPARTICIÓN GENERALIZADA DE CARGA COMO MÉTODO DE BALANCEO DE
TRÁFICO EN REDES DE SENSORES QUE USEN ENRUTAMIENTO CON
TRAYECTORIAS MÚLTIPLES**

TESIS

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

Presenta:

ALBERTO GONZÁLEZ LUQUE

Ensenada, Baja California, México, Julio del 2007.

RESUMEN de la tesis que presenta **Alberto González Luque**, como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS en ELECTRÓNICA Y TELECOMUNICACIONES. Ensenada, Baja California. Julio de 2007.

REPARTICIÓN GENERALIZADA DE CARGA COMO MÉTODO DE BALANCEO DE TRÁFICO EN REDES DE SENSORES QUE USEN ENRUTAMIENTO CON TRAYECTORIAS MÚLTIPLES

Resumen aprobado por:

Dr. José Rosario Gallardo López
Director de Tesis

Una red inalámbrica de sensores (WSN) es un sistema formado por nodos que cuentan con algún tipo de sensor y que son capaces de enviar sus mediciones a un nodo especial, conocido como sumidero, que usualmente es más poderoso que los demás en cuanto a capacidad de procesamiento, memoria o batería. Es bien sabido que uno de los requisitos más importantes de las WSNs es funcionar con una alta eficiencia en energía.

Por otro lado, la mayoría de los protocolos de enrutamiento seleccionan una ruta única (la de menor costo) para enviar información entre cada par fuente-destino. Sin embargo, en una red razonablemente densa, pueden existir varias rutas con costo aceptable entre una fuente y su destino. El concepto de enrutamiento con trayectorias múltiples consiste en dar a cada nodo fuente, en todo momento, la posibilidad de usar cualquiera de varias rutas para llegar a un destino en particular. Estas rutas múltiples pueden ser usadas de manera alternada o concurrente. El enrutamiento multi-trayectoria ha sido una de las direcciones más importantes en las que se ha hecho investigación reciente en el área de enrutamiento. Se propuso inicialmente como un medio para mejorar el balanceo de carga y la confiabilidad de las redes de paquetes. Desde entonces se ha aplicado a varios tipos de redes, tanto fijas como móviles, y con diferentes fines, tales como suavizar las variaciones del tráfico de una conexión en particular, aliviar el congestionamiento de las redes mediante el balanceo de carga, apoyar las políticas de calidad de servicio, mejorar la confiabilidad de la red, e incluso para aumentar la privacidad de la información en tránsito.

Además de las ventajas que el enrutamiento multi-trayectoria introduce en otros tipos de redes, como se mencionó en el párrafo anterior, en las WSNs tiene la ventaja adicional de extender el tiempo de vida del sistema mediante la distribución más homogénea del consumo de energía entre los nodos.

En este trabajo se proponen técnicas que permiten la aplicación del enrutamiento con trayectorias múltiples en un entorno de WSN, incluyendo métodos para descubrir rutas múltiples entre cada nodo sensor y el sumidero, para asignar un peso a cada una de las rutas, así como para decidir la ruta que seguirá cada paquete generado por un nodo específico en su camino al sumidero. Nuestras propuestas se evalúan usando simulaciones y se comparan con otras opciones planteadas previamente en la literatura.

Palabras clave: enrutamiento con trayectorias múltiples, repartición generalizada de carga, redes inalámbricas de sensores.

ABSTRACT of the thesis presented by **Alberto González Luque** as a partial requirement to obtain the **MASTER OF SCIENCE** degree in **ELECTRONICS AND TELECOMMUNICATIONS**. Ensenada, Baja California, Mexico. July 2007.

GENERALIZED LOAD SHARING AS A LOAD-BALANCING TECHNIQUE IN WIRELESS SENSOR NETWORKS THAT USE MULTIPATH ROUTING

Abstract approved by:

Dr. José Rosario Gallardo López
Thesis Supervisor

A wireless sensor network (WSN) is a system consisting of nodes equipped with a sensor and capable of sending the collected measurements to a special node known as the sink, which is usually more powerful than the others in term of processing capacity, memory or battery lifetime. It is well known that one of the key requirements of WSNs is to work with high energy efficiency.

On the other hand, most current routing protocols work by selecting a single path (that with the minimum cost) to send information between a specific source-destination pair. However, in a reasonably well-connected network, there may exist several paths with acceptable cost between a source and a destination. The concept of multipath routing consists of giving a source node the possibility to use any of several paths to a particular destination at any given time. The multiple paths may be used alternately or concurrently. Multipath routing, also called traffic dispersion, has been one of the most important recent directions in the area of routing.

Multipath routing was first proposed as a means for load balancing and reliability enhancing in packet switching networks. Since then, the multipath routing technique has been applied to various types of networks, both fixed and mobile, and for different purposes, such as smoothing out the traffic corresponding to a specific connection, alleviating network congestion by balancing the load, supporting Quality of Service (QoS), improving reliability, and even enhancing the privacy of the information being sent.

In addition to the advantages that multipath routing brings to other types of networks, as mentioned above, in a WSN it has the additional advantage of extending the lifetime of the system by distributing more homogeneously the energy consumption among the nodes of the system.

In this work we propose techniques that allow the application of multipath routing in a WSN environment, including methods to discover multiple routes between each sensing node and the sink, to assign a weight to each one of these routes, and to decide the path that each packet being generated by a specific sensing node will follow en route to the sink. Our proposals are evaluated using simulations and compared to other options previously proposed in the literature.

Keywords: Multipath routing, generalized load sharing, wireless sensor networks.

DEDICATORIA

A mis padres

Roberto González Romo y Hermelinda Luque por su cariño, apoyo incondicional, consejos, amor y confianza brindados a través de mi vida, los mejores padres que alguien podría tener, doy gracias a dios por haberme permitido ser su hijo y formar parte de la estupenda familia que han formado, ¡los quiero mucho!

A mis hermanas

Marissa, Marcela, Rocío e Imelda, que además de excelentes hermanas son excelentes amigas, gracias por los momentos, anécdotas y consejos dados que me han servido para salir adelante, espero seguir cosechando anécdotas junto a ustedes: mis hermanas, mis amigas, ¡las quiero mucho!

A Sarai

gracias niña por todo este maravilloso tiempo juntos, me has permitido ver la vida de otro color, doy gracias a dios por haberte conocido, por haber encontrado a una mujer inteligente, hermosa y excelente persona, con la cual he compartido anécdotas y penas, gracias por todo Sarita, ¡te quiero mucho!

A mis tíos

que siempre han confiado en mi, especialmente a mi tío Jesús, que está más emocionado por mi tesis que yo, además de mi abuela y bisabuela.

AGRADECIMIENTOS

Primeramente quisiera agradecer a mi director de tesis Dr. José Rosario Gallardo López quien además de ser un excelente profesor es una excelente persona, además por el tiempo dedicado a este trabajo de tesis y consejos que han servido para pensar mejor en mi futuro profesional, muchas gracias y mucho éxito en su sabático.

A mis sinodales, Dr. Jaime Sánchez, Dr. Luis Villaseñor y Dr. Hugo Hidalgo, excelentes personas y profesionales, por sus consejos y ayuda para este trabajo de tesis, lo que nos permitió mejorar las propuestas.

A mis amigos de generación por todas las noches de desvelo y exámenes pasados, a mi compañero y amigo desde hace ya varios años Lennin, a mis compañeros del famosísimo cubo 241, Kobe, Iván y sus historias de Loreto, Andalón y sus tours y Varun, a Andrés por ser un muy buen amigo y por las atenciones que tuvo cuando todo era desconocido para nosotros en Ensenada, por supuesto que a Sarita por todos los maravillosos momentos juntos, a Mario y sus famosas carnes asadas, a mis compañeros del grupo CGI, Jorge, Richard, Rubén, Edna, a los laguna, Marco, Sergio y Daniel, a mi amigo proveniente de la escuela del fútbol de Pachuca Jonatan, a Dania y su novio Dani a quienes considero excelentes personas, y por último pero no menos importante a Aldo y Javier. Además de agradecer a Erica (la ñoña más ñoña), Leonel, Jorge, Paúl (la escandalosa) a quienes considero muy buenos amigos. Para todos ellos ¡mucho éxito!

Un agradecimiento muy especial al Dr. David Covarrubias a quien considero mi amigo, gracias por sus consejos, al Dr. Roberto Conte por los ánimos y “cortas” pláticas y demás personas que forman parte de esta gran institución que es el CICESE.

Al CICESE por haberme dado la oportunidad de superarme profesionalmente y permitirme conocer a tan excelentes personas.

Al CONACYT por haberme brindado la oportunidad, a través de ayuda financiera, de superarme profesionalmente, gracias por apoyar la ciencia y tecnología para el logro de un México mejor.

ÍNDICE

	Página
LISTA DE FIGURAS.....	iii
CAPÍTULO I	
INTRODUCCIÓN.....	1
I.1 Antecedentes.....	2
I.2 Motivación.....	4
I.3 Planteamiento del problema.....	5
I.4 Objetivos.....	6
I.5 Infraestructura.....	7
I.6 Organización de la tesis.....	7
CAPÍTULO II	
ENRUTAMIENTO POR TRAYECTORIAS MÚLTIPLES UTILIZANDO REPARTICIÓN GENERALIZADA DE CARGA PARA REDES DE SENSORES INALÁMBRICAS.....	8
II.1 Protocolo de enrutamiento por trayectorias múltiples.....	8
II.2 Asignación de pesos.....	10
II.3 Método de repartición de carga.....	15
II.3.1 GLS modificado.....	15
II.3.2 GLS con difusión.....	21
II.4 Resumen del capítulo.....	23
CAPÍTULO III	
SIMULACIÓN Y RESULTADOS.....	25
III.1 Modelo de simulación.....	25
III.1.1 Modelado del sistema en OPNET.....	28
III.2 Resultados y discusión.....	33

	Página
III.2.1 GLS modificado.....	33
III.2.2 GLS con difusión.....	45
III.3 Resumen del capítulo.....	52
 CAPÍTULO IV	
CONCLUSIONES Y TRABAJO FUTURO.....	53
IV.1 Conclusiones.....	53
IV.2 Trabajo futuro.....	54
 REFERENCIAS.....	 57

LISTA DE FIGURAS

	Página
Figura 1 - Formato del mensaje para las fases 1 y 2.....	10
Figura 2 - Formato del mensaje de prueba (WPRB).	11
Figura 3 - Formato del mensaje de respuesta a prueba (WRSP).	12
Figura 4 - Formato de los mensajes de datos (DATA).	15
Figura 5 - Modelos de la repartición generalizada de carga (GLS): (a) Modelo original de GLS, (b) Modelo modificado de GLS.	19
Figura 6 - Comparación entre algoritmos: (a) Algoritmo GLS original y (b) Algoritmo GLS modificado.	21
Figura 7 - Ejemplo de la distribución de tráfico: (a) con GLS modificado, (b) con GLS con difusión.	23
Figura 8 - Topología de la red de sensores empleada en el experimento.	26
Figura 9 - Múltiples rutas formadas por el protocolo de enrutamiento por trayectorias múltiples empleado. Rutas primarias (líneas continuas) y algunas rutas alternas (líneas punteadas) son mostradas como ejemplo.....	26
Figura 10 - Máquinas de estado finito, estados del modelo de procesos. (a) Estado forzado y (b) Estado no forzado.....	29
Figura 11 - Modelo de red del experimento en OPNET.....	30
Figura 12 - Modelo de nodo del experimento en OPNET.....	31
Figura 13 - Modelo de procesos del módulo <code>multipath_routing_and_GLS</code> en OPNET.	32
Figura 14 - Número promedio de saltos contra β	34
Figura 15 - Número promedio de saltos contra energía residual del nodo a	35
Figura 16 - Índice de equidad contra β	36
Figura 17 - Índice de equidad contra energía residual del nodo a	37
Figura 18 - Tasa de transmisión de los nodos sensores a , b , c y d , sin falla y $\beta = 0$	38
Figura 19 - Tasa de transmisión de los nodos a , b , c y d cuando el nodo a falla, $\beta = 0$	39
Figura 20 - Tasa de transmisión de los nodos a , b , c y d cuando el nodo a falla, $\beta = 1$	40
Figura 21 - Energía residual en los nodos a , b , c y d . Escenario GLS modificado, actualización de ciclo cada 5 segundos.....	41
Figura 22 - Energía residual en los nodos a , b , c y d . Escenario GLS modificado sin actualizaciones de ciclo de búsqueda de rutas.	42

Figura 23 - Energía residual en los nodos <i>a, b, c</i> y <i>d</i> . Escenario GLS modificado utilizando solo la mejor ruta.	43
Figura 24 - Energía residual en los nodos <i>a, b, c</i> y <i>d</i> . Escenario GLS modificado todas las rutas el mismo peso.....	44
Figura 25 - Energía residual en los nodos <i>a, b, c</i> y <i>d</i> . Escenario que usa la ruta más corta.	45
Figura 26 - Índice de equidad contra β , GLS con difusión.....	46
Figura 27 - Índice de equidad contra energía residual del nodo <i>a</i> , GLS con difusión.	47
Figura 28 - Tasa de transmisión de los nodos <i>a, b, c</i> y <i>d</i> . GLS con difusión.	48
Figura 29 - Energía residual en los nodos <i>a, b, c, d</i> y <i>e</i> . Escenario GLS con difusión.	49
Figura 30 - Energía residual de los nodos <i>a, b, c</i> y <i>d</i> , escenario GLS modificado con actualización de ciclo cada 5 segundos, nodo <i>y</i> en la configuración de red.	50
Figura 31 - Energía residual de los nodos <i>a, b, c</i> y <i>d</i> , escenario GLS modificado utilizando solo la mejor ruta, nodo <i>y</i> en la configuración de red.....	50
Figura 32 - Energía residual de los nodos <i>a, b, c, d</i> y <i>f</i> , escenario GLS con difusión, nodo <i>y</i> en la configuración de red.....	51

CAPÍTULO I

INTRODUCCIÓN

Las redes inalámbricas de sensores (WSNs) son un tipo particular de red ad hoc, en las cuales los nodos son “sensores inteligentes”, es decir, dispositivos pequeños (aproximadamente del tamaño de una moneda) esparcidos densamente y equipados con funcionalidades de sensado avanzadas (térmica, presión, acústicas, entre otras), un procesador pequeño, y un transceptor inalámbrico de corto alcance. Las mediciones hechas por los nodos sensores son normalmente enviadas a uno o más dispositivos especiales llamados sumideros. Otras características de este tipo de redes es que generalmente son estáticas o cuasi-estáticas y homogéneas, salvo posiblemente por el sumidero, que puede tener mayores capacidades. Es por las propias características de dichos dispositivos que están limitados en batería, ancho de banda y en capacidades computacionales. Una de las ventajas de las redes de sensores es que pueden ser empleados por científicos e ingenieros para observar y supervisar el medio ambiente que nos rodea. Estos dispositivos pequeños, inteligentes y baratos tienen varios campos de aplicaciones incluyendo el militar, el civil, industrial, entre otros.

Algunos usos que se les dan, son:

- Control del medio ambiente (por ejemplo, tráfico, hábitat, seguridad)
- Detección y diagnóstico industriales (por ejemplo, fábricas, ambientes pesados)
- Protección de la infraestructura (por ejemplo, distribución del agua)
- Conocimiento del campo de batalla (por ejemplo, seguimiento de objetivos múltiples)

Algunas otras aplicaciones pueden ser encontradas en Callaway (2004), Holger *et al* (2005), Krishnamachari (2005), Santi (2005) y Zhao y Guibas (2004). Para que las redes de sensores inalámbricas puedan ser utilizadas ampliamente se tienen que solucionar algunos retos, como por ejemplo, extender el tiempo de vida de la red, que la red sea robusta a fallas en los enlaces, que se adapte al medio ambiente, que los dispositivos sean capaces de autoconfigurarse y que la red sea segura y privada.

Uno de los mayores retos es el extender el tiempo de vida de las redes de sensores inalámbricas (las cuales están limitadas en energía debido a que sus nodos se alimentan mediante baterías no recargables) a través de protocolos y técnicas que aseguren que los dispositivos consuman la menor cantidad de energía posible, pero manteniendo su funcionalidad.

I.1 ANTECEDENTES

La técnica de enrutamiento con trayectorias múltiples, que consiste en establecer más de una ruta que podrán seguir los paquetes que correspondan a una conexión dada, se utiliza en redes en general con el fin de reducir las fluctuaciones del tráfico que fluye por enlaces individuales, lo que a su vez conlleva a un mejor desempeño de la red en general.

En el caso particular de las redes ad hoc y sobre todo de las redes de sensores, en las que la utilización eficiente de energía es fundamental, el enrutamiento con trayectorias múltiples es además útil para balancear la carga y no agotar la batería de los nodos de un enlace en particular cuando haya que transmitir mucho tráfico entre un par de nodos. Ha habido muchas propuestas acerca de cómo seleccionar las múltiples rutas que se van a utilizar entre cada par de nodos y acerca de cómo repartir la carga entre ellas, desde la más simple consistente en una ronda cíclica (Round Robin – RR), pasando por RR generalizado y métodos probabilísticos [Adishesu *et al*, 1999] [Var der Lande, 1998] [Lee *et al*, 1997].

Recientemente en Leung *et al* (2006) se ha propuesto un método denominado repartición generalizada de carga (Generalized Load Sharing – GLS) para balancear la carga impuesta a las diferentes rutas, que nace de un método similar al que dio origen al ya famoso método

de repartición generalizada de un procesador (Generalized Processor Sharing – GPS) y su versión paquetizada denominada encolamiento equitativo ponderado (Weighted Fair Queueing – WFQ) [Parekh y Gallager, 1993]. La repartición de carga se lleva a cabo cuando se tiene una sola fuente de carga y múltiples procesadores que pueden atenderla, que es el caso, y su función consiste en asignar a los diferentes procesadores una porción de dicha carga, mientras que la repartición de un procesador se lleva a cabo cuando se tienen múltiples fuentes y un solo procesador que puede atenderlas, y su función consiste en compartir por turnos el procesador para atender las diferentes cargas recibidas. Sin embargo, la formulación del análisis de ambas puede hacerse de manera similar. El algoritmo GLS parte de que cada ruta de las varias existentes entre una fuente y un destino de tráfico tiene asignado un peso. En el caso que nos interesa, de una WSN, estaría sin definir cómo seleccionar las múltiples rutas que se van a utilizar entre cada par de nodos y cómo asignarles un peso.

En Lou (2005) se propuso un protocolo para encontrar trayectorias múltiples en una red inalámbrica de sensores, partiendo de que el tráfico originado por los nodos siempre va a un punto en común, que es el sumidero. Las múltiples rutas que se construyen no se unen, consiguiendo con esto mayor robustez contra fallas y una red más confiable. Este mecanismo, con algunas modificaciones leves, resultó muy útil para nuestro trabajo.

En Lu *et al* (2006), se ha propuesto un protocolo de enrutamiento para encontrar trayectorias múltiples en una red de sensores y balancear el tráfico que fluye por dichas trayectorias, pero desafortunadamente tiene serias deficiencias, entre las que se pueden mencionar las siguientes:

- La búsqueda de rutas se inicia cuando ya hay información por enviar al sumidero, lo que puede introducir grandes retardos para la entrega de dicha información; lo anterior en principio no ahorra gran cosa dado que se supone que la red es cuasi-estática, por lo que podrían buscarse las rutas con anticipación y seguirían siendo válidas cuando hagan falta dada la poca movilidad de los nodos.

-
- La forma en que se selecciona en cada nodo el siguiente salto para formar una ruta no garantiza que las trayectorias encontradas sean disjuntas (que no tengan nodos en común), pero se supone que lo son a la hora de asignarles un peso.
 - Para obtener el costo de cada enlace es necesario estar siempre enterados del nivel de energía residual e inicial de los nodos vecinos, lo cual es costoso y complejo para cada nodo.
 - La deficiencia que consideramos más grave de dicho algoritmo es la ecuación utilizada para elegir el siguiente nodo en una ruta, ya que establece criterios que favorecen la creación de rutas largas, además de que no toma en cuenta la energía disponible en cada nodo vecino para decidir cuánto tráfico irá por él, sino que se basa en la energía consumida, lo cual es una forma poco conveniente de tomar en cuenta la energía de los nodos.
 - No se menciona el algoritmo empleado para repartir la carga, sino que sólo se hace mención a la forma de asignar un peso a cada una de las múltiples rutas formadas, pero sin mencionar como será repartida la carga para asegurar que esos pesos se respeten.
 - No se resuelve el problema de optimización planteado, ya que la forma en que se balancea la carga es local; es decir, se balancea la carga sólo entre las rutas descubiertas por un mismo nodo, sin tomar en cuenta el estado de los demás nodos que conforman la red, lo que potencialmente ocasiona un desbalance de carga a nivel global.

En base a lo anterior, concluimos que el uso del mecanismo propuesto en Lu *et al* (2006) no es confiable, por lo que resulta necesario hacer una nueva propuesta que esté mejor fundamentada y que tenga un desempeño que pueda probarse, en nuestro caso a través de simulaciones.

I.2 MOTIVACIÓN

Muchos de los protocolos que existen actualmente para WSNs utilizan una sola ruta para transmitir la información desde los nodos fuente al sumidero. Algunos protocolos que

utilizan múltiples rutas para esparcir la carga [Ganesan *et al*, 2000] [Wu *et al*, 2004] eligen la robustez como prioridad al momento de diseñar el protocolo para la red de sensores. La transmisión de los datos recae mayormente en la ruta óptima, que en la mayoría de los casos también es la ruta más corta. Las rutas alternativas son usadas sólo cuando hay fallas en la ruta óptima o primaria. Aunque los protocolos en los cuales se construye una sola ruta para llegar al sumidero son simples, flexibles y escalables, utilizar una sola ruta para enviar el tráfico al sumidero tiene desventajas que no pueden ser ignoradas. Una de las más graves es el terminar de forma prematura con la vida de los nodos sensores que participan en esa ruta.

Además del balanceo de carga y la extensión del tiempo de vida de la red, otros beneficios potenciales del uso de enrutamiento multitrayectoria serían:

- aligerar las variaciones de tráfico debido a que las ráfagas generadas se reparten entre las múltiples rutas,
- reducir el congestionamiento de la red mediante el balanceo de tráfico,
- dar apoyo al otorgamiento de calidad de servicio mediante la reducción de retardos y pérdidas de paquetes,
- mejorar la confiabilidad de la red y su auto reparación en caso de fallas,
- incluso, mejoramiento de la privacidad de la información enviada mediante técnicas especiales de codificación [Lou *et al*, 2005].

I.3 PLANTEAMIENTO DEL PROBLEMA

Para poder usar la técnica de enrutamiento por trayectorias múltiples en una red de sensores hacen falta tres ingredientes:

- Un método para el descubrimiento de rutas múltiples, preferentemente sin traslapes, entre cada nodo sensor y el sumidero

- Un algoritmo para la asignación de un factor de peso a cada una de esas rutas, que definirá la porción del tráfico generado por el nodo que será enviado por la ruta en cuestión.
- Un mecanismo para asegurarnos de que la cantidad de tráfico enviada por cada ruta es lo más aproximada posible a la fracción definida por su factor de peso.

Como se mencionó anteriormente, ha habido varias propuestas de algoritmos para el descubrimiento de rutas múltiples. Sin embargo, nuestro trabajo toma como base lo propuesto en Lou (2005) para encontrar las múltiples trayectorias entre los nodos sensores y el sumidero, dado que este algoritmo fue diseñado específicamente para una red de sensores.

Es necesario proponer un algoritmo original para la asignación de pesos a las diferentes rutas ya que, hasta donde sabemos, no existe en la literatura uno que pueda ser usado para este fin.

Finalmente, el algoritmo que se empleará para distribuir equitativamente la carga originada por los nodos sensores se basa en lo realizado en Leung *et al* (2006), con algunas modificaciones para mejorar su funcionamiento.

I.4 OBJETIVOS

En el presente proyecto se propone definir metodologías para encontrar las múltiples rutas que se pueden utilizar entre cada pareja fuente-destino de tráfico en una red de sensores, para asignarles un peso dependiendo del estado de la ruta extremo a extremo, así como para difundir esa información a los diferentes nodos que participan en las rutas para que tomen las decisiones adecuadas de enrutamiento. Finalmente, se propone definir e implantar una versión mejorada de GLS que permitirá una repartición más eficiente de la carga entre las diferentes rutas.

I.5 INFRAESTRUCTURA

Para el desarrollo de este trabajo de tesis se contó con la siguiente infraestructura que permitió la simulación de una red de sensores inalámbrica, la mayor parte fue proporcionada por el Centro de Investigación Científica y Educación Superior de Ensenada (CICESE):

- Computadora portátil Toshiba, procesador centrino a 1.73 GHz, 512 MB de RAM y 80 GB de disco duro.
- Conexión a Internet.
- Biblioteca CCMC-UNAM
- Biblioteca CICESE
- Simulador de redes OPNET v11.5.A
- Equipo de cómputo

Además de análisis matemático y programación en lenguaje C++, pudiendo con esto obtener algunos resultados analíticos necesarios para este trabajo.

I.6 ORGANIZACIÓN DE LA TESIS

En esta parte se presenta cómo está estructurado este trabajo y se describen de forma breve los temas a tratar en los capítulos siguientes.

En el capítulo II se describirá en sí lo que es la propuesta, parte por parte. Esto es, se describirá la metodología para seleccionar las múltiples rutas que existirán entre cada par fuente-destino, la forma en que se asignarán los pesos a las múltiples rutas encontradas y las versiones modificadas de los algoritmos empleados para colocar el tráfico generado por los nodos sensores.

Por último, en el capítulo III se presentan los resultados obtenidos de las simulaciones en OPNET, con lo cual se podrá evaluar el desempeño de nuestras propuestas y así con esto concluir y proponer modificaciones a las propuestas para trabajo futuro en el capítulo IV.

CAPÍTULO II

ENRUTAMIENTO POR TRAYECTORIAS MÚLTIPLES UTILIZANDO REPARTICIÓN GENERALIZADA DE CARGA PARA REDES DE SENSORES INALÁMBRICAS

II.1 PROTOCOLO DE ENRUTAMIENTO POR TRAYECTORIAS MÚLTIPLES

Como ya se ha mencionado anteriormente, el protocolo de enrutamiento por trayectorias múltiples está basado en el trabajo realizado en Lou (2005), solamente se han agregado algunos conceptos, que se indican más adelante.

El protocolo de enrutamiento descrito en Lou (2005) consiste de dos fases. En la primera fase, que se repite periódicamente, el sumidero difunde un mensaje de actualización de rutas, cuyo formato se muestra en la figura 1. El tipo de este mensaje (*mtype*) será RPRI, indicando que su propósito es el descubrimiento de la ruta primaria desde cada nodo al sumidero. El valor del campo *mid* identifica el ciclo de actualización de rutas y será el mismo en cada mensaje transmitido durante las dos fases del ciclo. El identificador de rama (*bid*) se dejará inicialmente vacío, el costo (*cst*) será puesto a cero y el campo *path* incluirá solamente al sumidero mismo. El campo *bid* cambia cuando un nodo sensor recibe por primera vez el mensaje y encuentra vacío este campo, entonces el nodo coloca su *nid* en el campo *bid*, indicando con esto que será el comienzo de la rama. Cada nodo sensor, cuando reciba la actualización por primera vez, redifundirá el mensaje modificando los campos *nid*, *bid*, *cst* y *path* como corresponda y marcará al nodo desde el cual recibió el mensaje (*nid*) como su *padre* y el *bid* recibido como su rama. Esto continúa recursivamente hasta que cada nodo en la red ha recibido una copia del mensaje y ha identificado a su *padre*.

Cuando un nodo, por ejemplo z , escucha una actualización de rutas con el mismo número de secuencia (*mid*) que uno que ya ha escuchado, marcará al vecino desde el cual es recibido como un *descendiente*, como *ancestro* o como *primo*, dependiendo de la *bid* y la ruta contenida en el mensaje. Si el mensaje tiene el mismo *bid* que el nodo receptor z , el vecino será un *descendiente* si z está en la *ruta* o un *ancestro* si no lo está. Si el *bid* es diferente, el vecino será marcado como *primo*. El *padre* representa el siguiente salto hacia el sumidero en la ruta primaria, y cada *primo* representa una potencial ruta alternativa hacia el sumidero. El nodo z mantendrá un conjunto Q_z de rutas alternas (no primarias) sin traslapes hacia el sumidero y guardará en ella toda la información relacionada, incluyendo *bid*, *cst*, y *path* de cada ruta. Este conjunto incluirá, en la primera fase, a lo más una ruta por cada *bid*, a saber aquella con el menor costo.

La segunda fase del protocolo de enrutamiento comienza independientemente en cada nodo sensor, y consiste en la transmisión de mensajes del tipo RALT, utilizados para compartir con los vecinos el conjunto de rutas alternativas encontradas por el nodo. Estos mensajes tienen el mismo formato mostrado en la figura 1. Para cada ruta alternativa descubierta, representada por el *primo* q , el nodo z envía un mensaje con el siguiente contenido: {RALT, *mid*, z , $q.bid$, $q.cst$, $q.path$ }. El campo $q.path$ incluye los nodos en la ruta comenzando con q hasta llegar al sumidero. Estos mensajes serán enviados cada vez que se descubra una nueva ruta, independientemente de si es o no adicionada al conjunto Q_z . Esto es diferente en nuestro caso a lo propuesto en Lou (2005) pero sirve para dar mayor efectividad al descubrimiento de rutas múltiples.

Cuando se recibe un mensaje RALT, un nodo z lo ignorará si él mismo está contenido en la *ruta* del mensaje. De lo contrario, la nueva ruta descubierta q será incluida en Q_z si no se traslapa con alguna otra ruta de menor o igual costo ya almacenada. Si esta ruta es adicionada a Q_z , el nodo z removerá de este conjunto todas las otras rutas alternas que pudo haber encontrado en el transcurso de la fase que tengan alguna intersección con q . Si esta ruta se descubre por primera vez, el nodo z enviará un mensaje RALT a sus vecinos avisando de la nueva ruta encontrada. La referencia [Lou, 2005] establece que un nodo deberá ignorar aquellos mensajes RALT provenientes de su *padre*, pero en nuestra

experiencia estos mensajes son algunas veces de utilidad para descubrir nuevas rutas más cortas.

Cuando la segunda fase termina, los nodos sensores han descubierto todas las rutas disponibles de mínimo costo al sumidero.



Figura 1 - Formato del mensaje RPRI y RALT utilizadas en las fases 1 y 2.

En donde:

mtype: tipo de mensaje (RPRI o RALT)

mid: número de secuencia del ciclo de actualización de ruta

nid: identificador del nodo que envía el mensaje

bid: identificador del nodo más cercano al sumidero en esta rama del árbol

cst: costo de la ruta hasta ahora (i.e. número de saltos al sumidero)

path: conjunto de nodos que el mensaje ha visitado hasta ahora

II.2 ASIGNACIÓN DE PESOS

En esta parte del capítulo se propone agregar dos fases más al protocolo descrito anteriormente. La finalidad es recabar información suficiente para asignar un peso a cada una de las múltiples trayectorias descubiertas por un nodo sensor. Para esto se definen dos

tipos de mensajes más: *prueba* (WPRB) y *respuesta a prueba* (WRSP), con formatos mostrados en las figuras 2 y 3, respectivamente.

La fase tres comienza independientemente en cada nodo sensor, digamos z , después de un periodo razonable de tiempo durante el cual ningún mensaje RALT es recibido. Si el conjunto Q_z no está vacío, o sea, si existe al menos una ruta alterna, el nodo enviará un mensaje WPRB a través de cada una de las N rutas descubiertas, incluyendo la ruta primaria y las alternas. El campo r_nid contendrá el nodo siguiente en la ruta y el campo $path$ contendrá la trayectoria completa desde el nodo sensor que originó el mensaje hasta llegar al sumidero. Cuando un nodo sensor recibe uno de estos mensajes, si el campo r_nid coincide con su propio identificador, incrementará un contador del número de rutas que pasarán a través de él (num_routes) y reenviará el mensaje, ajustando r_nid al siguiente nodo en la ruta. El mensaje llegará eventualmente al sumidero, el cual mantendrá una tabla (*probe_table*) de las *rutas* contenidas en los mensajes de prueba recibidos.

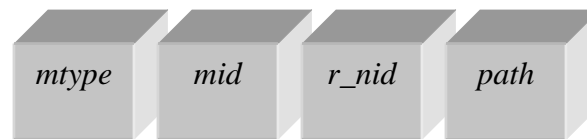


Figura 2 - Formato del mensaje de prueba (WPRB).

En donde:

mtype: tipo de mensaje colocado en prueba (WPRB)

mid: número de secuencia del ciclo de actualización de ruta

r_nid: identificador del nodo que ha de recibir el mensaje

path: ruta desde el nodo que originó el mensaje hasta el sumidero



Figura 3 - Formato del mensaje de respuesta a prueba (WRSP).

En donde:

mtype: tipo de mensaje colocado en respuesta a prueba (WRSP)

mid: número de secuencia del ciclo de actualización de ruta

nid: identificador del nodo que ha de recibir el mensaje

b-neck: cuellos de botella en energía y carga

r_path: ruta inversa desde el sumidero hasta el nodo sensor

Finalmente, la fase cuatro será iniciada por el sumidero después de un periodo de tiempo razonablemente largo en el cual ningún mensaje WPRB es recibido. El sumidero enviará un mensaje WRSP a través de cada una de las rutas almacenadas en *probe_table*. El campo *r_path* del mensaje contendrá la ruta inversa desde el sumidero hasta el nodo sensor correspondiente y el campo *r_nid* contendrá el siguiente nodo en tal ruta inversa. Inicialmente, los campos *load-b-neck* y *energy-b-neck* serán dejados vacíos. Cuando un nodo sensor reciba uno de estos mensajes, si el campo *r_nid* coincide con su propio identificador, éste reenviará el mensaje ajustando *r_nid* al siguiente nodo en la ruta inversa

y ajustando los campos de cuello de botella como sigue. Si el valor ya almacenado en *load-b-neck* no es válido o es menor que su contador *num_routes*, ajustará *load-b-neck* a *num_routes*, de otra manera, dejará el valor sin alterar. Similarmente, si el valor almacenado en *energy-b-neck* no es válido o es mayor que su energía restante *available_energy*, ajustará *energy-b-neck* a *available_energy*, de otra manera, dejará el valor sin alterar. Cabe hacer notar que se supone que los nodos pueden de alguna forma saber que tanta energía resta en sus baterías; si esto no es así, entonces el campo *energy-b-neck* será dejado sin uso (tomado como uno). El mensaje alcanzará eventualmente al nodo sensor al final de la ruta en reversa. Cuando un nodo sensor se da cuenta de que todos sus mensajes de prueba han sido respondidos o después de un periodo razonablemente largo de tiempo en el cual ningún mensaje WRSP es recibido, lo que pase primero, asignará a la ruta *i* el peso p_i de acuerdo a la siguiente expresión:

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

Donde:

ε_i = cuello de botella en energía de la ruta *i*, recibido en el mensaje de respuesta a prueba relacionado (o cero si ninguno es recibido)

λ_i = cuello de botella en carga de la ruta *i*, recibido en el mensaje de respuesta a prueba relacionado (o uno si ninguno es recibido)

h_i = número de saltos de la ruta *i*

$\beta \in (0,1)$ = factor que define el impacto que queremos que tenga el número de saltos en el peso

El razonamiento detrás de la ecuación (1) es el siguiente. La cantidad de datos que se pueden enviar a través de una ruta es proporcional a la energía restante de su nodo más débil. Además, en cada nodo la capacidad de transmisión tiene que ser compartida entre todas las rutas que van a través de tal nodo, por lo tanto, la capacidad disponible para cada

ruta es inversamente proporcional al número de rutas que pasan a través del nodo en cuestión. Por otra parte se desea involucrar tantos nodos como sea posible en la transmisión de datos para igualar el consumo de energía entre los nodos, pero al mismo tiempo se quiere evitar en lo posible el uso de rutas largas dado que la cantidad de energía gastada por el sistema a causa de los mensajes que van desde el nodo sensor hasta el sumidero es proporcional al número de saltos en la ruta; no está clara una regla de proporcionalidad en este caso entre el peso asignado a una ruta específica y su número de saltos, sino que creemos que existe un compromiso más complejo que depende de la topología, densidad y tamaño del sistema. Para permitir la flexibilidad necesaria, introducimos el exponente β , para variar de cierta forma la importancia que se le va a dar al número de saltos al momento de calcular los pesos para las rutas, el cual puede ser variado heurísticamente de sistema a sistema para mejorar el desempeño.

Después de utilizar la ecuación (1) para todas las N rutas descubiertas por un nodo sensor, los pesos tienen que ser normalizados de tal forma que todos ellos sumados sean la unidad. Cabe hacer notar que el propósito fundamental es balancear la carga y el consumo de energía por nodo (no por enlace), y nos concentramos en los nodos cuello de botella solamente. El tratar de igualar la carga, por ejemplo, en todos los enlaces, puede causar que los nodos más populares (aquellos seleccionados por muchos otros nodos para ser parte de una de sus rutas) sean subutilizados al tener que llevar la misma carga que los nodos menos populares, o puede causar que los nodos menos populares sean sobrecargados al tener que llevar la misma carga que los nodos más populares, todo esto debido a que se obliga a realizar una asignación de pesos con el fin de igualar la carga en toda la red, lo cual trae como resultado una asignación no muy justa de la carga y por lo tanto el problema anteriormente mencionado (es por esta razón que solamente se realiza el balanceo en los nodos cuello de botella). Es importante mencionar que cada cierto tiempo (que depende de factores tales como la energía, densidad, tamaño de la red, entre otros), se iniciará un nuevo ciclo de búsqueda de rutas y de asignación de pesos.

El formato de los mensajes de datos se muestra en la figura 4. Estos incluyen el identificador del nodo que originó el mensaje como una forma de identificar sin

ambigüedades el flujo, y el identificador de rama para definir la ruta que el paquete debe seguir hasta el sumidero.

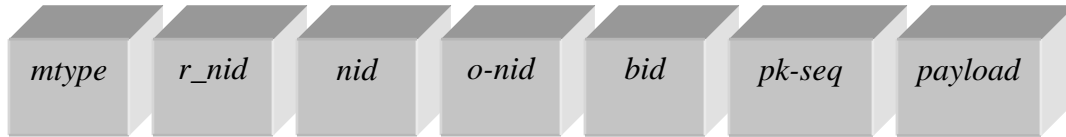


Figura 4 - Formato de los mensajes de datos (DATA).

En donde:

mtype: tipo de mensaje colocado a datos (DATA)

r_nid: identificador del nodo que ha de recibir el mensaje

nid: identificador del nodo que envía el mensaje

o_nid: identificador del nodo que originalmente generó el mensaje

bid: identificador del nodo más cercano al sumidero dentro de la ruta a ser seguida

pk-seq: número de secuencia del paquete a ser transmitido

payload: información útil de las capas superiores.

II.3 MÉTODO DE REPARTICIÓN DE CARGA

II.3.1 GLS MODIFICADO

Como se mencionó en el capítulo anterior, el algoritmo para la repartición generalizada de carga (GLS) ha sido recientemente propuesto en Leung *et al* (2006). En esta parte de la tesis se propone una versión modificada de GLS enfocada a mejorar su eficiencia.

Como ya es sabido existe un paralelismo entre GLS y la ampliamente conocida técnica de compartición generalizada de un procesador (GPS) [Parekh y Gallager, 1993] en el sentido de que este último es una técnica para compartir un servidor entre diferentes usuarios mientras que el primero reparte la carga generada por un solo usuario entre muchos servidores. En nuestro caso un servidor es un canal de comunicación y la carga es el tráfico generado por los usuarios.

El algoritmo GPS garantiza un mínimo de servicio a cada una de las sesiones, y si algunas de ellas no lo utiliza todo, se distribuye entre las sesiones que sí puedan utilizarlo. Sin embargo, el servicio será retomado por la sesión en el momento que ella lo requiera. Es importante mencionar que una sesión no puede exigir posteriormente que se le compense por la capacidad de servidor que le correspondía pero que fue distribuida entre las demás sesiones, es decir, el servidor GPS no le regresa el servicio que la sesión no aprovechó en su momento. A este enfoque se le conoce como “úselo o piérdalo”.

El algoritmo que mejor se aproxima a GPS, implementado paquete por paquete, se conoce como WFQ. Este algoritmo trabaja mediante el etiquetado de cada paquete con el tiempo en el que finalizaría su servicio en GPS y mediante la atención de la sesión cuyo paquete al frente de su cola tenga la etiqueta menor. La ecuación utilizada para calcular la etiqueta para el k -ésimo paquete generado por el i -ésimo usuario, conocido como su tiempo virtual de finalización, es la siguiente:

$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i \cdot C} \quad (2)$$

Donde:

S_i^k = tiempo en el que iniciaría el servicio en GPS para el paquete en cuestión

L_i^k = tamaño del paquete en cuestión

$\phi_i \cdot C$ = porción de la capacidad del servidor asignada al usuario i

Además:

$$S_i^k = \max\{V(a_i^k), F_i^{k-1}\} \quad (3)$$

Donde:

$V(a_i^k)$ = Tiempo virtual de arribo del paquete en cuestión

F_i^{k-1} = Tiempo virtual de finalización para el paquete previo generado por el mismo usuario

Note que si la ecuación (2) tuviera la siguiente expresión:

$$F_i^k = F_i^{k-1} + \frac{L_i^k}{\phi_i \cdot C} \quad (4)$$

Entonces el sistema trabajaría apropiadamente sólo si todos los usuarios estuvieran permanentemente generando paquetes para transmitir. Habría problemas cuando un usuario no generara paquetes por largos periodos de tiempo, dado que acumularía derechos durante su tiempo inactivo y el servidor tendría que compensar todo el tiempo que no atendió a tal usuario (aún cuando no es culpa del servidor), afectando grandemente a los demás usuarios privándolos de servicio por largos periodos de tiempo. El papel de S_i^k en (2) es entonces introducir el enfoque de “úselo o piérdalo”, mencionado anteriormente, en el sentido de que a un usuario sólo se le garantiza su porción de capacidad de servicio cuando tenga paquetes por transmitir. Algo similar a esto se empleará en nuestra propuesta de GLS, como se explica en los párrafos siguientes.

Antes de explicar nuestro algoritmo GLS modificado, definimos $W_i(k)$ y $\hat{W}_i(k)$ como la carga de trabajo esperada y real, respectivamente, a ser enviada por la ruta i , calculada justo después de que la decisión de enrutamiento para el k -ésimo paquete ha sido realizada. También definimos $R_i(k)$ como la carga de trabajo residual *normalizada* de la ruta i justo antes de realizar la decisión de enrutamiento para el k -ésimo paquete, es decir, la cantidad de carga que debe ser enviada por la ruta i con el fin de lograr la carga de trabajo esperada, dividida por p_i para dar equidad (esta normalización no se hace en Leung *et al* (2006), pero

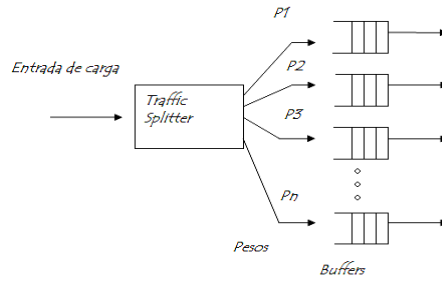
creemos que es necesaria dado que una pequeña carga de trabajo residual es más significativa conforme el peso de la ruta se vuelve más pequeño). Si $S(j)$ es el tamaño de j -ésimo paquete, entonces se tienen las siguientes ecuaciones:

$$W_i(k) = p_i \sum_{j=1}^k S(j) \quad (5)$$

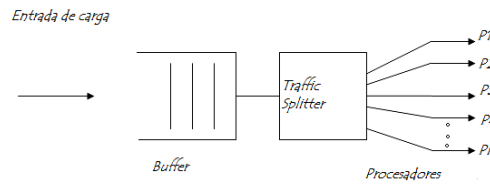
$$R_i(k) = \frac{W_i(k) - \hat{W}_i(k-1)}{p_i} \quad (6)$$

El algoritmo GLS en Leung *et al* (2006) utiliza un buffer de salida por cada ruta y decide, a la llegada del paquete, la ruta por la que debe ser transmitido seleccionando aquella con el $R_i(k)$ positivo más grande (si existiera) y almacenando el paquete en el buffer de salida correspondiente para su futura transmisión. Si hubiera empates, éstos serían resueltos por el peso de enrutamiento más grande o, si siguiera el empate, por el identificador de ruta más pequeño.

En nuestra opinión, un valor negativo (positivo) de $R_i(k)$ no significa necesariamente que hemos enviado más (o menos) tráfico por la ruta i del que puede manejar. Ésta es la razón por la que introducimos el concepto de notificación explícita, que indicará que un enlace entre dos nodos (un salto en la ruta) está listo para recibir nuevos paquetes, utilizando la técnica de ventana deslizante. Para ser más específicos, si un número fijo de paquetes N_w ha sido transmitido por un enlace y el transmisor no ha recibido un *acuse de reenvío* (FW-ACK) desde el receptor deseado, indicando que un bloque de uno o más de esos paquetes ha sido reenviado al siguiente salto, supondremos que el enlace está ocupado y no se enviará ningún nuevo paquete por esa ruta mientras persista esa condición. Aquellas rutas para las cuales el siguiente salto es el sumidero no necesitarán este tipo de acuses para continuar transmitiendo paquetes de datos; éstas nunca serán consideradas ocupadas.



(a)



(b)

Figura 5 - Modelos de la repartición generalizada de carga (GLS): (a) Modelo original de GLS, (b) Modelo modificado de GLS.

Nuestro algoritmo se define entonces como sigue. Proponemos utilizar un solo buffer en la entrada y ninguno en las salidas, como se muestra en la figura 5b. Cada paquete será removido del buffer sólo cuando exista por lo menos una ruta lista para transmitirlo. Si todas las rutas se encuentran ocupadas, el sistema esperará a que llegue el siguiente *acuse de reenvío* antes de intentar de nuevo. Definimos un conjunto de rutas ocupadas $B(k)$, que está inicialmente vacío cuando va a tomarse la decisión de enrutamiento para el paquete k . Se calcula la carga de trabajo residual para todas las rutas usando la ecuación (6) y se selecciona el valor más grande. Si la ruta ganadora está ocupada, será movida a $B(k)$ y se seleccionará un nuevo ganador de entre las rutas restantes, y así sucesivamente hasta que se encuentre un ganador que no esté ocupado o hasta que todas las rutas acaban en $B(k)$. Si se

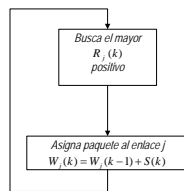
encuentra una ruta ganadora que no esté ocupada, digamos que es la ruta m , el paquete será transmitido por ella y las variables de la carga de trabajo real cambiarán como sigue:

$$\hat{W}_m(k) = \hat{W}_m(k-1) + S(k) \quad (7)$$

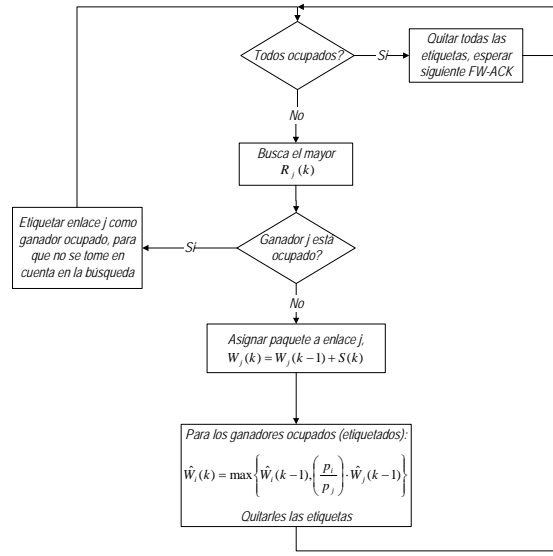
$$\hat{W}_i(k) = \max \left\{ \hat{W}_i(k-1), \left(\frac{P_i}{P_m} \right) \cdot \hat{W}_m(k-1) \right\}, \forall i \in B(k) \quad (8)$$

La finalidad del segundo término dentro las llaves en la ecuación (8) es similar a la de S_i^k en la ecuación (2). Es decir, tiene la intención de evitar que las rutas acumulen el derecho de transmitir paquetes durante el tiempo en que están ocupadas, como lo hace el algoritmo descrito en Leung *et al* (2006). Para lograr lo anterior, incrementamos artificialmente la carga de trabajo “real” de aquellas rutas que han estado ocupadas por periodos de tiempo relativamente largos seleccionando entre el máximo de la carga residual real al momento de procesar el k -ésimo paquete de las rutas ganadoras no ocupadas y la carga residual real antes de procesar el k -ésimo paquete de la ruta ganadora que no estuvo ocupada, esto hace que la carga residual de esa ruta no tenga un valor muy grande que le permita estar ganando la contienda por periodos de tiempo superiores al que debería tener si hubiera procesado el paquete cuando se le había asignado, ya que como se ha señalado a la ruta que se le deba la mayor carga residual y no este ocupada es a la que se le asigna el paquete, lo que permite dar mayor equidad al algoritmo.

La figura 6 muestra una comparación entre el algoritmo GLS original [Leung *et al*, 2006] y el algoritmo GLS modificado a través de diagramas de flujo.



(a)



(b)

Figura 6 - Comparación entre algoritmos: (a) Algoritmo GLS original y (b) Algoritmo GLS modificado.

II.3.2 GLS CON DIFUSIÓN

En la sección anterior se describieron las modificaciones realizadas al algoritmo GLS con el fin de mejorar su desempeño. Las modificaciones realizadas dieron lugar a un algoritmo que denominamos GLS modificado, que logra una mayor robustez mediante la detección rápida de fallas en las rutas a través del uso de acuses de reenvío y una mayor equidad mediante la normalización de la carga de trabajo residual, la detección de rutas ocupadas y la introducción de la filosofía úselo o piérdalo. Otra ventaja es el hecho de que nuestra versión usa un solo buffer a la entrada en lugar de buffers múltiples, uno a la salida de cada ruta, como lo hacía el algoritmo original.

Normalmente los nodos cuellos de botella en carga serán aquéllos que se encuentren a un salto del sumidero, ya que el tráfico que se genera en cualquier parte de la red debe

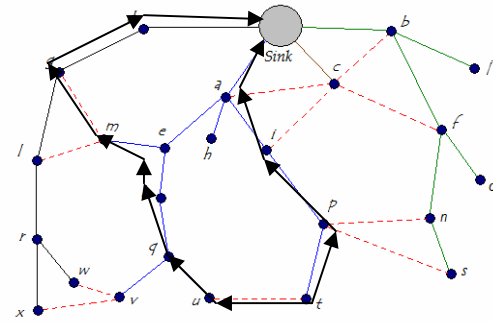
inevitablemente pasar por ellos para llegar a su destino. Dependiendo de la topología de la red, es posible que algunos nodos encuentren rutas que pasen sólo por un subconjunto de esos nodos cuello de botella y, de entrada, estarán generando un desbalance inevitable entre la carga que se impone a los nodos cuello de botella debido a que la ruta que seguirá un paquete se establece desde el nodo que lo genera y no cambia en ningún punto de su trayectoria.

A pesar de las mejoras introducidas en el algoritmo GLS modificado con respecto al original, mencionadas anteriormente, aún persiste esta posibilidad de desbalance de la carga que procesan los nodos cuello de botella. Una manera de solucionarlo sería el permitir que la ruta que seguirá un paquete no se establezca desde su nodo origen, sino que se fuera definiendo en cada nodo de la ruta, conforme el paquete avanza. Esta idea es la que da origen a una nueva propuesta que denominamos GLS con difusión. Específicamente, la difusión consistiría en permitir que el tráfico que se genera en un nodo se vaya ramificando en cada salto que dé. Lo anterior se logra colocando en cada nodo intermedio los paquetes que se reciben de otros nodos en la misma cola en la que se colocan los paquetes que se generan localmente y dando a todos los paquetes el mismo tratamiento, es decir, decidiendo quién será el siguiente nodo que visitará un paquete usando las rutas descubiertas y los pesos asignados localmente, independientemente del origen del paquete. La única precaución que habría que tomar sería la de evitar que un paquete se envíe de nuevo al nodo del que se recibió.

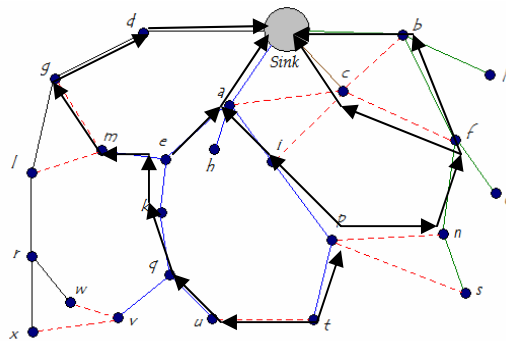
El efecto esperado es que los paquetes generados por un nodo pasen eventualmente por más nodos cuello de botella que aquéllos por los que habrían pasado si se hubiera fijado la ruta desde su origen. Es decir, que haya mayor *difusión* del tráfico generado en los nodos y por consiguiente una distribución más homogénea de la carga entre los nodos cuello de botella.

Para comprender mejor este problema, en la figura 7 se muestra una comparativa de lo que pasa en GLS modificado y GLS con difusión. En esa configuración de red, los nodos r , t , u , v , w , y x , sólo encuentran dos rutas para llegar al sumidero, que pasan por los nodos cuello de botella a y d . Esto crea un desbalance cuando se utilizan los mecanismos GLS y GLS modificado. En la figura 7a se muestra por simplicidad lo que pasaría sólo con el tráfico

generado por el nodo t . Si se usa el algoritmo GLS con difusión, por el contrario, el tráfico podría ramificarse en los nodos p y f de tal manera que cierta porción pasaría también por los nodos b y c logrando un mejor balance del tráfico.



(a)



(b)

Figura 7 - Ejemplo de la distribución de tráfico: (a) con GLS modificado, (b) con GLS con difusión.

II.4 RESUMEN DEL CAPÍTULO

En este capítulo se han explicado las modificaciones realizadas al protocolo de enrutamiento por trayectorias múltiples, que incluyen el hecho de que sí se procesan los

mensajes RALT provenientes de nodos *padres* y sí se difunden las rutas nuevas encontradas, independientemente de si se agregan o no a Q_z , además de que se han agregado dos fases más al protocolo en donde se utilizan dos nuevos tipos de paquete:

- los de prueba (WPRB), que son enviados por los nodos sensores hacia el sumidero para que cada nodo intermedio sepa cuántas rutas pasarán por él y para que el sumidero conozca las rutas descubiertas por los demás nodos de la red,
- y los de respuesta a prueba (WRSP), que sirven para identificar los cuellos de botella en carga y en energía de cada ruta, que a su vez servirán para que cada nodo pueda asignar un peso a sus múltiples rutas descubiertas.

Después de haber descubierto rutas múltiples y haber asignado un peso a cada una de ellas, se comienza a transmitir datos, los cuales son repartidos por las rutas múltiples a través del algoritmo de repartición generalizada de carga (GLS). La versión original de este algoritmo se modificó para mejorar su desempeño. Las modificaciones consistieron en quitar los buffers de salida que existían en cada ruta descubierta por un nodo sensor y en dejar sólo un buffer de entrada; también en evitar que una ruta acumule derechos de transmisión de paquetes mientras esté ocupada al aplicar la política de úselo o piérdalo y en mejorar la equidad a través de la normalización de la carga de trabajo residual; finalmente, en mejorar la robustez del sistema mediante la detección rápida de fallas usando acuses de reenvío.

Además de lo anterior, se propuso una versión adicional del algoritmo GLS, que aquí se ha denominado GLS con difusión, y que tiene como propósito el esparcir el tráfico a través de la red de manera más equitativa, y así reducir el desbalance que se obtiene con GLS modificado.

CAPÍTULO III

SIMULACIÓN Y RESULTADOS

En este capítulo, se presentan los resultados obtenidos mediante la simulación de las diferentes propuestas hechas en este trabajo de tesis. Para esto se utilizó un simulador orientado a eventos discretos.

Las métricas de desempeño utilizadas son:

- Índice de equidad en la distribución de la carga entre los nodos cuello de botella
- El número promedio de saltos que viaja cada paquete de datos, que es proporcional a la cantidad de energía usada para hacerlo llegar al sumidero
- La rapidez con que el sistema se recupera después de que un nodo falla
- El tiempo de vida del sistema, definido como aquél transcurrido hasta la muerte del primer nodo por falta de energía

En lo siguiente se presenta la topología de la red de sensores, además de una breve descripción de la herramienta empleada para realizar las simulaciones, incluyendo los modelos de nodos y de procesos. Se mostrarán también los resultados obtenidos de las simulaciones realizadas. Por último, se tiene un resumen del capítulo.

III.1 MODELO DE SIMULACIÓN

La herramienta de simulación utilizada fue el paquete OPNET v11.5.A, de cuyas características se hablará brevemente en esta sección. La red de sensores simulada consta de 24 nodos sensores y un nodo sumidero. Estos nodos sensores cuentan con las mismas

características en cuanto a capacidades computacionales, el tipo de tráfico que generan y batería (red homogénea), además de que estos mismos nodos sensores se suponen cuasi-estáticos y la red de sensores es lo suficientemente densa como para formar múltiples trayectorias desde cada nodo fuente hacia el nodo sumidero. Cada ciclo de búsqueda de rutas y asignación de pesos es de una hora. La posición de estos nodos sensores y el nodo sumidero se muestra en la figura 8. En la figura 9 se muestran las rutas primarias (líneas continuas) y algunas de las rutas alternas descubiertas (líneas punteadas) como ejemplo.

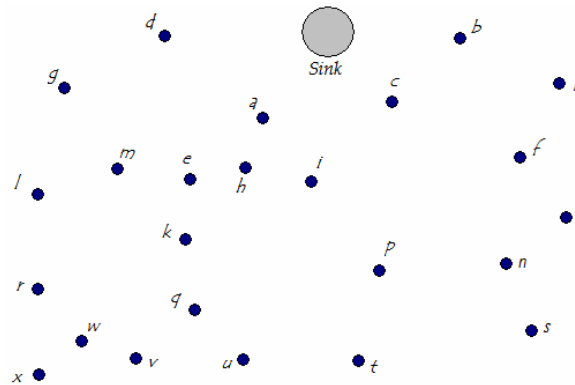


Figura 8 - Topología de la red de sensores empleada en el experimento.

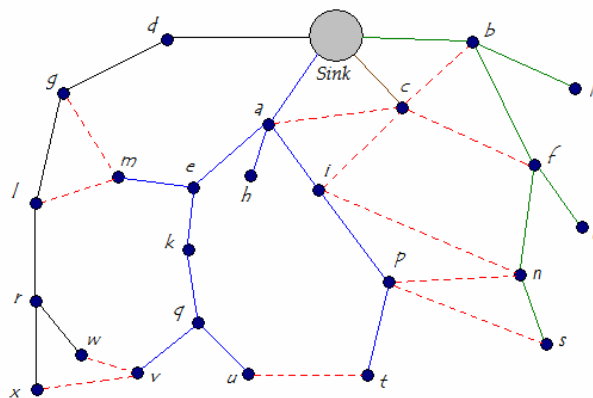


Figura 9 - Rutas múltiples formadas por el protocolo de enrutamiento por trayectorias múltiples empleado. Rutas primarias (líneas continuas) y algunas rutas alternas (líneas punteadas) son mostradas como ejemplo.

En las simulaciones se supone que cada nodo sensor cuenta con una antena omnidireccional. Se ha utilizado el protocolo MAC IEEE 802.11 (CSMA/CA) que es con el que se cuenta en OPNET v11.5.A con tasa de transmisión a 2 Mbps y se ha supuesto una capa física ideal en la cual no existe probabilidad de bit erróneo ó de pérdida de paquetes. Hay que hacer notar que el estándar que más se acomoda a las características y necesidades de una red de sensores inalámbrica es el estándar IEEE 802.15.4, pero como en la versión empleada para simular nuestros experimentos no se incluye este estándar se optó por emplear el protocolo MAC de IEEE 802.11. Además, cada nodo sensor genera 5 kbps de tráfico a tasa constante. El tamaño de la ventana N_w es puesta a 5, debido a que en las pruebas realizadas este resultó ser el mejor valor, ya que si se elige un valor menor como ventana se tomaría una decisión apresurada de que un nodo ha fallado y si tenemos un valor muy grande de ventana el tiempo en el que el sistema se daría cuenta de una falla sería demasiado, los temporizadores que nos sirven para saber cuando acaba una fase (periodos razonables de tiempo mencionados en el capítulo dos) son de 2 segundos para el sumidero y 0.5 segundos para los nodos sensores en el caso de esta tesis y dependiendo del tamaño, densidad y otros factores que se presenten en la red de sensores estos valores pueden variar.

Como se ha mencionado anteriormente, una de las métricas que nos permitirá medir el desempeño de nuestro algoritmo es el llamado *índice de equidad*, el cual representa qué tan bien está balanceada la carga a través de la red de sensores, o sea, nos permitirá saber qué tan equitativamente se está distribuyendo la carga a través de la red, tomándose como referencia los nodos cuello de botella. Los nodos cuello de botella en carga son aquéllos que manejarán la mayor cantidad de tráfico. El índice de equidad toma valores menores o iguales a uno, de tal forma que cuando este índice tenga el valor de la unidad, la carga estará perfectamente balanceada. Este índice nace de la desigualdad de la suma de Chebyshev, la cual está definida como sigue: para todo a y $b \in \mathfrak{R}^N$,

$$a = \{a_1, a_2, a_3, \dots, a_n\}$$

$$b = \{b_1, b_2, b_3, \dots, b_n\}$$

Y además,

$$a_1 \geq a_2 \geq a_3 \geq \dots \geq a_n$$

$$b_1 \geq b_2 \geq b_3 \geq \dots \geq b_n$$

Se tiene que:

$$n \sum_{j=1}^n a_j b_j \geq \left(\sum_{j=1}^n a_j \right) \left(\sum_{j=1}^n b_j \right) \quad (9)$$

Cuando $a_j = b_j \forall j$

$$n \sum_{j=1}^n a_j^2 \geq \left(\sum_{j=1}^n a_j \right)^2 \quad (10)$$

Lo cual define la desigualdad de la suma de Chebyshev. En nuestro caso, $n = Nb$ es el número de nodos cuello de botella en carga y $a_j = r_j$ es la cantidad de tráfico que el nodo j tendrá que manejar, dado por la suma de los pesos correspondientes de todas las rutas que pasarán a través de él. De aquí, se deriva la siguiente desigualdad:

$$\Phi = \frac{\left(\sum_{j=1}^{Nb} r_j \right)^2}{Nb \sum_{j=1}^{Nb} r_j^2} \leq 1 \quad (11)$$

Con esto logramos obtener una medida del desempeño del sistema en cuanto a la distribución de carga. Cuando el índice de equidad mostrado en la ecuación (11) llega a la unidad, se puede decir que la carga está perfectamente distribuida. Entre más cercano a la unidad se encuentre este valor mejor balanceada está la carga en la red.

III.1.1 MODELADO DEL SISTEMA EN OPNET

En esta parte se verá cómo se han modelado las propuestas en el simulador OPNET v11.5.A, pero antes se describirá brevemente lo que es la herramienta de simulación OPNET.

OPNET es un lenguaje de simulación orientado a las comunicaciones. Una gran ventaja de esta herramienta es que proporciona acceso directo al código fuente. En OPNET se cuenta con una jerarquía interna, la cual es necesario comprender antes de realizar el modelado del sistema. Esta jerarquía de diseño es la siguiente:

- Modelo de red: En este modelo irán definidas las redes y subredes de la simulación.
- Modelo de nodos: Aquí se define la estructura interna de los nodos, y ésta se compone de módulos, que pueden llevar a cabo diferentes funciones.
- Modelo de procesos: En este modelo se definen los estados que determinan el comportamiento de los módulos que caracterizan al modelo de nodos.

El modelo de procesos se compone de estados que definen las funciones que realizará el módulo al que pertenecen. Este conjunto de estados junto con las reglas que definen las transiciones entre ellos son también llamados máquinas de estados finitos (FSM), y la definición del funcionamiento de estos estados implica la programación en lenguaje C/C++. Cada uno de estos estados se divide en dos partes. Las instrucciones de entrada, que incluyen el código que se ejecuta cuando se entra al estado y las instrucciones de salida, que incluyen el código que se ejecuta justo antes de que el sistema cambie de estado.

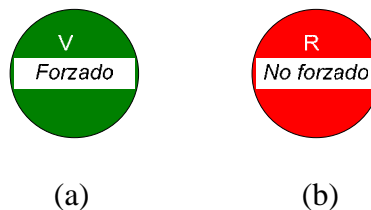


Figura 10 - Máquinas de estado finito, estados del modelo de procesos. (a) Estado forzado y (b) Estado no forzado.

Además de esto, como podemos observar en la figura 10, se cuenta con dos tipos de estados. Los estados no forzados cuyo color característico es rojo y los estados forzados cuyo color característico es el verde en OPNET. Un estado no forzado es aquel en el cual se ejecutan las instrucciones de entrada y se hace una pausa esperando una interrupción para poder pasar a las instrucciones de salida y cambiar al estado que las condiciones de transición determinen. Por otra parte, un estado forzado no espera a que ocurra interrupción alguna, sino que el código contenido en las instrucciones de entrada se ejecuta al entrar al estado para luego ejecutar el código contenido en las instrucciones de salida y cambiar al estado que las condiciones de transición determinen. También habrá que comentar que en OPNET se cuenta con procedimientos de kernel, los cuales son procedimientos que nos permitirán acceder a los servicios que ofrece OPNET. Por ejemplo, creación y destrucción de paquetes, interrupciones, lectura de atributos, interrupciones de la simulación, entre otros.

Ahora que se han descrito brevemente las características y funcionamiento de OPNET, ya podemos explicar el modelado de nuestra propuesta. Una representación del modelo de red, que muestra el número de nodos sensores, su colocación y topología, se muestra en la figura 11.



Figura 11 - Modelo de red del experimento en OPNET.

El modelo de nodos que compone a cada uno de los nodos incluidos en la figura 11, se muestra en la figura 12.

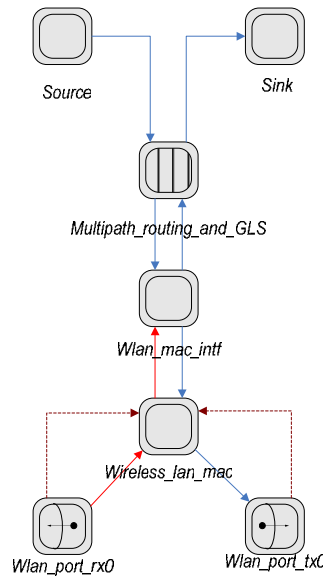


Figura 12 - Modelo de nodo del experimento en OPNET.

En ella se puede observar que el modelo de nodos se compone de varios módulos que realizan una función en específico. El módulo denominado *Wlan_port_rx0* se encarga de recibir los paquetes que van dirigidos al nodo, para luego pasarlos a la capa MAC representada por los módulos denominados *Wireless_lan_mac* y *Wlan_mac_intf*. Después el paquete pasa al módulo *Multipath_routing_and_GLS*, que es el módulo en donde se realizan las funciones del algoritmo GLS y de enrutamiento por trayectorias múltiples, procesando el paquete de acuerdo a su tipo. Finalmente el paquete pasa al módulo denominado *Sink*, cuyo único fin es el de servir de contador de estadísticas. Por otra parte, en el módulo *Source* se generan paquetes, que pasan al módulo *Multipath_routing_and_GLS*, para que tome la decisión de por qué trayectoria se ha de colocar el paquete, de acuerdo al algoritmo GLS. Posteriormente el paquete pasa a la capa

MAC, representada por los módulos *Wireless_lan_mac* y *Wlan_mac_intf*, que lo procesa y envía al siguiente módulo *Wlan_port_tx0* para ser transmitido al nodo destino. Este es el funcionamiento en general del modelo de nodos.

En la figura 13 se puede observar el modelo de procesos del módulo *Multipath_routing_and_GLS*, que fue diseñado específicamente para llevar a cabo las propuestas de esta tesis y así poder evaluar su desempeño.

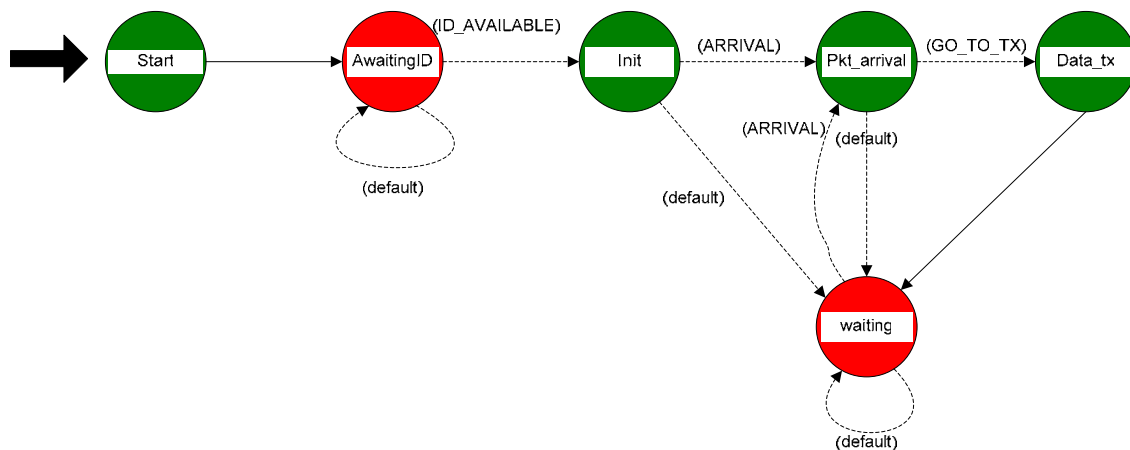


Figura 13 - Modelo de procesos del módulo *multipath_routing_and_GLS* en OPNET.

La flecha al inicio de la figura indica el comienzo del proceso, los estados *Start* y *AwaitingID* sirven para dar tiempo a que los demás módulos que componen a las capas inferiores (capa MAC) puedan inicializarse, de tal manera que el nodo sensor pueda leer la dirección MAC que le corresponderá a este nodo y que se usará como identificador de nodo para propósitos de enrutamiento. En el estado *Init* lo que se hace es inicializar ciertos parámetros que serán necesarios en la simulación y en los estados siguientes. En el estado *Waiting* se espera a que ocurra alguna interrupción, que puede darse cuando se genera un paquete en la capa de aplicación, cuando se recibe un paquete de la subcapa MAC o cuando expira un temporizador. Si se genera un paquete en la capa de aplicación, simplemente se

guardará en la cola correspondiente y se regresará a esperar un nuevo evento. Si se recibe un paquete de la subcapa MAC se procederá de acuerdo a su tipo y se enviará la respuesta correspondiente. De la misma forma, si expira un temporizador, se tomarán las medidas adecuadas, que pueden incluir el envío de mensajes o la actualización de variables. Al estado *Data_tx* se pasa cuando se dan las condiciones para enviar un paquete.

Cabe mencionar que un nodo manejará una cola independiente por cada ruta que pase por él y los paquetes que corresponden a esas rutas tendrán definido, desde que salen de su nodo origen, el conjunto de nodos que visitarán en su viaje hacia el sumidero.

Por otra parte para el algoritmo GLS con difusión, lo único que cambia es la forma en la que se trata un paquete que arriba de capas inferiores. O sea, los paquetes que van dirigidos al nodo por formar parte de la ruta hacia el sumidero son tratados de la misma forma en la que se trata a los paquetes generados por el nodo sensor, esto es, los mensajes de tipo DATA que son paquetes que llevan carga útil y son recibidos desde capas inferiores, serán almacenados en el mismo buffer en el que se guardan los mensajes de datos generados localmente para luego aplicar el algoritmo GLS, lo que trae como consecuencia que el paquete de datos proveniente de otros nodos no tenga una trayectoria fija por la que tendrá que viajar hacia el sumidero, sino que ahora en cada nodo sensor se decidirá la trayectoria para ese paquete de datos, lo que ocasionará que se distribuya la carga de manera más homogénea entre los nodos cercanos al sumidero.

III.2 RESULTADOS Y DISCUSIÓN

III.2.1 GLS MODIFICADO

En esta sección se presentarán los resultados obtenidos para el algoritmo GLS modificado. Después de haberse presentado los resultados para ese algoritmo se verán en la siguiente sección algunos resultados obtenidos para el algoritmo GLS con difusión.

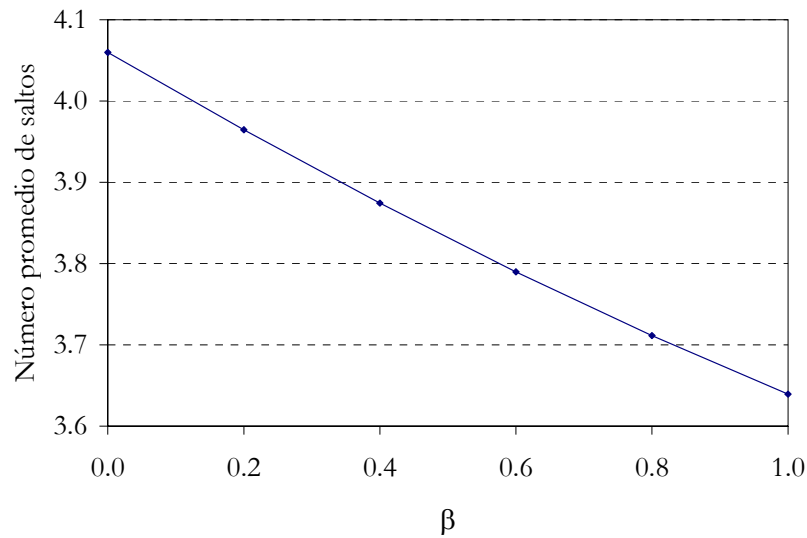


Figura 14 - Número promedio de saltos contra β .

En la figura 14, se puede observar el número promedio de saltos que se tienen que dar para enviar un paquete al sumidero contra el factor de peso que se le da a los saltos. Como podemos observar, conforme aumentamos β , siendo 1.0 el valor más alto, el número de saltos promedio que componen a las múltiples trayectorias se reduce, lo que significa que cada paquete tiene que dar en promedio un número menor de saltos. Con esto se logra que se reduzca la energía consumida por la transmisión de paquetes, ya que como sabemos el consumo de energía es directamente proporcional al número de veces que el paquete se tiene que transmitir. De hecho, si observamos la gráfica en la figura 14, se observa una diferencia del número promedio de saltos de aproximadamente 0.42 entre el mínimo y el máximo, lo que se puede traducir en un ahorro de aproximadamente 12% en el consumo de energía.

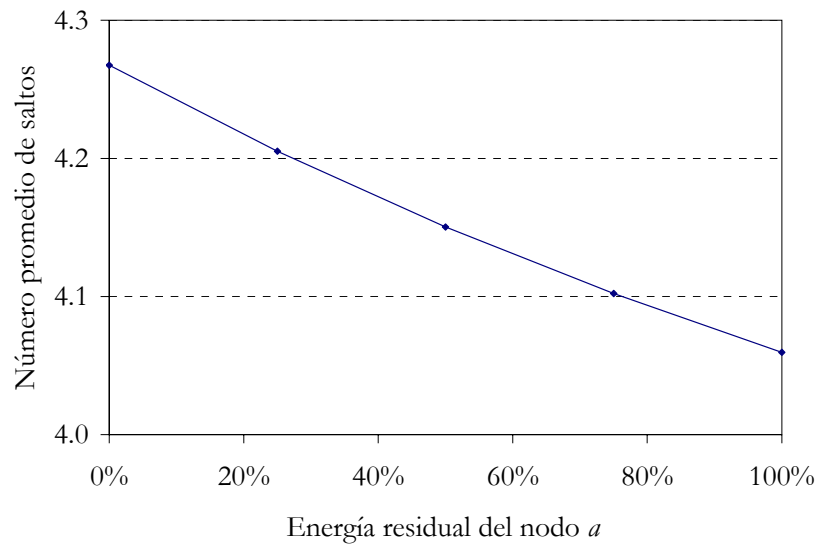


Figura 15 - Número promedio de saltos contra energía residual del nodo *a*.

En la figura 15 se muestra el número promedio de saltos contra la energía residual del nodo sensor *a* (la energía residual de los nodos *b*, *c* y *d* se mantiene al 100%). Este nodo ha sido seleccionado ya que es uno de los cuatro nodos que están a un salto del nodo sumidero y por lo tanto es uno de los nodos que más tráfico maneja. Como se puede observar, cuando el nodo sensor cuenta con la totalidad de su energía, el número promedio de saltos es menor que cuando la energía residual del nodo sensor disminuye hasta llegar a 0%. Esto es debido a que cuando el nodo sensor cuenta con la totalidad de su energía, los nodos sensores vecinos pueden enviar más tráfico a través de él, mientras que cuando la energía residual del nodo sensor *a* está por agotarse, se tratará de enviar el tráfico que pasaba a través de él por otros nodos que cuenten con mayor energía, a pesar de que las rutas sean más largas.

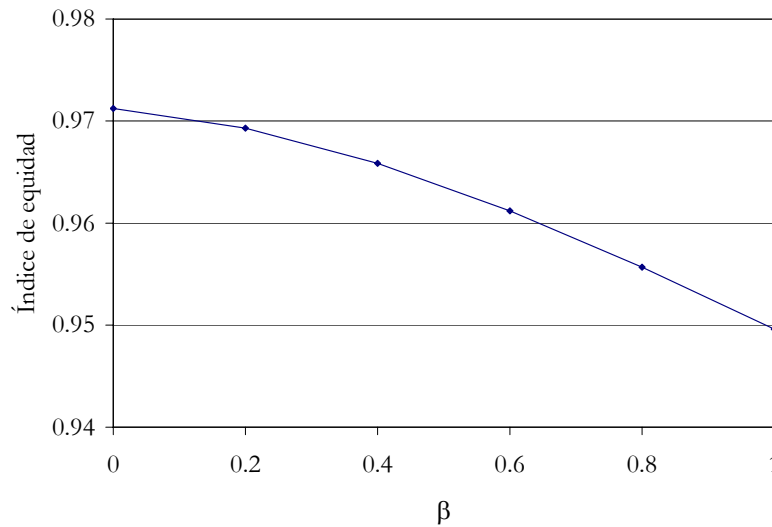


Figura 16 - Índice de equidad contra β .

En la figura 16 se puede observar la variación que tiene el índice de equidad conforme se le da más importancia al número de saltos. Cuando el factor de importancia dado a los saltos es de uno, el índice de equidad es el mínimo alcanzado, debido a que las rutas que cuentan con un número de saltos menor para llegar al nodo sumidero pueden no ser las mejores para distribuir de mejor manera la carga de la red. Por otra parte cuando tenemos un valor de cero para β , o sea, que al momento de calcular los pesos para cada trayectoria descubierta en cada nodo sensor, no se le dará importancia al número de saltos, se logra distribuir la carga hacia el sumidero de mejor manera.

Sin embargo, a pesar de que el ahorro de energía es considerable al variar β de cero a uno, como se mostró en la figura 14, el precio que hay que pagar en equidad no es muy grande ya que incluso en el peor de los casos el índice de equidad es de aproximadamente 0.95, como puede observarse en la figura 16.

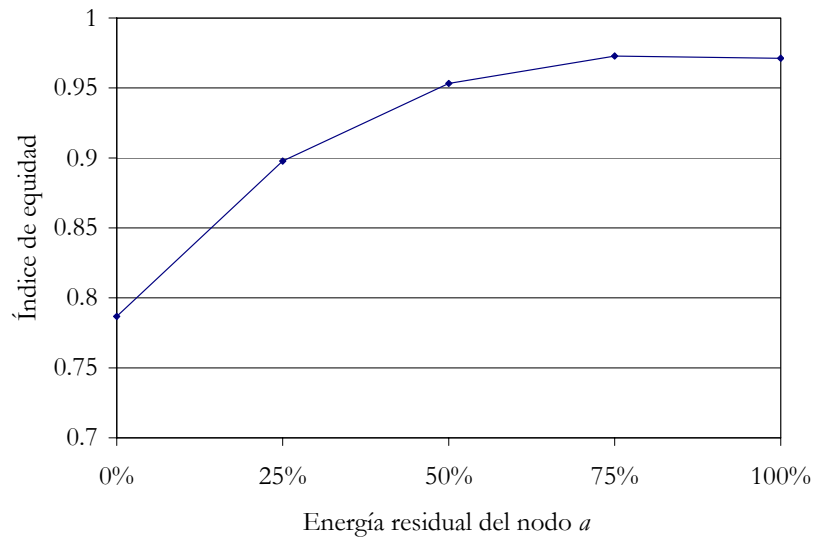


Figura 17 - Índice de equidad contra energía residual del nodo a .

En la figura 17 se grafica el índice de equidad contra la energía residual del nodo a . Como se puede observar cuando la energía residual del nodo a es del 100% se alcanza un índice de equidad mayor del que se alcanza cuando la energía residual es del 0%. Esto es debido que, al perder sus funcionalidades el nodo sensor, o sea, llega al 0% de su energía restante, se tienen que buscar otras trayectorias para poder enviar el tráfico que pasaba por el nodo sensor a . Esto, como lo demuestra la gráfica en la figura 17, crea un desbalance de carga, lo que lleva al índice de equidad a un valor aproximado de 0.78, que a la postre significa acabar con la vida de la red de sensores antes de tiempo.

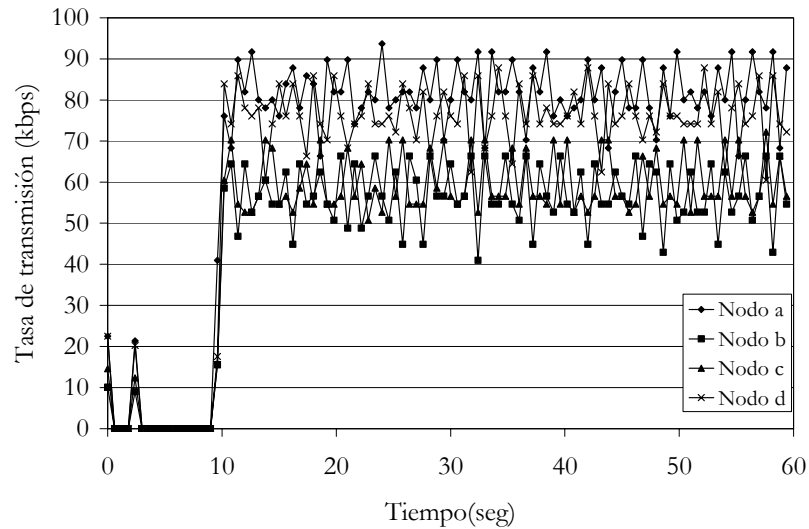


Figura 18 - Tasa de transmisión de los nodos sensores a , b , c y d , sin falla y $\beta = 0$.

En la figura 18 se pueden observar las tasas de transmisión de los nodos que están a un solo salto del nodo sumidero. Estos nodos son los que manejan las tasas de transmisión más altas ya que por ellos es por donde pasa todo el tráfico de la red. El tráfico incluye encabezados de capa MAC (capa 2), retransmisiones debidas a colisiones, tramas RTS/CTS y acuses de recibo. Además se observa que, a partir de los tres segundos de simulación, el proceso de búsqueda y asignación de pesos ha terminado.

Una prueba que se hizo al sistema, como fue mencionado anteriormente, consiste en hacer fallar uno de estos nodos para saber qué tan robusto contra fallas es el sistema. Específicamente, se hizo fallar al nodo a a partir de los 30 segundos de simulación de tal forma que los acuses de reenvío, necesarios para saber si una ruta está ocupada o no, no son recibidos por los nodos que han enviado paquetes al nodo a . La respuesta del sistema a esta falla es mostrada en la figura 19.

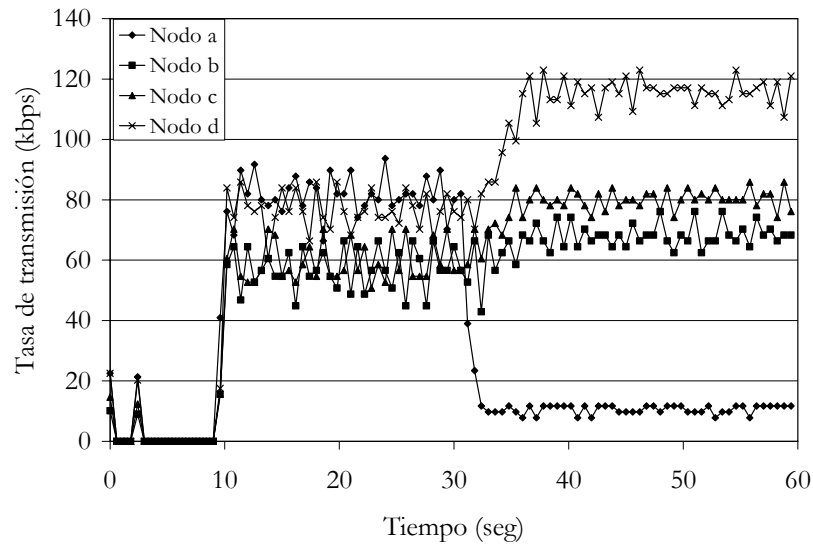


Figura 19 - Tasa de transmisión de los nodos *a*, *b*, *c* y *d* cuando el nodo *a* falla, $\beta = 0$.

Como se puede observar en la figura 19, al segundo 30 la tasa de transmisión del nodo sensor *a* sufre una drástica caída. Como consecuencia, la tasa de transmisión de los nodos *b* y *c* sufre un pequeño incremento y la tasa de transmisión del nodo sensor *d* se incrementa en casi el 50% con respecto a la tasa de transmisión que tenía antes de la falla. Lo que sucede al ocurrir esta falla es que al no enviar los mensajes de tipo FW-ACK, la ventana deslizante que nos indica si las rutas están ocupadas se cierra, lo cual significa que todos los nodos marcarán la trayectoria hacia el nodo *a* como ocupada, y seleccionarán otras rutas que no estén ocupadas. Lo más importante a analizar en este caso es el tiempo de respuesta del sistema a esta falla. Como podremos observar el tiempo de respuesta desde el segundo 30 en que falla el nodo *a* hasta que se reasigna el tráfico por los demás nodos es de aproximadamente seis segundos, lo cual es rápido. Y lo más importante es que se llega a un nuevo estado de equilibrio sin necesidad de iniciar un nuevo ciclo para descubrir rutas, por lo que se puede afirmar que el sistema es robusto a fallas debido a las características

propias del protocolo de enrutamiento por trayectorias múltiples y del algoritmo GLS modificado. Es importante mencionar que si se hubiera seleccionado un valor mayor de N_w el sistema hubiera tardado más en darse cuenta de que algún nodo ha fallado, y por lo tanto, el tiempo de respuesta a la falla hubiera sido mayor. El tiempo para darse cuenta cierto nodo de que en la ruta ha habido una falla depende del tamaño de la ventana, el tiempo de interarribo de paquetes, el número de saltos al nodo que ha fallado y a la proporción de paquetes que se envían por una ruta.

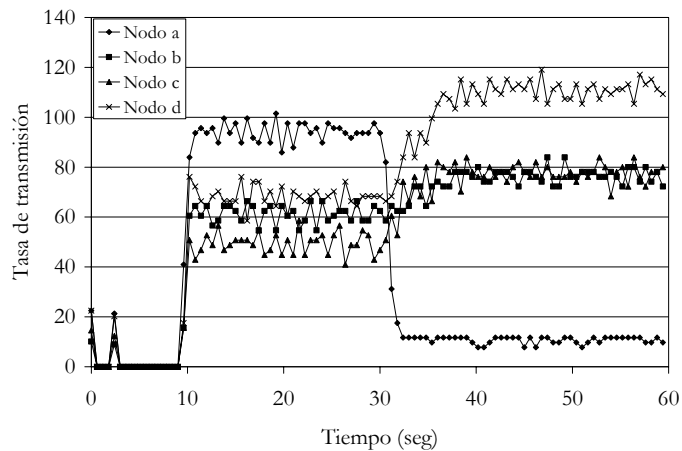


Figura 20 - Tasa de transmisión de los nodos a , b , c y d cuando el nodo a falla, $\beta = 1$.

En la figura 20 se presentan las tasas de transmisión de los nodos a , b , c y d , cuando se hace fallar al nodo sensor a , pero en este caso el valor de β es uno. Como se puede observar si comparamos con la figura 19, lo único que cambia son las tasas de transmisión de los nodos. El cambio es ligero y se debe a que se transmiten más paquetes por rutas con menor cantidad de saltos, resultando en ese pequeño incremento en la tasa de transmisión de los nodos sensores. El tiempo de respuesta del sistema es de nuevo de aproximadamente seis segundos.

Otra métrica, además de las ya mostradas anteriormente, es el tiempo de vida del sistema. Para medir el desempeño a través de esta métrica se crearon 4 escenarios adicionales a GLS modificado. En el primer escenario se tiene a GLS modificado pero sin ciclos de actualización de rutas; en el segundo escenario se tiene a GLS modificado pero sólo se utiliza la mejor ruta, o sea, la que tiene mayor peso; en el tercer escenario se tiene que se le da el mismo peso a todas las rutas descubiertas por cada nodo sensor; y en el último escenario sólo se emplea la ruta más corta para enviar el tráfico (la de menor cantidad de saltos). Los resultados obtenidos se muestran a continuación. Cabe mencionar que los niveles iniciales de energía de los nodos y la cantidad de energía consumida por cada bit transmitido no corresponden a valores reales, sino que se usaron valores arbitrarios con el solo propósito de que la energía de los nodos se consumiera rápidamente para poder hacer una comparación entre las diferentes alternativas.

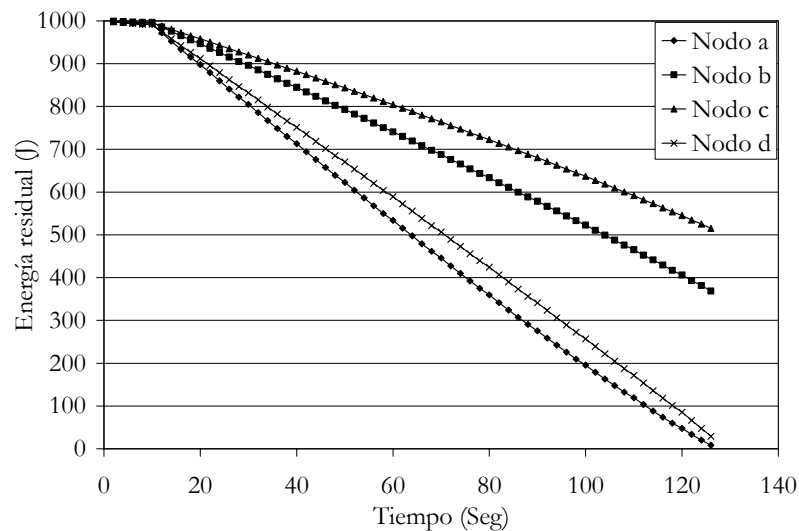


Figura 21 - Energía residual en los nodos *a*, *b*, *c* y *d*. Escenario GLS modificado, actualización de ciclo cada 5 segundos.

En la figura 21 se puede observar la energía residual del escenario GLS modificado, como se puede observar el tiempo de vida en este escenario es de 127.5 segundos y también se

observa que es al nodo *a* al que se le termina primero la energía, este escenario será comparado con los 4 escenarios siguientes. Es importante mencionar que lo que se trata de mostrar con este resultado es qué pasa con la energía cuando ocurren varios ciclos de actualización. Se trató de seleccionar un tiempo pequeño para la actualización de rutas que nos permitiera que se llevara a cabo la búsqueda de rutas y asignación de pesos (aproximadamente 3 segundos).

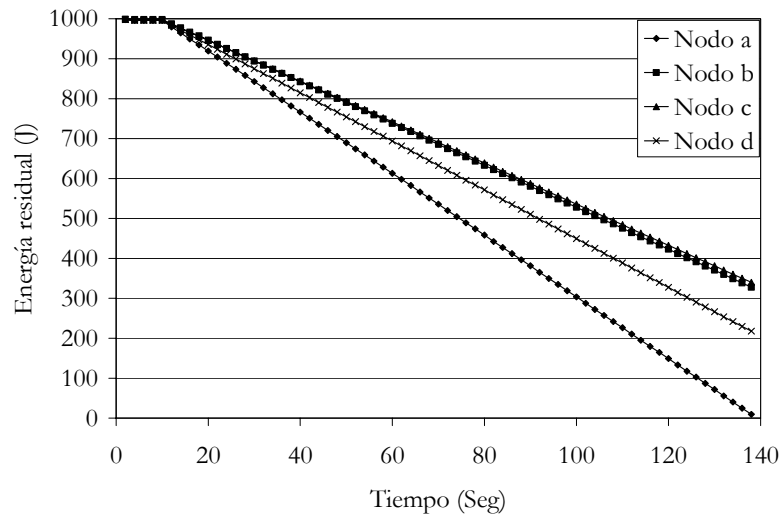


Figura 22 - Energía residual en los nodos *a*, *b*, *c* y *d*. Escenario GLS modificado sin actualizaciones de ciclo de búsqueda de rutas.

En la figura 22 se muestran los resultados obtenidos para el escenario GLS modificado sin actualizaciones de ciclo, como se puede observar el tiempo de vida en este escenario es de 139.3 segundos lo que resulta un poco mayor (~9%) que el resultado obtenido para el escenario GLS modificado el cual incluye actualizaciones de ciclo, esto solamente muestra que los mensajes de actualización de ciclo consumen un poco más de energía, es por esta razón que cuando no se utilizan los ciclos de actualización se incrementa el tiempo de vida del sistema.

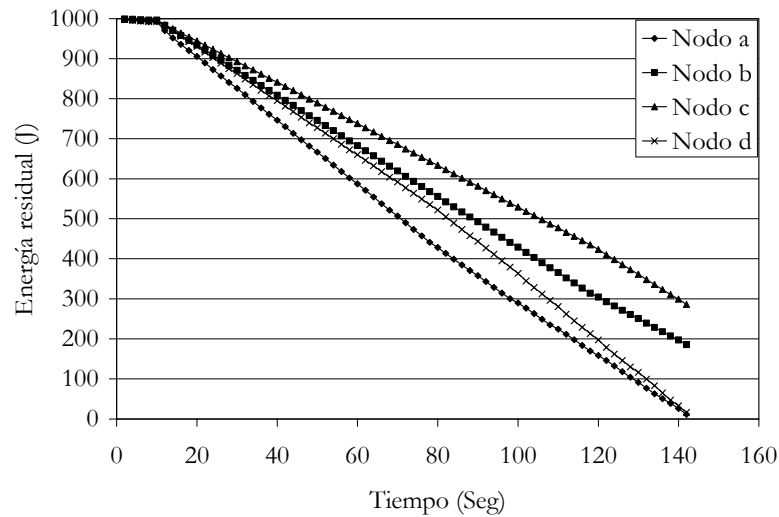


Figura 23 - Energía residual en los nodos *a*, *b*, *c* y *d*. Escenario GLS modificado utilizando solo la mejor ruta.

En la figura 23 se muestran los resultados obtenidos en cuanto al tiempo de vida para el escenario GLS modificado utilizando la mejor ruta. Como se puede observar el tiempo de vida del sistema es de 143.8 segundos, lo cual indica que este escenario es el de mejor desempeño en cuanto a tiempo de vida hasta ahora, y comparado con el primer escenario la diferencia es significativa (~12%). Se podría pensar que el emplear una sola ruta para enviar el tráfico sería mejor que enviarlo por rutas múltiples de forma concurrente, pero esto depende del propósito deseado. Por ejemplo, si se desea una red robusta y segura, es mejor emplear enrutamiento por trayectorias múltiples, mientras que si solamente se desea incrementar el tiempo de vida de la red, se podría emplear este escenario en donde el tráfico se envía por la mejor ruta descubierta, cabe hacer mención de que a pesar de que se emplea una sola ruta para enviar la información, este escenario sigue haciendo uso del enrutamiento con trayectorias múltiples. Además, cabe mencionar que este escenario conserva algunos ingredientes de nuestra propuesta, ya que usa una de las rutas descubiertas y decide cuál es la mejor en base al peso asignado. Esto, por lo tanto,

comprueba que el método de asignación de pesos propuesto hace su tarea de manera adecuada.

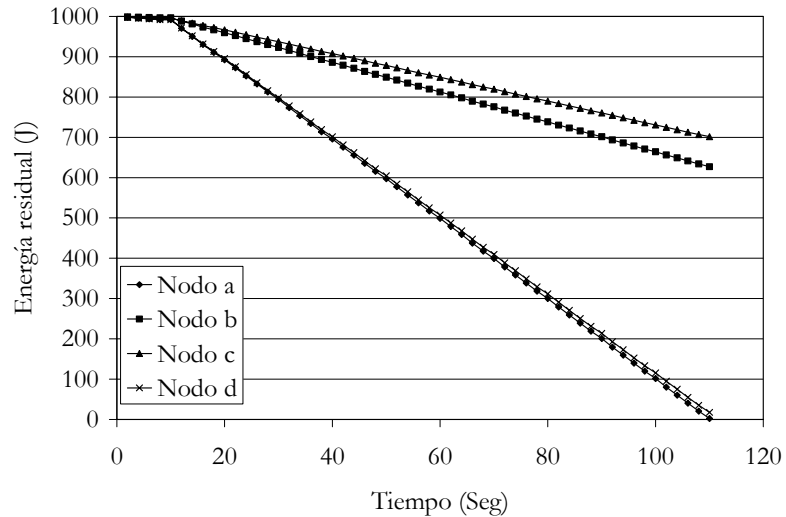


Figura 24 - Energía residual en los nodos *a*, *b*, *c* y *d*. Escenario GLS modificado todas las rutas el mismo peso.

En la figura 24 se muestran los resultados obtenidos para el escenario en el que se le asigna el mismo peso a todas las rutas. Este escenario carece de la asignación de pesos de forma inteligente, que es una característica de nuestra propuesta. Debido a la incapacidad de realizar lo anteriormente dicho, el tiempo de vida del sistema es de 110.1 segundos. El primer escenario lo mejora en 16% aproximadamente, el segundo lo mejora en 27% aproximadamente y el tercer escenario lo mejora en 31% aproximadamente, lo cual demuestra la importancia de la asignación inteligente de los pesos.

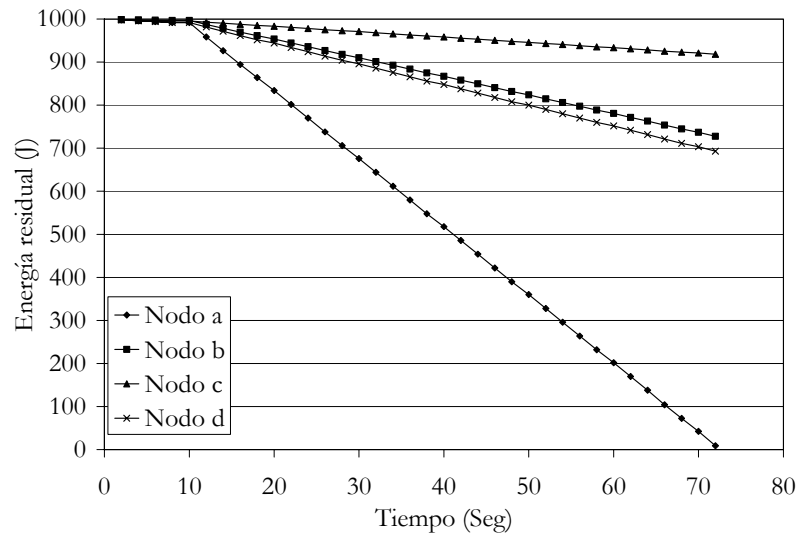


Figura 25 - Energía residual en los nodos *a*, *b*, *c* y *d*. Escenario que usa la ruta más corta.

En la figura 25 se muestran los resultados obtenidos para el escenario que usa la ruta más corta, es decir, el tráfico se envía siempre por la ruta primaria. Los resultados son evidentemente inferiores, ya que al enviar el tráfico sólo por las rutas más cortas se termina abusando de estas y terminado prematuramente con la vida de la red. El tiempo de vida del sistema cae a 72.5 segundos, desempeño que es superado ampliamente por los demás escenarios.

III.2.2 GLS CON DIFUSIÓN

Ahora que ya se han analizado y visto los resultados obtenidos de la simulación de nuestra propuesta GLS modificado, se presentan los resultados obtenidos de la variante GLS con difusión. La premisa, como se mencionó en el capítulo anterior fue que la introducción del mecanismo de difusión al algoritmo GLS modificado mejoraría la distribución de la carga

de tráfico en la red de sensores, ya que en el algoritmo GLS modificado se introducía inherentemente cierto nivel de desbalance de carga en la red.

En la figura 26 podemos observar el índice de equidad contra β . Como puede verse, los valores obtenidos para este experimento del índice de equidad no bajan de 0.98 al variar β . De hecho, si comparamos los resultados obtenidos para GLS modificado y GLS con difusión, el máximo que se logra en GLS con difusión es de 0.99, mientras que en GLS modificado el máximo que se logra es de aproximadamente 0.97, dos centésimas que pueden significar un ahorro de energía considerable con respecto al tiempo, y por supuesto un poco más de vida para la red. También observamos que el valor mínimo para el índice de equidad que se obtiene en GLS con difusión es de 0.98, mientras que con GLS modificado se obtiene un mínimo de 0.95, una diferencia de tres centésimas, con lo que se demuestra que con la introducción del mecanismo de difusión al algoritmo GLS modificado se logra esparcir mejor el tráfico a través de la red.

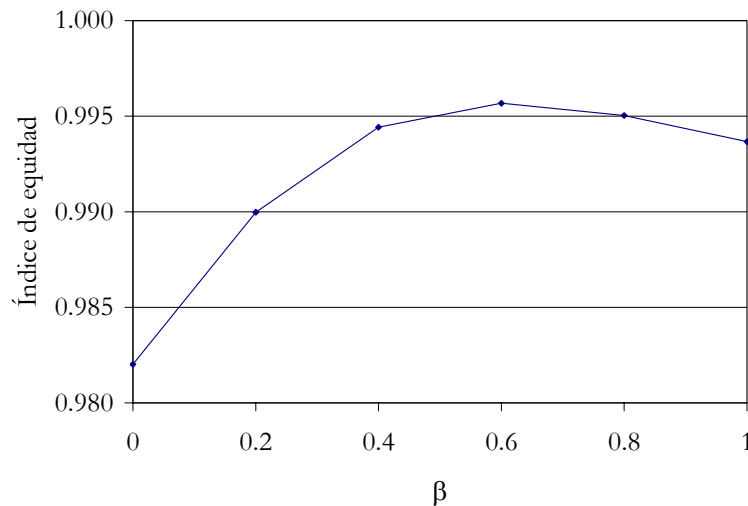


Figura 26 - Índice de equidad contra β , GLS con difusión.

También se obtuvieron resultados del índice de equidad contra la energía residual del nodo a . Estos resultados son sólo para confirmar la mejora que experimenta el índice de equidad con el algoritmo GLS con difusión.

En la figura 27 se puede observar si se compara con la figura 17, que los valores del índice de equidad son mayores con GLS con difusión que para GLS modificado. Cuando se tiene que la energía residual del nodo a es del 0% el índice de equidad es de 0.78, mientras que para GLS con difusión se tiene un valor de aproximadamente 0.9, lo cual es una gran mejora ya que la diferencia es de aproximadamente 12 centésimas, cuando el nodo sensor a ha perdido sus funcionalidades. Con esto se obtiene la ventaja de poder seguir esparciendo el tráfico en la red de sensores de una manera más equitativa a pesar de que uno de los cuatro nodos más importantes en nuestra topología quede sin funcionalidad.

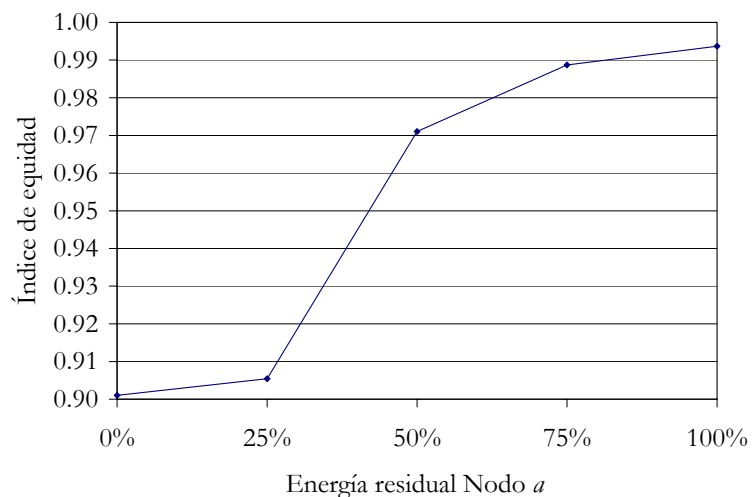


Figura 27 - Índice de equidad contra energía residual del nodo a , GLS con difusión.

Con estos resultados se comprueba que la introducción del mecanismo de difusión al algoritmo GLS modificado mejora la distribución de la carga en la red de sensores. Otra prueba de lo anterior se puede observar en la figura 28.

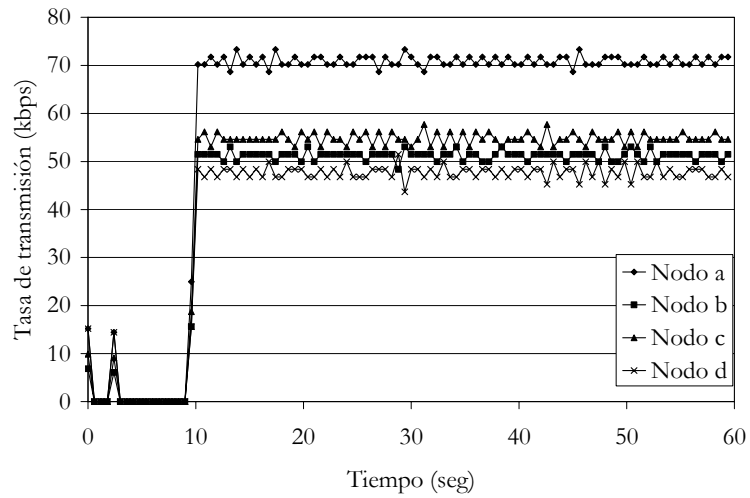


Figura 28 - Tasa de transmisión de los nodos *a*, *b*, *c* y *d*. GLS con difusión.

En la figura 28 se pueden observar las tasas de transmisión de los nodos *a*, *b*, *c* y *d* en GLS con difusión. Ha de notarse como los nodos *b*, *c* y *d* cuentan con una tasa de transmisión aproximada, lo que indica que en estos nodos cuello de botella se está logrando un balanceo de carga. El nodo *a* cuenta con una tasa de transmisión mayor que los otros tres nodos, esto debido al desbalance que introduce la misma configuración de la red.

Otra métrica empleada en GLS con difusión es el tiempo de vida del sistema. Los resultados obtenidos se muestran en la figura 29.

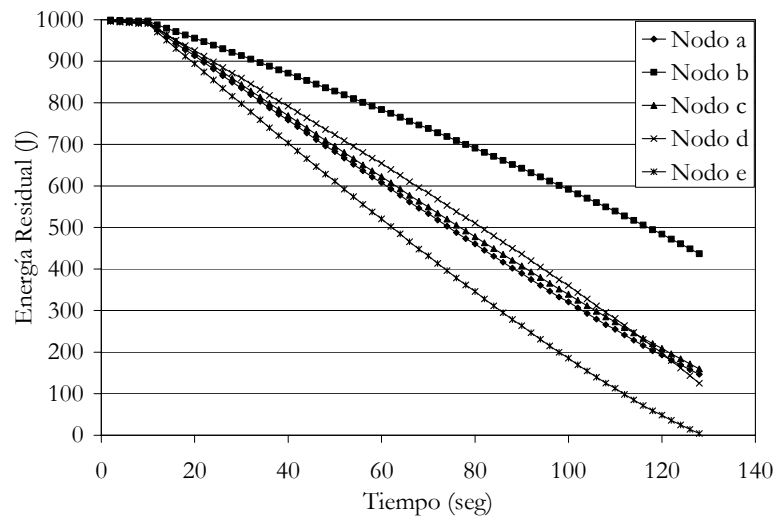


Figura 29 - Energía residual en los nodos *a*, *b*, *c*, *d* y *e*. Escenario GLS con difusión.

Se puede observar que el tiempo de vida del sistema al emplear GLS con difusión es de 129.3 segundos. Si lo comparamos con el tiempo de vida obtenido del escenario de GLS modificado la diferencia es mínima (1.8 segundos). Esto se podría deber a la configuración de red seleccionada en la que, como se mencionó antes, hay de entrada un alto desbalance hacia los nodos *a* y *d*, a tal grado que no puede ser corregido ni con el método de difusión.

Ahora para probar la efectividad de GLS con difusión se puso a prueba en otro escenario. En este escenario un nodo, digamos *y*, que genera tráfico como si juntara lo generado por 10 nodos. La configuración de la red es tal que ese nuevo nodo sólo puede encontrar una ruta hacia el sumidero debido a que sólo tiene un vecino. Esta configuración es implementada y se compara el tiempo de vida del sistema en los escenarios GLS modificado con actualizaciones de ciclo, GLS modificado utilizando solo la mejor ruta y GLS con difusión, los resultados obtenidos se muestran a continuación.

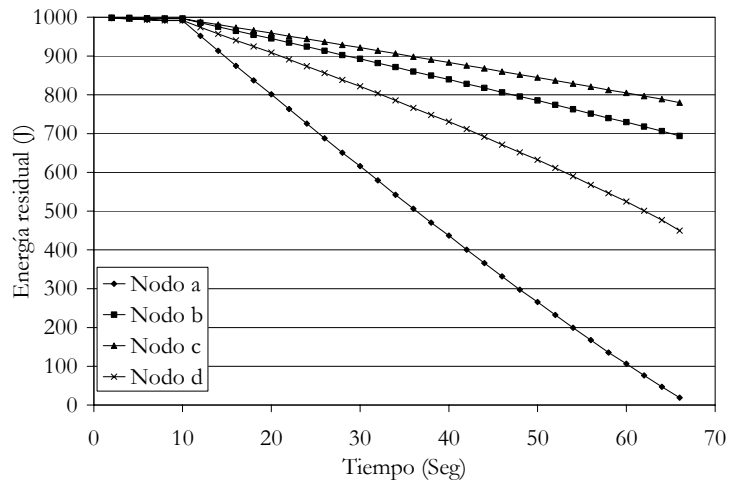


Figura 30 - Energía residual de los nodos *a*, *b*, *c* y *d*, escenario GLS modificado con actualización de ciclo cada 5 segundos, nodo *y* en la configuración de red.

Como podemos apreciar el tiempo de vida del sistema en el escenario GLS modificado con actualización de ciclo es de 67.5 segundos.

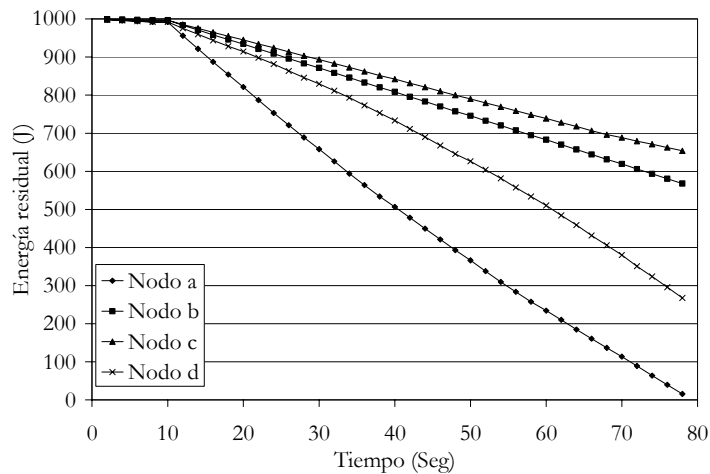


Figura 31 - Energía residual de los nodos *a*, *b*, *c* y *d*, escenario GLS modificado utilizando solo la mejor ruta, nodo *y* en la configuración de red.

Ahora se tiene que el sistema tiene un tiempo de vida de 79.3 segundos para el escenario GLS modificado utilizando solo la mejor ruta.

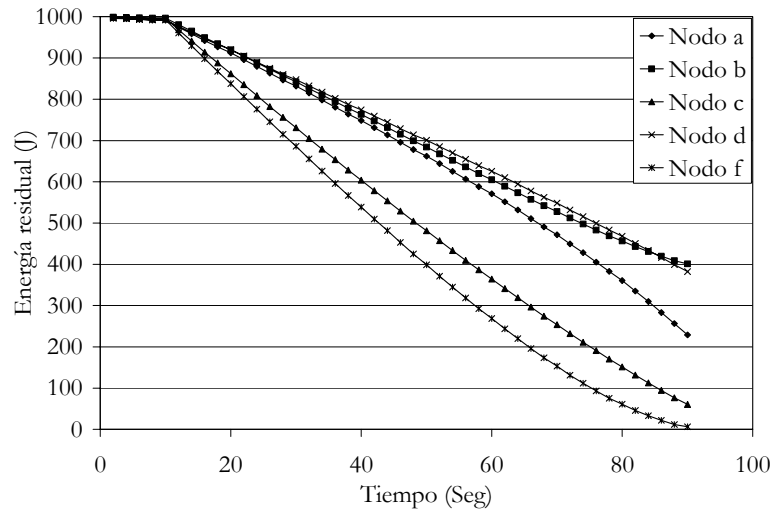


Figura 32 - Energía residual de los nodos a, b, c, d y f, escenario GLS con difusión, nodo y en la configuración de red.

Ahora el tiempo de vida del sistema es de 91.8 segundos para el escenario GLS con difusión, si comparamos este resultados con los obtenidos en los otros dos escenarios donde se incluye al nodo y se tiene que GLS con difusión mejora en un 36% a GLS modificado con actualización de ciclo y supera en aproximadamente 16% el desempeño de tiempo de vida que se tiene en el escenario GLS modificado utilizando solo la mejor ruta, esto al incluir al nodo y en la configuración de red, en la anterior configuración, en donde no se incluía al nodo y, no se pueden ver claramente los beneficios de emplear GLS con difusión, pero a través de esta configuración de red se ha comprobado la utilidad de GLS con difusión, al obtener el mejor desempeño en tiempo de vida del sistema de entre los demás escenarios comparados.

III.3 RESUMEN DEL CAPÍTULO

En este capítulo se han presentado los resultados obtenidos para los algoritmos GLS modificado y GLS con difusión.

Primeramente se comenzó describiendo brevemente las características y funcionamiento de la herramienta de simulación empleada para poner a prueba nuestra propuesta. Se mostró la jerarquía empleada en OPNET para implementar los modelos y así realizar las simulaciones. Esta jerarquía consta de un modelo de red, un modelo de nodos y un modelo de procesos. Se presentó el modelo específico de nuestras propuestas en OPNET.

En otra parte de este capítulo se pudieron observar los resultados obtenidos para nuestras propuestas. Se presentaron resultados que demuestran que el algoritmo GLS modificado trabaja bien en términos de equidad y de eficiencia en el uso de energía.

Además de esto se comprobó que la acción del protocolo de enrutamiento por trayectorias múltiples junto al algoritmo GLS modificado daba robustez al sistema, ya que al hacer fallar a un nodo, el sistema responde considerablemente rápido a la falla. Específicamente se observó que en aproximadamente 6 segundos se redistribuye la carga en la red sin necesidad de ejecutar de nuevo los procedimientos de descubrimiento de rutas, siendo esta una muy buena característica para nuestra propuesta. Además, al dar importancia al número de saltos a la hora de asignar los pesos a las rutas podemos ahorrarnos hasta un 12% en el consumo de energía.

También se pudieron observar los resultados obtenidos para nuestra propuesta denominada GLS con difusión. El fin era el de mejorar el índice de equidad. Los resultados obtenidos demuestran que al introducir el mecanismo de difusión a nuestra propuesta, el tráfico se esparce más, permitiendo con esto dar trabajo mejor balanceado a los nodos que se encuentran a un salto del nodo sumidero, consiguiéndose así un índice de equidad más cercano a la unidad.

CAPÍTULO IV

CONCLUSIONES Y TRABAJO FUTURO

IV.1 CONCLUSIONES

Las principales contribuciones de este trabajo son las siguientes:

- Se propuso una metodología mejorada con la cual se descubren múltiples rutas que no se cruzan entre sí, la cual incrementa la confiabilidad y robustez de la red.
- Se propuso una forma de asignar pesos a cada una de las múltiples trayectorias encontradas con lo cual, dependiendo de los cuellos de botella en carga y en energía, así como del factor de importancia que se le desee dar al número de saltos con el que cuenta cada ruta, se define la cantidad de tráfico que será colocada en cada ruta hacia el sumidero.
- Se obtuvo una versión mejorada del algoritmo GLS (GLS modificado), para la repartición del tráfico entre las diferentes rutas dependiendo de su peso, con la cual se logra una mayor robustez mediante la detección rápida de fallas en las rutas a través del uso de acuses de reenvío y una mayor equidad mediante la detección de rutas ocupadas y la introducción de la filosofía úselo o piérdalo.
- Se obtuvo también una versión adicional de GLS, llamada GLS con difusión, con la cual se ha logrado una mejoría adicional en la distribución de la carga en la red, lo cual fue comprobado a través de simulaciones.

Las principales conclusiones que se pueden obtener de los resultados de simulación son las siguientes:

- Los algoritmos corren razonablemente rápido, ya que el descubrimiento de rutas y la asignación de pesos se dan en aproximadamente tres segundos.
- Se logra un balanceo de carga entre los nodos cuello de botella, y además de esto se logra reajustar la carga en caso de que algún nodo falle sin necesidad de iniciar un nuevo ciclo para el descubrimiento de rutas, además que esto lo realiza en un periodo de tiempo razonablemente corto (seis segundos aproximadamente).
- Al tomar en cuenta a los nodos cuello de botella en energía al momento de calcular los pesos y asignarlos a cada ruta, se permite una mejor distribución del consumo de energía cuando los nodos cuenten con bajas reservas.
- Las metodologías propuestas permiten que cada nodo sensor utilice todas las rutas que ha descubierto, con un compromiso entre involucrar tantos nodos como sea posible para la transmisión de datos, pero a la vez evitando en cierta medida el uso de rutas largas (mayor número de saltos).

IV.2 TRABAJO FUTURO

En este trabajo de tesis se ha propuesto una metodología para seleccionar las múltiples rutas que se pueden utilizar entre cada pareja fuente-destino de tráfico, una forma de asignar un peso a cada ruta descubierta, dependiendo del estado extremo a extremo, con lo cual se define la cantidad de tráfico que se enviará por cada una y además una versión mejorada de GLS con la que se reparte más eficientemente la carga entre las diferentes rutas, pero se requiere más trabajo de investigación para mejorar aún más el desempeño de todos estos elementos trabajando juntos. Éstos podrían incluir:

- **Movilidad de los nodos:** Es importante mencionar que la suposición aquí realizada fue que los nodos fueran cuasi-estáticos. Sería bastante útil que las metodologías aquí propuestas se mejoraran para permitir a los nodos sensores tener mayor

movilidad, ya que esto posibilitaría que los nodos sensores se adaptaran mejor al medio ambiente. Un mecanismo de actualización de posición es requerido para permitir que cada nodo esté pendiente de la posición de sí mismo y la posición de sus vecinos constantemente. Aquí el reto es mantener bajo el consumo de energía debido a las constantes actualizaciones de posición.

- **Optimización Cross-Layer:** La comunicación entre redes de sensores inalámbricas está fuertemente influenciada por el medio físico, ya que la calidad de los canales radio varía con el tiempo. El diseño cross-layer es benéfico para la mejora de la calidad de servicio (QoS) y la eficiencia de la red. Tomando en cuenta las condiciones del medio físico, así como información que provee la capa MAC (información acerca del estado de las conexiones de radio) al momento de asignar los pesos a las múltiples rutas, permitiría que las rutas elegidas fueran las que tengan las mejores condiciones. El protocolo podría seleccionar las rutas con la mejor calidad de canal y evitar utilizar rutas con condiciones inestables, además de que permitiría tomar en cuenta la calidad del canal al momento de calcular pesos y colocar el tráfico, logrando con esto efficientar el consumo de energía y prolongar la vida de la red.
- **Heterogeneidad:** En aplicaciones reales habrá dispositivos que cuenten con capacidades diferentes (con respecto a computación, comunicación, y sensado). Esta heterogeneidad de los dispositivos pueden tener un número importante de consecuencias de diseño. Es importante mencionar que en este trabajo se supuso que los nodos contaban con capacidades iguales, o sea, eran homogéneos. Sería interesante entonces adaptar estas metodologías para que pudiera funcionar con dispositivos de diferentes capacidades.
- **Esquema híbrido:** Se podrían obtener las mejores características de GLS modificado y GLS con difusión e implementarlas en una red. La manera en que este esquema híbrido funcionaría es que, por default, el enrutamiento sería como en GLS modificado, pero si un nodo fuente sólo logra descubrir un número muy pequeño de rutas (el umbral podría ser 2) entonces puede pedir a los nodos por los

que pasarán sus paquetes que usen el mecanismo de difusión. Habría que incluir en los paquetes un bit que dijera si se solicita o no la difusión. Un nodo que reciba un paquete que solicita difusión de tráfico ya no respetará la ruta incluida en el paquete sino que decidirá localmente por dónde se irá. Además, si el número de rutas que él conoce es mayor que el umbral mencionado anteriormente, le quitará al paquete la etiqueta de que debe ser difundido, de lo contrario dejará ese bit como estaba.

REFERENCIAS

Adishesu H.; Varghese G. y Parulkar G. 1999. **“An Architecture for Packet-striping Protocols”**, ACM Transactions on Computer Systems, 17(4): 249-287 p.

Al-karaki, J. y Kamal, A. 2004. **"Routing Techniques in Wireless Sensor Networks: A Survey"**, IEEE Wireless Communications, 11(6): 6-28 p.

Baronti, P.; Pillai, P.; Chook, V.; Chessa, S.; Gotta, A. y Hu Y. 2006. **"Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards"**, Reporte técnico 2006-TR-18.

Callaway, E. H. 2004. **"Wireless Sensor networks: Architectures and protocols"**, CRC Press LLC, EUA, 350 pp.

Dai, H. y Han, R. 2003. **"A Node-Centric Load Balancing Algorithm for Wireless Sensor Networks"**, IEEE Global Telecommunications Conference, San Francisco, EUA, 1 – 5 de Diciembre, 1: 548 – 552 p.

Gallardo, J. R.; González, A.; Villaseñor, L. y Sánchez, J. 2007. **"Multipath Routing Using Generalized Load Sharing For Wireless Sensor Networks"**, Wireless and Optical Communications 2007, Montreal, Québec, Canadá, 6 pp.

Ganesan, D.; Govidan, R.; Shenker, S. y Estrin, D. 2000. **"Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks"**, Mobile Computing and Communications Review, 1(2): 1-13 p.

Holger, K. y Willig, A. 2005. **"Protocols and Architectures for Wireless Sensor Networks"**, John Wiley & Sons Ltd, Inglaterra, 497 pp.

Krishnamachari, B. 2005. **"Networking Wireless Sensors"**, 1a edición, Cambridge University Press, EUA, 202 pp.

Lee T. T.; Liew S. C. y Ding Q. L. 1997. **“Parallel Communications for ATM Network Control and Management”**, Performance Evaluation, 30 (4): 243-264 p.

Leung K.C. y Li V.O.K. 2006. **“Generalized Load Sharing for Packet-Switching Networks I: Theory and packet based algorithm”**. IEEE/ACM Trans. On Parallel and Distributed Systems, 17(7): 694-702 p.

Lou W. 2005. **"An Efficient N-to-1 Multipath Routing Protocol in Wireless Sensor Networks"**. Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems(MASS), Washington, DC, 7 – 10 de Noviembre.

Lou W., Liu W. y Zhang Y. 2005. **“Performance Optimization using Multipath Routing in Mobile Ad Hoc and Wireless Sensor Networks”**, En: M. Cheng, Y. Li, D.-Z. Du (Eds.) **“Combinatorial Optimization in Communication Networks”**, Kluwer Academic Publishers, 117- 146 p.

Lu Y. 2005. **"Multipath Routing Algorithm for Wireless Sensor Networks"**. Tesis de maestría, University of British Columbia, Vancouver, Canadá, 79 pp.

Lu Y. y Wong V. 2006. **"An Energy-Efficient Multipath Routing Protocol for Wireless Sensor Networks"**. International Journal of Communication Systems, 20(7): 747- 766 p.

Manual de usuario OPNET. <http://www.opnet.com> [consulta: Enero/2007].

Medina Castro, P. 2004. **“Introducción de Mecanismos de Calidad de Servicio en el Protocolo de Acceso al Medio de Redes Locales Inalámbricas del Tipo IEEE 802.11”**. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, México, 118 pp.

Parekh, A.K. y Gallager, R.G. 1993. **“A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case”**. IEEE/ACM Trans. Networking, 1(3): 344-357 p.

Santi, P. 2005. **"Topology Control in Wireless Ad Hoc and Sensor Networks"**, John Wiley & Sons Inc., 252 pp.

Wu, J.; Dulman, S. y Havinga P. 2004. "**Reliable Splitted Multipath Routing for Wireless Sensor Networks**", IFIP International Conference, NPC 2004, China, 18 – 20 de Octubre.

Var der Lande J. 1998. "**Inverse Multiplexing Over ATM: Broadband Access for Less**", Computer Technology Review, 18(12): 24-26 p.

Zhao, F. y Guibas, L. 2004. "**Wireless Sensor Networks: An Information Processing Approach**", Elsevier Inc., EUA, 358 pp.