INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Algoritmos bioinspirados para el encaminamiento de datos en redes inalámbricas de sensores

TESIS

Que para obtener el grado de
Maestro en Ciencias de la Computación

Presenta
Ing. Christian Horacio Domínguez Medina

Directora de tesis
Dra. Nareli Cruz Cortés

México D.F., 26 de mayo del 2011

SIP-14

INSTITUTO POLITECNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 29 del mes de abril de 2011 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"ALGORITMOS BIOINSPIRADOS PARA EL ENCAMINAMIENTO DE DATOS EN REDES INALÁMBRICAS DE SENSORES"

DOMÍNGUEZ	**MEDINA**	**CHRISTIAN HORACIO**
(Apellido Paterno)	(Apellido Materno)	(Nombre)

Con registro: A 0 9 0 2 4 7

aspirante al grado de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente

Dr. Sergio Suárez Guerra

Secretario

Dr. Edgardo Manuel Felipe Riverón

Primer vocal (Director)

Dra. Nareli Cruz Cortés

Segundo vocal

Dr. Marco Antonio Moreno Armendáriz

Tercer Vocal

Dr. Amadeo José Argüelles Cruz

El Presidente del Colegio

Dr. Luis Alfonso Villa Vargas

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de **México** el día **25** del mes de **mayo** del año **2011**, el que suscribe **Christian Horacio Domínguez Medina** alumno del Programa de **Maestría en Ciencias de la Computación** con número de registro **A090247**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de la **Dra. Nareli Cruz Cortés** y cede los derechos del trabajo intitulado "**Algoritmos bioinspirados para el encaminamiento de datos en redes inalámbricas de sensores**", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Éste puede ser obtenido escribiendo en la siguiente dirección chrisdom27@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

ATENTAMENTE

C. Christian Horacio Domínguez Medina

Agradecimientos

Esta tesis está dedicada a mi padres, Elvia Medina y Rafael Domíguez, a quienes agradezco de todo corazón por su apoyo.

Agradezco a mis hermanos Karen y Oswaldo Domíguez por la compañía y el apoyo que me brindan. Se que cuento con ellos siempre.

Agradezco a mi amiga Tania García porque en todo momento ha sido mi modelo a seguir y me ha motivado incluso en los momentos más difíciles.

Agradezco haber encontrado el amor y la amistad de mi mejor amiga Joany, quien me ayuda a seguir hacia adelante.

Agradezco a mis amigos de generación, a Bety, Richard, Rodrigo, Ale, Juan, Andrés, Enrique, Blanquita, Obdulia, Luis, Juan, Alfonso, Raúl, Bety Aurora y a todos los integrantes del laboratorio de IA, sin ellos hubiera sido mucho más difícil concluir este logro.

Agradezco a mis profesores, por su disposición, ayuda brindada y por mostrarme lo poco que conozco.

Agradezco a mi comité tutorial, al Dr. Marco, Dr. Edgardo, Dr. Amadeo, Dr. Sergio y Dr. Francisco por indicarme el camino a seguir.

Agradezco a la Dra. Nareli Cruz Cortés, guía y amiga.

Agradezco al Instituto Politécnico Nacional, al Centro de Investigación en Computación.

Agradezco al CONACyT por su apoyo a través de los proyectos 132073, 107688 y 60240.

Agradezco a mi país porque espera lo mejor de mí.

Y agradezco a todos aquellos, que por mi mala memoria no pude plasmar sus nombres aquí.

Gracias a todos.

Christian Horacio Domínguez Medina.

Resumen

Las redes inalámbricas de sensores se han convertido en un tópico muy activo de investigación en los últimos años.

El problema del encaminamiento de datos en las redes inalámbricas de sensores es una parte que debe ser tomada muy en cuenta si se desea maximizar el tiempo de vida de la red y minimizar la latencia en la transmisión de datos. Puesto que el tamaño de este tipo de redes puede incrementarse según la aplicación, el método de encaminamiento de datos se vuelve más complejo debido a la cantidad de nodos sensores que se tienen en la red.

Los nodos sensores de estas redes son dispositivos con limitantes considerables tanto en capacidad de cómputo, como en memoria, comunicación inalámbrica y batería.

Por otro lado, los algoritmos de optimización por colonia de hormigas han sido propuestos para tratar de resolver el problema del encaminamiento de datos en redes inalámbricas de sensores.

En este trabajo de tesis, se presenta una comparación de dos algoritmos de encaminamiento de datos basados en la optimización por colonia de hormigas para redes inalámbricas de sensores en diferentes escenarios. Con el objetivo de incrementar el tiempo de vida útil de la red y decrementar la latencia en la transmisión de datos, se propone un algoritmo de encaminamiento de datos que combina las características de ambos algoritmos estudiados. La comparación tiene en cuenta indicadores de niveles de consumo de energía lo más justos y realistas posibles.

Abstract

Wireless Sensor Networks have become an active research topic in the last years.

The routing problem is a very important part of this kind of networks that need to be considered in order to maximize the network life time and minimize data transmissions latency.

As the size of the network increases, the routing problem becomes more complex due the amount of sensor nodes in the network.

These sensor nodes are very constrained in memory capabilities, processing power, wireless communications and batteries.

On the other hand, Ant Colony Optimization based routing algorithms have been proposed to solve the routing problem, trying to deal with the constraints of the sensor nodes.

In this thesis is presented a comparison of two Ant Colony-Based Routing Algorithms, under different scenarios at simulation level.

Furthermore, in order to increase the wireless sensor network life time a new routing algorithm that combines the characteristics of the studied algorithms is defined.

The comparison takes into account current amounts of energy consumption as fair as possible.

Índice general

Agradecimientos **4**

Resumen **5**

Abstract **6**

Siglas y abreviaturas **15**

Glosario **17**

1. Introducción **22**

1.1. Motivación y justificación 22

1.2. Definición del problema 25

1.3. Objetivos 25

1.3.1. General 25

1.3.2. Particulares 25

1.4. Alcances y limitaciones 26

1.5. Apoyos 26

1.6. Contribuciones 26

1.7. Publicaciones en congresos internacionales 26

1.8. Organización de la tesis 27

2. Marco teórico **28**

2.1. Redes inalámbricas de sensores 28

2.1.1. Redes de computadoras 28

2.1.1.1. Redes inalámbricas 30
2.1.1.2. Redes *ad-hoc* o MANET 31
2.1.2. Descripción de las redes inalámbricas de sensores (RIS) 31
2.1.2.1. Aplicaciones 33
2.2. Algoritmos de encaminamiento para RIS 35
2.2.1. Clasificación de algoritmos de encaminamiento para RIS 36
2.2.1.1. Algoritmos basados en la estructura de la red 36
2.2.1.2. Algoritmos basados en el criterio de encaminamiento . 38
2.3. Optimización por colonia de hormigas (ACO) 39
2.3.1. Antecedentes 39
2.3.2. Comportamiento de las hormigas reales 41
2.3.3. Definición del algoritmo de ACO 42

3. Estado del arte **45**

3.1. Esquema general de un algoritmo de encaminamiento basado en ACO para RIS 45
3.2. Algoritmos de encaminamiento basados en ACO para RIS 46
3.2.1. Algoritmo de encaminamiento para el cuidado de la calidad en el servicio (QoS) basado en ACO 46
3.2.2. Algoritmo de difusión directa con cuidado en la energía basado en feromona 47
3.2.3. Algoritmo de encaminamiento adaptativo mejorado 49
3.2.4. Algoritmo de retardo y energía 50
3.2.5. Algoritmo de agregación de datos basado en ACO 51
3.2.6. Algunos otros ejemplos 53

4. Metodología **54**

4.1. Definiciones 54
4.2. Algoritmos seleccionados 55
4.2.1. Algoritmo de encaminamiento consciente de la ubicación basado en ACO (ACLR) 55

4.2.2. Algoritmo de encaminamiento basado en hormigas con eficiencia de energía (EEABR) 59
4.2.3. Comparación entre ACLR y EEABR 61
4.2.3.1. Experimentos 61
4.2.3.2. Resultados 64
4.2.3.3. Conclusiones de la comparación entre ACLR y EEABR 74
4.3. Algoritmo de encaminamiento propuesto para mejorar el tiempo de vida de la RIS (IACAR) 74
4.4. Encaminamiento por inundación dirigida 77

5. Experimentos y resultados **79**
5.1. Experimentos 79
5.2. Resultados 79
5.2.1. Primer escenario 80
5.2.2. Segundo escenario 81
5.2.3. Tercer escenario 82
5.2.4. Cuarto escenario 83

6. Conclusiones y trabajo futuro **85**
6.1. Comentarios finales 85
6.2. Conclusiones 86
6.3. Posibles aplicaciones 87
6.4. Trabajo futuro 88

Referencias **88**

Índice de figuras

1.1. Red inalámbrica de sensores. 23

2.1. Modelo simplificado para las comunicaciones. 29
2.2. Ejemplo de configuración de una LAN inalámbrica de celda única. . . . 30
2.3. Configuraciones de redes LAN inalámbricas. 32
2.4. Componentes hardware de un nodo sensor. 33
2.5. Algoritmos de encaminamiento en RIS. 37
2.6. Comportamiento de las hormigas reales. 42

3.1. Inicio del algoritmo de encaminamiento basado en ACO. 45
3.2. Dinámica de los niveles de feromona de la RIS. 46
3.3. ACO-QoSR - Proceso de descubrimiento de rutas. 47
3.4. Difusión directa - Idea básica. 48

4.1. Radio de cobertura de los nodos sensores. 55
4.2. Primer escenario - Posiciones de s_o y s_b. 62
4.3. Segundo escenario - Posiciones de s_o y s_b. 62
4.4. Tercer escenario - Posiciones de s_o y s_b. 63
4.5. Primer escenario. Consumo de energía. 66
4.6. Primer escenario. Eficiencia de energía. 66
4.7. Primer escenario. Latencia. 67
4.8. Primer escenario. Consumo de energía de ACLR con diferente número de nodos desplegados. 68
4.9. Primer escenario. Consumo de energía de EEABR con diferente número de nodos desplegados. 68

4.10. Primer escenario. Latencia de ACLR con diferente número de nodos desplegados. 68

4.11. Primer escenario. Latencia de EEABR con diferente número de nodos desplegados. 68

4.12. Primer escenario. Eficiencia de energía de ACLR con diferente número de nodos desplegados. 68

4.13. Primer escenario. Eficiencia de energía de EEABR con diferente número de nodos desplegados. 68

4.14. Segundo escenario. Consumo de energía. 69

4.15. Segundo escenario. Eficiencia de energía. 69

4.16. Segundo escenario. Latencia. 69

4.17. Segundo escenario. Consumo de energía de ACLR con diferente número de nodos desplegados. 70

4.18. Segundo escenario. Consumo de energía de EEABR con diferente número de nodos desplegados. 70

4.19. Segundo escenario. Latencia de ACLR con diferente número de nodos desplegados. 71

4.20. Segundo escenario. Latencia de EEABR con diferente número de nodos desplegados. 71

4.21. Segundo escenario. Eficiencia de energía de ACLR con diferente número de nodos desplegados. 71

4.22. Segundo escenario. Eficiencia de energía de EEABR con diferente número de nodos desplegados. 71

4.23. Tercer escenario. Consumo de energía. 71

4.24. Tercer escenario. Eficiencia de energía. 71

4.25. Tercer escenario. Latencia. 72

4.26. Tercer escenario. Consumo de energía de ACLR con diferente número de nodos desplegados. 72

4.27. Tercer escenario. Consumo de energía de EEABR con diferente número de nodos desplegados. 72

4.28. Tercer escenario. Latencia de ACLR con diferente número de nodos desplegados. 73

4.29. Tercer escenario. Latencia de EEABR con diferente número de nodos desplegados. 73

4.30. Tercer escenario. Eficiencia de energía de ACLR con diferente número de nodos desplegados. 73

4.31. Tercer escenario. Eficiencia de energía de EEABR con diferente número de nodos desplegados. 73

5.1. Primer escenario. Consumo de energía. 80

5.2. Primer escenario. Eficiencia de energía. 80

5.3. Primer escenario. Latencia. 80

5.4. Segundo escenario. Consumo de energía. 82

5.5. Segundo escenario. Eficiencia de energía. 82

5.6. Segundo escenario. Latencia. 82

5.7. Tercer escenario. Consumo de energía. 83

5.8. Tercer escenario. Eficiencia de energía. 83

5.9. Tercer escenario. Latencia. 83

5.10. Consumo de energía en el cuarto escenario. 84

5.11. Eficiencia de energía en el cuarto escenario. 84

5.12. Latencia en el cuarto escenario. 84

Índice de tablas

2.1. Comparación de nodos. 34

2.2. Clasificación y comparación de algoritmos de encaminamiento para RIS [1]. 39

3.1. IAR - Estrutura de la tabla de encaminamiento para el nodo k. 49

4.1. Tamaño de rutas escenario 1. 64

4.2. Promedio del tamaño de rutas por número de nodos desplegados. . . . 65

4.3. Tamaño de rutas escenario 2. 65

4.4. Tamaño de rutas escenario 3. 66

5.1. Escenario cuatro. 79

Índice de algoritmos

1. Enfoque estándar de ACO. 43
2. Algoritmo ACLR. 58
3. Algoritmo EEABR. 61
4. Algoritmo IACAR. 77

Siglas y abreviaturas

ACAR: Algoritmo basado en Ant Colony Optimization con agregación de datos (*Data Aggregation Based on ACO algorithm*).

ACLR: Algoritmo de encaminamiento consciente de la ubicación basado en ACO (*Ant Colony Optimization-based Location-Aware Routing*).

ACO: Optimización por colonia de hormigas (*Ant Colony Optimization*).

ACO-QoSR: Algoritmo de encaminamiento para el cuidado de la calidad en el servicio (*ACO-based QoS Routing algorithm*).

APTEEN: Protocolos eficientes en energía sensibles a umbral con adaptación periódica (*Threshold Energy Efficient Sensor Network Protocol*).

CADR: Encaminamiento de difusión anisótropa restringida (*Constrained Anisotropic Diffusion Routing*).

CM: Módulo de control (*Control Module*).

DD: Difusión directa.

DFRF: Sistema de encaminamiento por inundación dirigida (*Directed Flood-Routing Framework*).

E&D: Algoritmo de retardo de energía (*Energy Delay algorithm*).

EAR: Encaminamiento consciente de la energía (*Energy Aware Routing*).

EEABR: Algoritmo de encaminamiento basado en hormigas con eficiencia de energía (*Energy-Efficient Ant-Based Routing algorithm*).

GAF: Fidelidad adaptativa geográfica (*Geographic Adaptive Fidelity*).

GEAR: Encaminamiento geográfico basado en ahorro de energía (*Geographic and Energy Aware Routing*).

GOAFR: Algoritmo de encaminamiento geométrico con la mezcla del encaminamiento por gradiente y por caras (*Greedy Other Adaptive Face Routing*).

GPS: Sistema de posicionamiento global (*Global Positioning System*).

GPSAL: GPS *Ant-Like*.

HPAR: Encaminamiento jerárquico de ahorro de energía (*Hierarchical Power-Aware Routing*).

IACAR: Algoritmo mejorado de encaminamiento basado en ACO para WSN (*Improved Ant Colony-Based Routing algorithm for Wireless Sensor Networks*).

IAR: Algoritmo de encaminamiento adaptativo mejorado (*Improved Adaptive Routing algorithm*).

LAN: Red de área local (*Local Area Network*).

LEACH: Jerarquía de *clustering* adaptativa de baja energía (*Low Energy Adaptive Clustering Hierarchy*).

MAN: Red de área metropolitana (*Metropolitan Area Network*).

MANET: Redes móviles *ad-hoc* (*Mobile and Ad-hoc NETworks*).

MCFA: Algoritmo de reenvío de costo mínimo (*Minimum Cost Forwarding Algorithm*).

MMAS: Sistema de hormigas MAX-MIN (*MAX-MIN Ant System*).

OA: Ómicron ACO.

PEGASIS: Concurrencia eficiente en potencia en sistemas de información de sensores (*Power-Efficient Gathering in Sensor Information Systems*).

PH: Concentración de iones de hidrógeno.

PRNET: Red de radio paquetes (*Packet Radio NETwork*).

QoS: Calidad en el servicio (*Quality of Service*).

RIS: Rede(s) Inalámbrica(s) de Sensores (*Wireless Sensor Network(s)*).

SOP: Protocolo de auto-organización (*Self Organizing Protocol*).

SPAN: Algoritmo coordinador consciente de la eficiencia de energía (*Energy-efficient Coordination Algorithm*).

SPIN: Protocolos de sensores para información vía negociación (*Sensor protocols for Information via Negotiation*).

TSP: Problema del agente viajero (*Travelling Salesman Problem*).

UM: Módulo de usuario (*User Module*).

WAN: Red de área amplia (*Wide Area Network*).

Glosario

Acceso Múltiple por Detección de Portadora (CSMA): En telecomunicaciones, se entiende como el escuchar el medio para saber si existe presencia de portadora en los momentos en los que se ocupa el canal. El fin es evitar colisiones, es decir que dos *host* hablen al mismo tiempo. Por otro lado define el procedimiento que estos dos *host* deben seguir si llegasen a usar el mismo medio de forma simultánea.

Acceso nómada: Enlace inalámbrico entre un concentrador (*hub*) de una LAN y un dispositivo terminal móvil equipado con una antena para comunicación inalámbrica (computadora, teléfono, etcétera). Un ejemplo de la utilidad de este tipo de conexiones es posibilitar a un empleado que vuelve de un viaje la transferencia de datos desde una computadora personal portátil a un servidor en la oficina.

***Ad-hoc*:** Locución latina que significa literalmente "para esto". Generalmente se refiere a una solución elaborada específicamente para un problema o fin preciso y, por tanto, no es generalizable ni utilizable para otros propósitos.

Algoritmo símplex: Descubierto por el matemático norteamericano George Bernard Dantzig en 1947, es una técnica popular para dar soluciones numéricas al problema de la programación lineal. El método usa el concepto de un *simplex*, que es un politopo de $N + 1$ vértices en N dimensiones: Un segmento de línea sobre una línea, un triángulo sobre un plano, un tetraedro en un espacio de tres dimensiones y así sucesivamente.

ARPANET: es una red de computadoras que fue creada por encargo del Departamento de Defensa de los Estados Unidos como medio de comunicación para los diferentes organismos del país. El primer nodo se creó en la Universidad de Los Ángeles California y fue la espina dorsal de Internet hasta 1990, tras finalizar la transición al protocolo TCP/IP iniciada en 1983.

Asíncrono: Suceso que no tiene lugar en total correspondencia temporal con otro suceso.

Batimetría: Estudio de la profundidad marina, de la tercera dimensión de los fondos lacustres o marinos. Un mapa o carta batimétrica normalmente muestra el relieve del fondo o terreno como isogramas, puede también dar información adicional de la navegación en superficie.

***Broadcast*:** Es la transmisión de un paquete de datos a todos los dispositivos en una red.

***buffer*:** Una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital, mientras se espera para que sea procesada.

***cluster*:** Conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.

Computación distribuida o informática en malla: Nuevo modelo para resolver problemas de computación masiva con el uso de un gran número de computadoras organizadas en racimos incrustados en una infraestructura de telecomunicaciones distribuida.

Computación *grid*: Tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están sujetos a un control centralizado.

Convergencia prematura: En el campo de la computación evolutiva se refiere al fenómeno de convergencia de la población antes del tiempo necesario para llegar al óptimo global. Esto provoca que el resultado del proceso de búsqueda arroje como solución un resultado subóptimo (generalmente un óptimo local). Este problema es en la práctica difícil de detectar, pues muchas veces no se conoce a priori el valor del óptimo global.

Correlación: Fuerza y dirección de una relación lineal entre dos variables. Se considera que dos variables cuantitativas tienen una correlación cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra.

Datagrama: Es una de las formas de encaminamiento de paquetes en una red de conmutación de paquetes. En la técnica de datagrama cada paquete se trata de forma independiente, donde cada uno contiene la dirección de destino. La red puede encaminar (mediante un enrutador) cada fragmento hacia el equipo terminal.

Estigmergia: Colaboración a través del medio físico. En sistemas descentralizados, tales como las colonias de hormigas, los diferentes componentes colaboran a través de pautas o hitos dejados en el medio: Feromonas, acumulación de objetos o cualquier otro tipo de cambio físico, como la temperatura. El concepto de estigmergia fue introducido por Pierre-Paul Grassé, un estudioso de las hormigas, para explicar cómo se lograban realizar las tareas en insectos sociales sin necesidad de planificación ni de un poder central. Actualmente ha sido tomado y extendido a una serie de algoritmos que forman parte de la inteligencia artificial.

Estocástico: Sistema que funciona, sobre todo, por el azar. Significa perteneciente o relativo al azar.

Etología: Del griego ethos, "costumbre", es la rama de la biología y de la psicología experimental que estudia el comportamiento de los animales en libertad o en condiciones de laboratorio, aunque son más conocidos por los estudios de campo. Los científicos dedicados a la etología se denominan etólogos. La etología corresponde al estudio de las características distintivas de un grupo determinado y cómo éstas evolucionan para

la supervivencia del mismo. La etología es la ciencia que tiene por objeto de estudio el comportamiento animal. Los seres humanos, también animales, forman parte del campo de estudio de la etología. Esta especialización se conoce con el nombre de etología humana.

Feromona: Sustancia química secretada por un individuo con el fin de provocar un comportamiento determinado en otro individuo de la misma u otra especie. Son por tanto un medio de señales cuyas principales ventajas son el gran alcance y la evitación de obstáculos, puesto que son arrastradas por el aire.

Fibra óptica: Medio de transmisión empleado habitualmente en redes de datos; un hilo muy fino de material transparente, vidrio o materiales plásticos, por el que se envían pulsos de luz que representan los datos a transmitir. El haz de luz queda completamente confinado y se propaga por el núcleo de la fibra con un ángulo de reflexión por encima del ángulo límite de reflexión total, en función de la ley de Snell. La fuente de luz puede ser un láser o un LED.

FIFO: Es el acrónimo en inglés de *first in, first out* (primero en entrar, primero en salir). Un sinónimo de FIFO es FCFS, acrónimo en inglés de *first come first served* (primero en llegar, primero en ser servido). Es un método utilizado en estructuras de datos, contabilidad de costos y teoría de colas. Guarda analogía con las personas que esperan en una cola y son atendidas en el orden en que llegaron, es decir, que la primera persona que entra es la primera persona que sale.

Grafo ponderado: Es aquel al que se le atribuye a cada arista un número específico, llamado valuación, ponderación o costo según el contexto.

Heurística: Capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como el arte y la ciencia del descubrimiento y de la invención o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente. Una heurística es un algoritmo que normalmente encuentra soluciones factibles, aunque no hay pruebas de que la solución no pueda ser errónea en algunos casos y que se ejecuta en un tiempo razonable, aunque no existe tampoco prueba de que siempre será así. En [2] se muestra un análisis donde se trata de establecer cúal es la probabilidad de encontrar una solución buena en un tiempo límite.

Isotropía: Característica de los cuerpos cuyas propiedades físicas no dependen de la dirección. Es decir, se refiere al hecho de que ciertas magnitudes vectoriales conmensurables, dan resultados idénticos con independencia de la dirección escogida para dicha medida.

MAC (*medium access control*): Conjunto de algoritmos y métodos de comprobación encargados de regular el uso del medio físico por los distintos dispositivos que lo comparten en la red.

Módem: Convierte las cadenas de bits generadas por una computadora personal y las transforma en señales analógicas que pueden ser transmitidas a través de una red y viceversa.

Multiplexado por División de Tiempo (TDM): Técnica que permite la transmisión de señales digitales y cuya idea consiste en ocupar un canal (normalmente de gran capacidad) de trasmisión a partir de distintas fuentes, de esta manera se logra un mejor aprovechamiento del medio de trasmisión. El Acceso múltiple por división de tiempo (TDMA, por sus siglas en inglés *time division multiple access*) es una de las técnicas de TDM más difundidas.

Optimización: En matemáticas se refiere al proceso de encontrar los mínimos y máximos de una función. Algunas de sus ramas son: Optimización combinatoria y optimización multiobjetivo.

***Overhead*:** Son los bytes adicionales que se deben transmitir como cabeceras (*headers*) en los paquetes de información que circulan en la red.

Paquete de datos: Unidad fundamental de transporte de información en todas las redes de computadoras modernas. El término datagrama es usado a veces como sinónimo. Un paquete está generalmente compuesto de tres elementos: Una cabecera (*header*) que contiene la información necesaria para trasladar el paquete desde el emisor hasta el receptor, el área de datos (*payload*) que contiene los datos que se desean trasladar, y la cola (tráiler), que comúnmente incluye código de detección de errores.

Procedimientos metaheurísticos: Clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos que combinan diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos.

Protocolo: Conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.

Redes de área amplia: (*Wide area network*, WAN) Redes que cubren una extensa área geográfica, requieren atravesar rutas de acceso público y utilizan, al menos parcialmente, circuitos proporcionados por una entidad proveedora de servicios de telecomunicación. Generalmente, una WAN consiste en una serie de dispositivos de conmutación interconectados.

Redes de área local: (*Local area network*, LAN) Es la interconexión de varias computadoras y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de 200 metros, con repetidores podría llegar a la distancia de un campo de 1 kilómetro. Su aplicación más extendida es la interconexin de computadoras personales y estaciones de trabajo en oficinas, fábricas, etc.

Redes de área metropolitana: (*Metropolitan area network*, MAN) Redes de alta velocidad (banda ancha) que da cobertura en un área geográfica extensa, proporciona capacidad de integración de múltiples servicios mediante la transmisión de datos, voz y vídeo, sobre medios de transmisión tales como fibra óptica y par trenzado.

Redes punto a punto: Son aquellas que responden a un tipo de arquitectura de red en las que cada canal de datos se usa para comunicar únicamente dos nodos, en contraposición a las redes multipunto.

Repetidor: Dispositivo electrónico que recibe una señal débil o de bajo nivel y la retransmite a una potencia o nivel más alto, de tal modo que se puedan cubrir distancias más largas sin degradación o con una degradación tolerable. El término repetidor se creó con la telegrafía y se refería a un dispositivo electromecánico utilizado para regenerar las señales telegráficas. El uso del término ha continuado en telefonía y transmisión de datos.

***Router*:** Dispositivo de encaminamiento que se utiliza para conectar dos redes que pueden o no ser similares.

Sensor: Dispositivo capaz de medir magnitudes físicas o químicas y transformarlas en variables eléctricas.

Sensor incrustado: Dispositivo diseñado para realizar una o algunas pocas funciones dedicadas, generalmente en sistemas de tiempo real.

Servidor: Computadora que forma parte de una red y provee servicios a otras computadoras denominadas clientes.

Sistema final: Dispositivo conectado a una de las redes de Internet que se utiliza para apoyar a las aplicaciones o servicios del usuario final.

WPAN: Es una red inalámbrica de área personal para la comunicación entre distintos dispositivos (tanto computadoras, puntos de acceso a Internet, teléfonos celulares, PDA, dispositivos de audio, impresoras) cercanos al punto de acceso.

Capítulo 1

Introducción

En este capítulo se ubica al lector en el lugar donde se realiza la investigación, la motivación y justificación de este trabajo, así como se define el problema que se va a atacar, el objetivo general y los particulares de este proyecto. Finalmente, se explica brevemente el contenido de la tesis.

1.1. Motivación y justificación

Existe una gran gama de aplicaciones donde es indispensable tener cuidado y llevar un control de los cambios en el entorno, los fenómenos del medio ambiente, por ejemplo monitoreo ambiental, detección de intrusos, seguridad, control de inventario, control y monitoreo médico, detección de condiciones del medio ambiente como temperatura, movimiento, sonido, intensidad luminosa, presencia de objetos, etcétera [3]. Comúnmente este tipo de trabajo ha sido realizado por una persona.

Con el avance de la tecnología, ha sido posible la introducción de los sistemas de información en sustitución de las personas, con lo que se obtienen mejoras de detección, regularidad y precisión en el monitoreo de variables del entorno en cuestión.

En la actualidad el campo de investigación de las redes inalámbricas de sensores (RIS) ha tenido un gran impulso en su uso en la resolución de este tipo de problemas de monitoreo, primordialmente por el avance en el diseño y fabricación de los sensores inalámbricos, cuya tarea principal es la captura y transmisión de información del medio ambiente.

Una RIS consiste en un gran número de sensores, que tienen la capacidad de comunicarse entre ellos vía inalámbrica, desplegados en el área que se desea monitorear. En la Figura 1.1 podemos observar una distribución típica de una RIS.

Este tipo de sensores son pequeños, relativamente baratos, de bajo consumo de energía, con memoria y energía limitadas, que además, son capaces tanto de procesar información localmente como de comunicarse de manera inalámbrica. De acuerdo con la aplicación para la cual sea diseñada la RIS, se deben definir las capacidades de captura

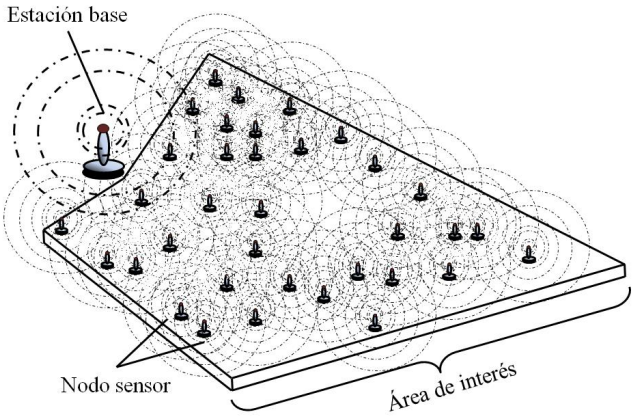

Figura 1.1: Red inalámbrica de sensores.

de datos de los sensores, las cuales pueden ser por ejemplo, capacidades de captura de temperatura, presión, humedad, movimiento, sonido, intensidad luminosa, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, pH, etcétera.

Cada sensor captura información del medio ambiente, y en su caso, es capaz de realizar algún proceso o cálculo a los datos obtenidos. Dicha información es requerida por un sistema central de control, encargado de realizar las operaciones pertinentes de reacción a las medidas capturadas en el entorno. Dicho sistema central de control está conectado a un sensor llamado estación base o punto de acceso, el cual tiene capacidades tanto de cómputo como de memoria mayores a los de los sensores comunes [4].

Cada sensor al capturar información del medio monitoreado (sensor origen), crea un paquete de información en memoria, que en dependencia de la aplicación puede contener diferentes datos, como el identificador del sensor, información de control, los datos capturados por el sensor, etcétera. Basado en el protocolo de comunicación utilizado por la RIS, se establece la ruta por la cual debe pasar el paquete desde el sensor donde fue capturada la información, hasta la estación base. La ruta no es más que la secuencia de los sensores a través de los cuales debe pasar el paquete para poder llegar a su destino.

El entorno o medio ambiente donde se despliegan los sensores de las RIS podría ser un lugar de difícil acceso, por lo que el cambio o sustitución de las baterías de los sensores de la red no es muy frecuente o incluso no se puede realizar, lo que provoca que al momento en que un sensor ha consumido toda su energía, se declare que ese sensor ha muerto[1] (ya no es útil para la red) y por lo tanto, la distribución de los sensores de la red cambia.

Cuando se diseña el algoritmo de encaminamiento de datos de una RIS, se debe es-

[1]Un sensor muerto es aquel que ya no es útil a la RIS. Las razones por las que un nodo muere pueden ser: Que el nodo sensor ha consumido toda su batería, que el nodo quedó incomunicado con respecto a los demás nodos de la red o que existe un fallo en el nodo que le impida realizar su tarea, etcétera.

tablecer el nivel de importancia entre dos puntos clave para el buen desempeño del mismo, el **tiempo de vida de la red** y el **retardo** o **latencia** en la transmisión de los datos.

El **tiempo de vida de la red** se refiere al tiempo en que la red es útil y depende de la aplicación donde se utiliza. Posiblemente la red deje de ser útil por ejemplo, al momento en que un sensor muera, o cuando la mitad de los sensores que componen la RIS mueran o estén incomunicados, o simplemente, cuando la estación base no tenga sensores con los que se pueda comunicar, es decir, cuando la estación base queda aislada o incomunicada, por lo tanto, el tiempo de vida de una RIS va relacionado directamente con la cantidad de energía de los sensores.

El **retardo** o **latencia** es el otro punto clave a valorar en el diseño del algoritmo de encaminamiento de una RIS; se refiere al tiempo que le toma a la información capturada del entorno para llegar desde el sensor origen hasta la estación base. Según la aplicación, el retardo puede ser vital, por ejemplo, que se requiera tener la información lo más rápido posible. Claramente se observa que, si se desea tener la información lo más pronto posible, entonces las rutas más cortas entre los sensores que capturan la información y la estación base serán siempre las que más se utilicen, por lo que la energía de los sensores que pertenecen a esas rutas se consumirá mucho más rápido y morirán antes que la del resto de los sensores de la RIS.

Tanto el tiempo de vida de la red como el retardo, están relacionados directamente al tamaño de las rutas establecidas por donde deben viajar los paquetes, desde los sensores que los generan hasta la estación base. Por ello el encaminamiento (algoritmo encargado de buscar y establecer las mejores rutas entre dos sensores) es una parte vital a considerar en el diseño de una RIS [5].

En las redes de computadoras tradicionales, es común que cada nodo de redistribución de la red (un enrutador, un *switch*, servidores, computadoras, etcétera) tenga una tabla de encaminamiento que indique las rutas más cortas entre dos nodos. En el caso de una RIS, sería imposible mantener una tabla de encaminamiento de este tipo en cada uno de los sensores, por cuestiones de las limitantes propias de estos dispositivos como su poca memoria y su limitado poder de cómputo, del gran número de sensores que puede llegar a tener una RIS (cientos o miles) y del cambio dinámico de la distribución de los sensores de la red.

Los algoritmos inspirados en algún fenómeno biológico han mostrado ser una opción competitiva para resolver algunos problemas difíciles de ingeniería y ciencias. Específicamente, la familia de los algoritmos de optimización por colonia de hormigas (*Ant Colony Optimization*, ACO), ha sido aplicada satisfactoriamente en la resolución del problema de encaminamiento, tanto en redes alambradas como inalámbricas [6].

ACO es una heurística que está inspirada en el comportamiento de las hormigas. Este algoritmo está compuesto por un conjunto de agentes llamados hormigas, que tienen la función principal de encontrar las mejores rutas entre un punto origen y un punto destino, con ayuda de la incorporación de una sustancia llamada feromona, que las hormigas segregan en el camino.

1.2. Definición del problema

Las RIS son utilizadas para el monitoreo de variables físicas del entorno de interés. Las características propias del tipo de sensores utilizados presentan limitantes directas, que impiden el uso de protocolos de redes de computadoras tradicionales, por lo que los protocolos para las RIS deben estar adaptados directamente a la aplicación que se esté diseñando.

Uno de los principales problemas que se tiene que resolver al momento de diseñar una RIS es el algoritmo de encaminamiento de datos, que afecta directamente su desempeño, tanto relacionado con el tiempo de vida de la red como con la latencia de la información.

Se debe encontrar un algoritmo de encaminamiento capaz de establecer cuál es la mejor ruta entre el sensor origen y la estación base, que tome en cuenta la dinámica en los cambios de distribución o ubicación y los niveles de energía de los sensores de la red y el retardo de la información.

1.3. Objetivos

A continuación se especifica tanto el objetivo general como los particulares de esta tesis.

1.3.1. General

- Proponer un algoritmo de encaminamiento, basado en la optimización por colonia de hormigas (ACO) para las redes inalámbricas de sensores (RIS), que maximice el tiempo de vida de la red y minimice la latencia de la transmisión de la información.

1.3.2. Particulares

- Definir indicadores de desempeño justos y realistas en cuestión de consumo de energía para la comparación de los algoritmos seleccionados.
- Establecer un escenario general de RIS bajo el cual realizar la simulación de los experimentos.
- Seleccionar e implementar dos algoritmos de encaminamiento de datos inspirados en ACO para RIS.
- Diseñar un algoritmo de encaminamiento que combine las principales características de los algoritmos estudiados.
- Comparar y evaluar el desempeño del algoritmo propuesto contra los algoritmos seleccionados.

1.4. Alcances y limitaciones

En este trabajo se realiza la programación y comparación de dos algoritmos ya existentes de encaminamiento de datos, así como a la creación de un algoritmo que combine las mejores características de ambos.

Puesto que las comparaciones que se desean realizar deben ser justas y lo más realistas posible en cuestión de consumo de energía, la simulación del desempeño de dichos algoritmos se realiza programándolos en lenguaje C y adaptándolos de tal forma que arrojen, al final de las corridas los valores de los indicadores que se establecen más adelante para su comparación.

1.5. Apoyos

La realización de esta tesis ha sido posible gracias al apoyo con sus becas de la Secretaría de Investigación y Posgrado (SIP-IPN) y del Consejo Nacional de Ciencia y Tecnología (CONACyT) a través de los proyectos 132073, 107688 y 60240.

1.6. Contribuciones

Los aportes principales de este trabajo son los siguientes:

- Una comparación justa y realista, desde el punto de vista del consumo de energía, entre dos de los algoritmos de encaminamiento inspirados en ACO para RIS más representativos del estado del arte.
- Un análisis de la relación entre los valores de los indicadores de desempeño obtenidos de los experimentos realizados con los algoritmos comparados.
- El diseño de un algoritmo de encaminamiento que tome las mejores características de los algoritmos mencionados y compita en desempeño con los mismos.

1.7. Publicaciones en congresos internacionales

- Christian Domínguez-Medina y Nareli Cruz-Cortés. ***Routing Algorithms for Wireless Sensor Networks using Ant Colony Optimization***, *9th Mexican International Conference on Artificial Intelligence (MICAI)*, noviembre 8-13, 2010, LNAI 6438. Pachuca, México.
- Christian Domínguez-Medina y Nareli Cruz-Cortés. ***Energy-Efficient and Location-Aware Ant Colony Based Routing Algorithms for Wireless Sensor Networks***, *Genetic and Evolutionary Computation Conference (GECCO)*, ACM, julio 12-16, 2011. Dublin, Irlanda. Aceptado.

1.8. Organización de la tesis

Este documento de tesis consta de seis capítulos.

- En el capítulo uno, la introducción explica brevemente el contenido de la tesis.
- En el capítulo dos se da una revisión al marco teórico de las RIS. Se establecen las diferencias principales entre las redes de computadoras clásicas y las RIS. Se define en detalle lo que es una RIS y las partes que lo componen. Se comentan las características del encaminamiento de datos para estas redes. Tambien la metaheurística ACO es presentada y se da la definición de un algoritmo de ACO.
- En el capítulo tres se explica el estado del arte respecto a los algoritmos de encaminamiento de datos para RIS basados en ACO. Se define el esquema general de un algoritmo de encaminamiento basado en ACO y se describen los ejemplos más representativos.
- En el capítulo cuatro se definen en detalle los algoritmos de encaminamiento estudiados y se lleva a cabo una comparación entre dichos algoritmos. Se crea y adapta un algoritmo donde se combinan las mejores características de los dos algoritmos y se especifican sus componentes.
- En el capítulo cinco se definen los experimentos y se muestran los resultados que se obtienen en la simulación de los algoritmos. Se realiza un análisis de los mismos y se explica el significado de los valores obtenidos de los indicadores propuestos.
- En el capítulo seis se detallan las conclusiones y el trabajo futuro que se desprenden del análisis que se realiza a los resultados obtenidos.

Capítulo 2

Marco teórico

2.1. Redes inalámbricas de sensores

En esta sección se muestran las diferencias principales entre las redes de computadoras clásicas y las RIS. Se define en detalle lo que es una RIS y las partes que lo componen.

2.1.1. Redes de computadoras

En las décadas de los 70's y 80's se produjo una sinergia entre los campos de las computadoras y las comunicaciones que desencadenó un cambio drástico en las tecnologías, productos y en las propias empresas que, desde entonces, se dedican conjuntamente a los sectores de las computadoras y las comunicaciones. De lo anterior se ha originado el campo de las redes de comunicación o redes de computadoras [7].

El objetivo principal de todo sistema de comunicaciones o red de computadoras, es intercambiar información entre dos o más entidades. Los elementos principales de un modelo de un sistema de transmisión de información (Figura 2.1) son: *La fuente* genera los datos a transmitir. *El transmisor* transforma y codifica la información para generar señales electromagnéticas susceptibles de ser transmitidas a través de algún sistema de transmisión. *El sistema de transmisión* puede ser desde una sencilla línea de transmisión hasta una compleja red que conecte a la fuente con el destino. *El receptor* acepta la señal proveniente del sistema de transmisión y la transforma de tal manera que pueda ser manejada por el dispositivo de destino.

Aunque el modelo presentado pueda parecer sencillo, en realidad implica una gran complejidad. Algunas de las tareas claves que se debe tener en cuenta en un sistema de comunicaciones son: La utilización del sistema de transmisión, la implementación de la interfaz, la generación de la señal, la sincronización de la comunicación, la gestión del intercambio de datos, la detección y corrección de errores, el control de flujo de datos, el direccionamiento y encaminamiento, La recuperación, el formato de mensajes, la seguridad y la gestión de la red.

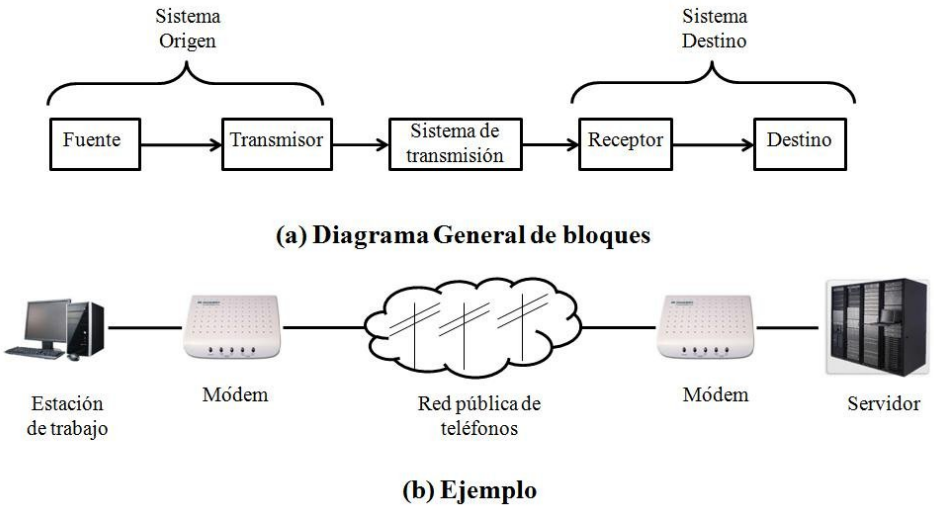

Figura 2.1: Modelo simplificado para las comunicaciones.

En el intercambio de datos entre computadoras, terminales y/u otros dispositivos de procesamiento, los procedimientos involucrados pueden llegar a ser bastante complejos. Considérese, por ejemplo, la transferencia de un archivo entre dos computadoras. En este caso debe haber un camino entre las dos computadoras, directo a través de una red de comunicaciones, pero además, el sistema fuente debe activar un camino directo de datos o bien debe proporcionar a la red de comunicaciones la identificación del sistema destino deseado y debe asegurarse de que el destino está preparado para recibir los datos. La aplicación de transferencia de archivos en el origen debe asegurarse de que el programa gestor en el destino está preparado para aceptar y almacenar el archivo para el usuario determinado y si los formatos de los dos archivos son incompatibles en ambos sistemas, uno de los dos deberá realizar una operación de traducción.

Se observa que debe haber un alto grado de cooperación entre las computadoras involucradas. En lugar de implementar toda la lógica para llevar a cabo la comunicación en un único módulo, el problema se divide en subtareas, cada una de las cuales se realiza por separado. En una arquitectura de protocolos, los distintos módulos se disponen en forma de pila vertical. Cada capa de la pila realiza el subconjunto de tareas relacionadas entre sí que son necesarias para comunicar con el otro sistema. Por lo general, las funciones más básicas se dejan a la capa inmediatamente inferior, olvidándose en la capa actual de los detalles de estas funciones. Además, cada capa proporciona un conjunto de servicios a la capa inmediatamente superior. Idealmente las capas deberían estar definidas de forma tal que los cambios en una capa no deberían necesitar cambios en las otras.

Para clasificar las redes tradicionalmente se consideran tres grandes categorías: Las redes de área amplia (*Wide Area Networks*, WAN), las redes de área local (*Local Area Networks*, LAN) y las redes de área metropolitana (*Metropolitan Area Network*, MAN). Las diferencias entre estas categorías son cada vez más difusas, tanto en términos tecnológicos como de posibles aplicaciones.

2.1.1.1. Redes inalámbricas

En los últimos años las LAN inalámbricas han ocupado un importante lugar en el mercado de las redes de área local. Cada vez más, las organizaciones se han dado cuenta de que las LAN inalámbricas son un complemento indispensable a las redes cableadas a fin de satisfacer las necesidades de movilidad, traslado y cobertura de lugares difíciles de cablear. Una red LAN inalámbrica es aquella que hace uso de un medio de transmisión inalámbrico. Hasta hace relativamente poco tiempo, las redes LAN inalámbricas eran poco usadas debido a su alto precio, la baja velocidad de transmisión, la existencia de problemas de seguridad y la necesidad de licencias. A medida que estos problemas se han ido solucionando, la popularidad de las LAN inalámbricas ha crecido rápidamente.

En la Figura 2.2 se muestra una configuración sencilla de una LAN inalámbrica típica en muchos entornos. Existe una LAN troncal cableada, como una Ethernet que conecta varios servidores, estaciones de trabajo y uno o más puentes o dispositivos de encaminamiento para la comunicación con otras redes. Adicionalmente, existe un módulo de control (CM, *control module*) que funciona como interfaz con la LAN inalámbrica. El módulo de control incluye funciones propias de un puente o de un dispositivo de encaminamiento para conectar la LAN inalámbrica con la troncal. Se deben destacar que algunos de los sistemas finales son dispositivos independientes, como estaciones de trabajo o servidores. Los concentradores (*hub*) u otros módulos de usuario (UM, *user module*) que controlan varias estaciones fuera de una LAN cableada pueden también formar parte de la LAN inalámbrica.

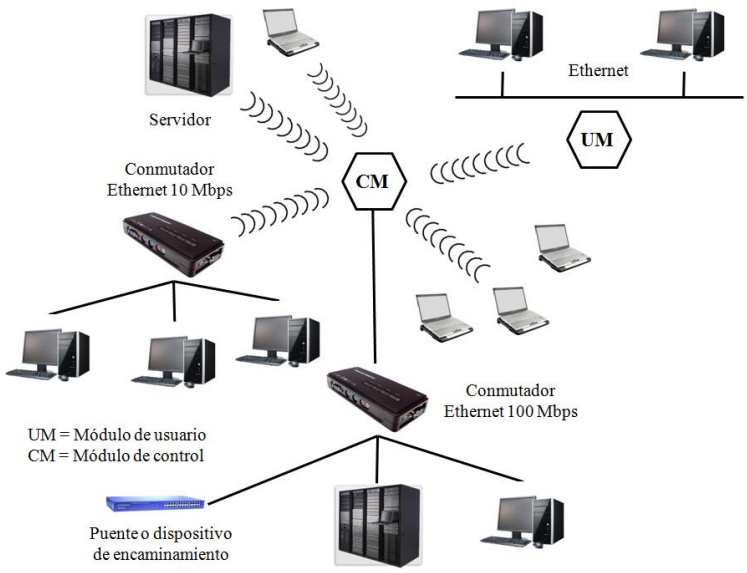

Figura 2.2: Ejemplo de configuración de una LAN inalámbrica de celda única.

2.1.1.2. Redes *ad-hoc* o MANET

Las redes *ad-hoc* tienen un origen militar. De forma casi contemporánea al nacimiento de Internet (o de su primer embrión, ARPAnet), a principios de los 70's, el ministerio de defensa americano se interesó por el nuevo proyecto de redes de transmisión de paquetes via radio (*packet radio networks*, PRNETs); su objetivo era posibilitar que las distintas unidades en un campo de batalla se pudieran comunicar entre sí, con libertad de movimiento y de forma cooperativa [8].

Una red *ad-hoc* también conocidas como redes móviles *ad-hoc* (*mobile ad-hoc network*, MANET) es una red entre dispositivos con las mismas capacidades (sin servidor central), establecida temporalmente para satisfacer alguna necesidad inmediata. Por ejemplo, un grupo de empleados, cada uno con su computadora, puede reunirse para una cita de negocios o para una conferencia, con conexión entre sus computadoras en una red temporal sólo durante la reunión.

Este tipo de redes inalámbricas son descentralizadas y son *ad-hoc* porque cada nodo está preparado para reenviar datos a los demás y la decisión sobre qué nodos reenvían los datos se toma de forma dinámica en función de la conectividad de la red. Esto contrasta con las redes tradicionales en las que los enrutadores llevan a cabo esa función. También difiere de las redes inalámbricas convencionales en las que un nodo especial, llamado punto de acceso, gestiona las comunicaciones con el resto de nodos.

En la Figura 2.3 se sugiere las diferencias entre una LAN inalámbrica *ad-hoc* y una LAN inalámbrica que proporciona ampliaciones de LAN y acceso nómada. En el segundo caso, la LAN inalámbrica presenta un infraestructura estacionaria consistente en una o más celdas con un módulo de control para cada una; dentro de cada celda pueden existir varios sistemas finales estacionarios. Las estaciones nómadas se pueden desplazar de una celda a otra. Por el contrario, en una red LAN *ad-hoc* no existe infraestructura; más aún, un conjunto de estaciones localizadas en el mismo dominio se pueden auto configurar dinámicamente para formar una red temporalmente.

2.1.2. Descripción de las redes inalámbricas de sensores (RIS)

Existe un tipo muy especial dentro del campo de las redes LAN inalámbricas *ad-hoc*, llamadas redes inalámbricas de sensores (*wireless sensor networks*). Una RIS está formada por un gran número de dispositivos embebidos (nodos sensores). Los nodos sensores son el elemento más común de una RIS. Son los encargados de recoger la información que se quiere medir. Habitualmente un nodo sensor está formado por los siguientes componentes: Un subsistema de comunicación inalámbrica de alcance limitado, que es el responsable del intercambio de mensajes con los nodos sensores vecinos, un subsistema de procesado (microcontrolador y memoria reducida) que lleva acabo el cómputo o procesamiento de la información recabada por el sensor, un subsistema del sensor que son uno o varios sensores que capturan información (mide un parámetro) del medio ambiente y una batería de energía eléctrica.

Los nodos de la RIS pueden ser distribuidos sobre un terreno firme, en el aire, bajo

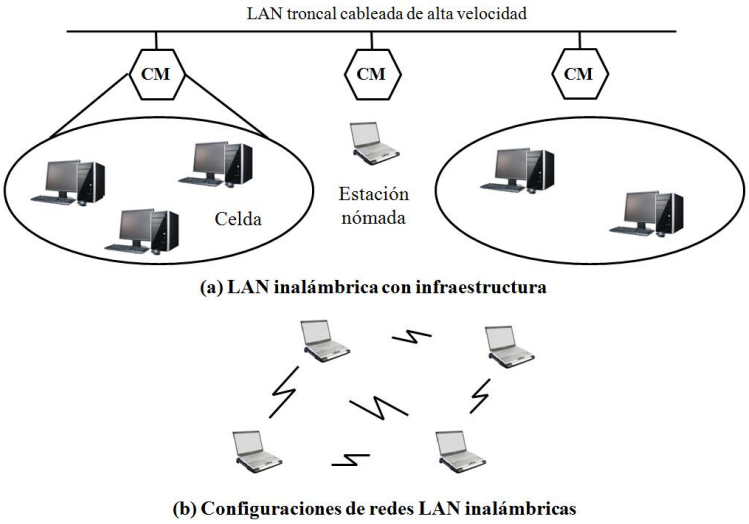

Figura 2.3: Configuraciones de redes LAN inalámbricas.

el agua, en el cuerpo humano, en vehículos, dentro de edificios, etcétera [9]. Estos dispositivos son responsables de recoger información del medio ambiente para controlar diversas condiciones en distintos puntos del área de interés. Entre las medidas que se pueden capturar por medio de estos sensores se tienen cantidades de temperatura, sonido, vibración, presión, movimiento, intensidad luminosa, humedad, presión, etcétera. Típicamente los nodos envían la información recogida a un dispositivo central (conocido como estación base, *gateway* o sumidero de la red) que actúa de interfaz, el cual traslada la información capturada por la red al usuario o sistema final (habitualmente está conectado a una red cableada y/o a Internet mediante conexión de banda ancha).

Las RIS son un caso particular de las redes inalámbricas *ad-hoc* multisalto (MANET) [10]. Las características de una RIS que hacen que se tome como una categoría distinta dentro de las redes inalámbricas son: La movilidad de los nodos, los cambios de rutas, el gran tamaño y densidad de la red, las limitantes de energía, capacidad de cómputo, memoria y de comunicación inalámbrica y la ausencia de infraestructura (por defecto en una RIS no existe ninguna entidad centralizada ni ninguna infraestructura. Cada nodo puede desempeñar tanto la función de terminal como de nodo de paso de ruta, debido a que no se realiza ningún tipo de distinción).

Un nodo sensor es un dispositivo con una serie de capacidades tanto de procesamiento de cómputo, memoria limitada, interfaz de comunicación inalámbrica y uno o varios sensores. Los componentes que son parte de un sensor se muestran en la Figura 2.4. El hardware básico de un nodo sensor se compone de un transceptor (transmisor/receptor), un microprocesador, uno o más sensores, memoria y batería. Estos componentes permiten trabajar al nodo sensor de manera adecuada, primeramente el nodo sensor captura la información del medio ambiente y la modifica internamente con ayuda de su procesador y memoria, para luego, mandar paquetes de la información capturada hacia la estación base mientras el nivel de energía de las baterías del nodo sensor lo permitan.

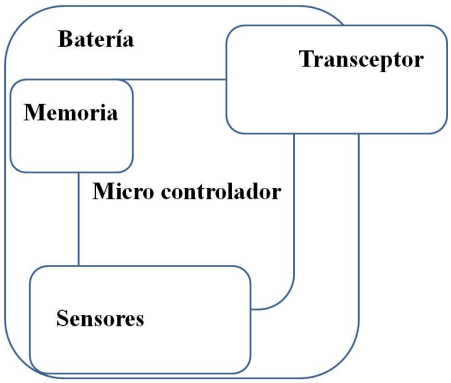

Figura 2.4: Componentes hardware de un nodo sensor.

Como se denota en [11], la tendencia es producir sensores a gran escala, a precios bajos, con mejor capacidad de cómputo y de tamaño reducido.

Si se quieren utilizar las RIS en aplicaciones como la captura de señales vitales en pacientes [12], el tamaño de los nodos sensores es importante. En general el tamaño del nodo sensor está limitado por el de su fuente de energía.

La alimentación de energía en los nodos sensores es proporcionada generalmente por una batería, o mediante un método de obtención de energía [13]. Un método para aumentar el tiempo de vida de las baterías de los nodos sensores es complementar la alimentación de las mismas con la energía del medio ambiente. Existen varias tecnologías para extraer energía del medio ambiente como la energía solar, térmica, energía cinética y la energía de vibración. Sin embargo, actualmente se carece de métodos económicamente factibles para explotar de manera eficiente estos recursos con un rendimiento óptimo. Puesto que uno de los principales puntos que se desea al momento de diseñar una RIS es que dure el mayor tiempo posible, es vital hacer un uso eficiente de la energía en los dispositivos [14]. Una importante característica de las RIS es reducir al mínimo el consumo de energía de los nodos, lo que proporciona al mismo tiempo el mayor rendimiento posible a los usuarios del sistema.

Existen varios fabricantes que trabajan en la tecnología de los nodos sensores [15], por ejemplo: *Crossbow* que desarrolla plataformas de hardware y software que dan soluciones para las RIS. Entre sus productos encontramos las plataformas Mica, Mica2, Micaz, Mica2dot, telos y telosb, la empresa *MVoteiv* que ha desarrollado la plataforma Tmote Sky y Tmote Invent y *Shockfish* que desarrolla TinyNode. En la Tabla 2.1 se observan las características de los sensores Micaz, Mica2, Mica2dot, Tmote y TinyNode.

2.1.2.1. Aplicaciones

En el pasado reciente, las RIS han encontrado su camino dentro de una gran variedad de aplicaciones y sistemas, con una vasta gama de requerimientos y características especiales. Como consecuencia, cada vez es más difícil hablar de los requisitos típicos

Tabla 2.1: Comparación de nodos.

	Micaz	**Mica2**	**Mica2dot**	**Tmote**	**TinyNode**
Distribuido por	Crossbow	Crossbow	Crossbow	Moteiv	Shockfish
Reloj	7.37 MHz	7.37 MHz	4 MHz	8MHz	8MHz
RAM	4 Kb	4 Kb	4 Kb	10 Kb	10 Kb
Batería	2 pilas AA	2 pilas AA	Pila de botón	2 pilas AA	Solar
Microcontrolador	Atmega 128L	Atmega 128L	Atmega 128L	MSP430	MSP430

respecto a los problemas de hardware y software de las RIS. Esto es especialmente problemático en un área de investigación multidisciplinaria, como las RIS, donde la estrecha colaboración entre los usuarios, los expertos del dominio de la aplicación, los diseñadores de hardware y desarrolladores de software, se necesitan mutuamente para poner en práctica sistemas eficaces en este ámbito.

En [16], los autores presentan una clasificación de algunas aplicaciones de muestra, que toman en cuenta el despliegue de los nodos, la movilidad, los recursos, costos, energía, heterogeneidad, modalidad, infraestructura, topología, cobertura, conectividad, tamaño, duración y calidad de servicio. Estas aplicaciones de ejemplo son:

Great Duck (observación de aves en la isla de Gran Pato, 2002), *ZebraNet* (2002), *Glacier* (monitoreo de un glaciar, 2004), *Herding* (cría de ganado, 2004), batimetría (2003), *Ocean* (vigilancia de océanos), *Grape* (seguimiento del crecimiento de las uvas, 2004), *Cold Chain* (gestión de la cadena de frío, 2004), *Avalanche* (rescate de víctimas de avalanchas, 2003), *Vital Sign* (monitoreo de signos vitales, 2004), *Power* (control de potencia, 2004), *Assembly* (montaje de piezas, 2002), *Tracking* (seguimiento de vehículos militares), *Mines* (sanación de campos de minas, 2003) y de *Sniper* (localización de francotiradores, 2004).

En [17] se muestran algunos ejemplos de aplicaciones generales para RIS para diferentes áreas:

- **Industrial:** Monitoreo y control de equipo industrial. Control de procesos de fabricación y automatización industrial. Vigilancia de fabricación.
- **Militar:** Conocimiento de la situación en el campo de combate. Detección de intrusos en las instalaciones, unidades de detección de movimientos en terreno enemigo, detección contra amenazas químicas y biológicas y control de logística en guerra urbana. Vigilancia del campo de batalla. Comando, control, comunicaciones, informática, inteligencia, vigilancia, reconocimiento, y los sistemas de ubicación. Asistencia militar o civil.
- **Localización de personas y lugares.**
- **Redes inalámbricas de tasa baja para localización precisa:** Rastreo de activos, personas, o cualquier cosa que pueda moverse en diferentes ambientes, incluidos los industriales, hospitalario, residencial y entornos de oficina.
- **Monitoreo:** Supervisar y controlar el mundo físico: El despliegue de redes distribuidas de sensores para una amplia gama de aplicaciones del control biológico

y ambiental, para ambientes marinos, terrestres y atmosféricos; observación de sistemas biológicos, ambientales y artificiales; monitoreo ambiental del agua y el suelo; etiquetado discreto de animales pequeños, de objetos en una fábrica o en entornos hospitalarios. Monitoreo de un hábitat (para determinar la población y comportamiento de animales y plantas). Mantenimiento de ciertas condiciones físicas (temperatura, luz, etc.).

- **Protección civil:** Detección y determinación de la ubicación de lugares de desastres. Detección de incendios, terremotos o inundaciones.
- **Automotriz:** Control de presión de neumáticos. Monitoreo de movilidad activa. Rastreo de vehículos. Control de tráfico.
- **Aeropuertos:** Etiquetas inteligentes. Etiquetas inalámbricas de equipaje.
- **Agricultura:** Sensores de niveles de humedad del suelo, de plaguicidas, de herbicidas, de pH.
- **Situaciones de emergencia:** Control de niveles de químicos peligrosos. Detectores de fuego. Monitoreo de áreas en desastre.
- **Maquinaria:** Monitoreo y control de maquinaria. Estudio de los movimientos de objetos o estructuras.
- **Terremotos o siniestros:** Sistemas de alerta.
- **Comercial:** Gestión de inventarios y control de la calidad del producto.
- **Salud:** Monitoreo de ubicaciones de pacientes y de condiciones de salud. Sensores de flujo sanguíneo, frecuencia respiratoria, electrocardiograma, presión arterial, etcétera. Observación de pacientes. Ayuda a pacientes con discapacidades. Mejora visual de un paciente con deficiencia óptica.

2.2. Algoritmos de encaminamiento para RIS

El encaminamiento de datos en una red determina el camino o ruta que atravesarán las unidades de datos (tramas, paquetes, mensajes, etcétera) desde el dispositivo origen hasta el destino.

La mayoría de los algoritmos de encaminamiento en redes interconectadas permiten elegir rutas que minimicen la latencia de la transmisión de datos. Los dispositivos de encaminamiento intercambian información para obtener una representación de los retardos a través del conjunto de redes. El encaminamiento de mínimo retardo ayuda a balancear la carga de la red, con el decremento de la congestión local y la reducción de los retardos experimentados por las conexiones.

El algoritmo de encaminamiento es una parte crítica en toda red. Los algoritmos de encaminamiento para redes inalámbricas deben tener en cuenta las características y limitantes propias de este tipo de redes [1].

El encaminamiento en las RIS es un problema complicado de resolver, debido a las características propias que distinguen a estas redes de otras redes inalámbricas como las

redes móviles *ad-hoc* o redes celulares. La dificultad del encaminamiento en RIS se debe en primer lugar, al número relativamente grande de nodos sensores en la red (de cientos a miles), donde no es posible construir un esquema de encaminamiento global para el despliegue de un gran número de nodos sensores, como tampoco sería recomendable o factible almacenar la cantidad de información referente a los identificadores de todos los nodos de la red en cada uno de los nodos. Por lo anterior, los protocolos de redes cableadas no pueden ser utilizados para las RIS.

Con el objetivo de que los nodos de las RIS utilicen una metodología cooperativa para establecer la comunicación, se debe fijar un algoritmo de encaminamiento que cumpla con las condiciones que se establezcan, de acuerdo con las necesidades propias de la aplicación.

Toda técnica de encaminamiento propuesta para trabajar en las RIS debe ser eficiente en cuanto al consumo de energía, ya que generalmente los sensores están alimentados por pilas y habitualmente se encuentran en lugares donde es difícil su cambio o reposición.

La rutas que defina el algoritmo de encaminamiento se deben establecer con base en un objetivo planeado, como por ejemplo: Maximizar el tiempo de vida de la red, asegurar que todo dato capturado del ambiente llegue a la estación base, minimizar la sobrecarga del tránsito de la red y del retardo de la información desde el nodo que la genera hasta la estación base, etcétera. Un buen camino es aquel que proporciona un costo mínimo de acuerdo con el objetivo planteado en alguna de las cuestiones mencionadas. No siempre el camino más corto es el de costo mínimo [18].

2.2.1. Clasificación de algoritmos de encaminamiento para RIS

Generalmente los algoritmos de encaminamiento para RIS se clasifican en dos grupos, tanto de acuerdo con la estructura de la red como al criterio de encaminamiento utilizado.

Como se observa en la Figura 2.5, los algoritmos de encaminamiento para este tipo de redes se pueden clasificar, primero por el tipo de estructura, en encaminamiento plano en el que todos los nodos desempeñan el mismo papel, encaminamiento jerárquico que tiene por objeto agrupar los nodos para que los nodos organizadores de la agrupación puedan hacer la agregación y reducción de datos con el fin de ahorrar energía, y encaminamiento basado en localización, los cuales utilizan la información de posición para transmitir los datos hasta las regiones deseadas de la red. Segundo, se clasifican por el criterio de encaminamiento: Basados en múltiples rutas, consultas, negociación, calidad de servicio (QoS) y basados en coherencia [19].

2.2.1.1. Algoritmos basados en la estructura de la red

La estructura de la red puede desempeñar un papel importante en el funcionamiento del algoritmo de encaminamiento para las RIS.

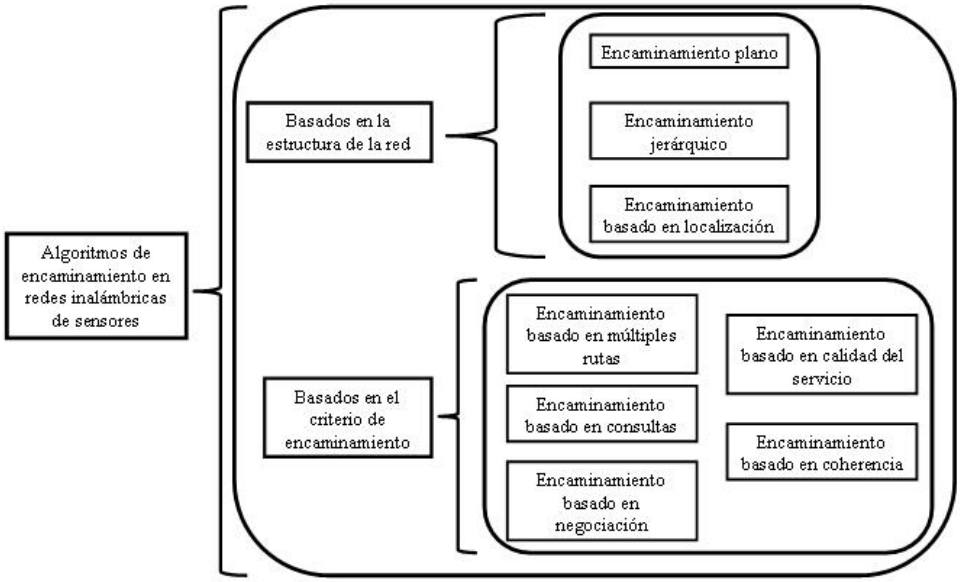

Figura 2.5: Algoritmos de encaminamiento en RIS.

- **Encaminamiento plano**: En las redes planas los nodos normalmente desempeñan el mismo trabajo y colaboran entre sí para llevar a cabo la tarea para la cual fue creada la red. Debido al gran número de nodos no es posible asignar un ID global para cada nodo. Esta consideración ha llevado al encaminamiento centrado en datos, donde la estación base envía consultas a determinadas regiones y espera los datos de los sensores ubicados en las regiones seleccionadas sin preocuparse por cuales fueron los nodos sensores que mandaron los datos capturados del medio ambiente.
- **Encaminamiento jerárquico**: Este tipo de encaminamiento fue propuesto originalmente para redes inalámbricas, son técnicas con ventajas especiales relacionadas con la escalabilidad y con la comunicación eficiente. Como tal, el concepto de encaminamiento jerárquico es utilizado también para realizar un encaminamiento eficiente en términos de energía en las RIS. En una arquitectura jerárquica, los nodos de mayor energía pueden ser utilizados para procesar y enviar la información, mientras que los nodos de baja energía pueden ser usados para realizar la detección y captura de la información del medio ambiente. La creación de grupos y la asignación de tareas especiales a los jefes de grupos, puede contribuir enormemente a la escalabilidad del sistema global, el tiempo de vida y la eficiencia de energía. El encaminamiento jerárquico es una forma eficaz para reducir el consumo de energía en un grupo y realizar la agregación de datos con el fin de disminuir el número de mensajes transmitidos a la estación base. Está compuesto principalmente de dos capas, donde una capa se utiliza para seleccionar el líder del grupo y la otra para el encaminamiento. Sin embargo, la mayoría de las técnicas en esta categoría no tienen que ver con rutas, sino con quién y cuándo enviar, procesar o agregar la información y distribución de canal.
- **Encaminamiento basado en localización**: En este tipo de encaminamiento

los nodos sensores son identificados por medio de sus ubicaciones. La distancia entre nodos vecinos puede estimarse con base en los niveles de la señal de entrada. Las coordenadas relativas de los nodos vecinos se pueden obtener mediante el intercambio de información entre los vecinos [20, 21, 22]. Alternativamente, la ubicación de los nodos puede estar disponible directamente mediante la comunicación con un satélite con el uso de un dispositivo GPS. Para ahorrar energía, algunos sistemas basados en ubicación mantienen los nodos dormidos si no hay actividad [23, 24].

2.2.1.2. Algoritmos basados en el criterio de encaminamiento

En esta sección se revisan los algoritmos de encaminamiento con diferentes funcionalidades. Cabe señalar que algunos de estos algoritmos pueden encontrarse dentro de una o varias de las categorías de encaminamiento mencionadas anteriormente.

- **Algoritmos de encaminamiento multicamino**: Estos algoritmos utilizan varias rutas en lugar de una sóla ruta con el fin de mejorar el rendimiento de la red. La tolerancia a fallos de un algoritmo de encaminamiento se mide por la probabilidad de que exista un camino alternativo entre una fuente y un destino, cuando la principal ruta falla.
- **Algoritmos de encaminamiento basado en petición**: En este tipo de encaminamiento los nodos destinos propagan una consulta de datos (tarea de detección) a un nodo a través de la red, y el nodo con estos datos los envía al nodo que inició la consulta. La difusión dirigida descrita anteriormente es un ejemplo de este tipo de encaminamiento.
- **Algoritmos de encaminamiento basado en negociación**: Estos algoritmos utilizan descriptores de datos con el fin de eliminar mediante una negociación, las transmisiones de datos redundantes. Las decisiones de comunicación también se basan en los recursos disponibles para ellas.
- **Encaminamiento basado en calidad de servicio (QoS)**: Al usar estos algoritmos la red tiene un equilibrio entre el consumo de energía y la calidad de los datos. En particular la red debe satisfacer determinadas métricas de QoS (retardo, energía, ancho de banda, etcétera) al entregar los datos a la estación base.
- **Encaminamiento basado en procesamiento de datos coherentes y no coherentes**: El procesamiento de datos es un componente importante en el funcionamiento de las RIS. Los algoritmos de encaminamiento emplean diferentes técnicas de procesamiento de datos. En general los nodos sensores cooperan entre sí en el procesamiento de datos. Dos ejemplos de técnicas de procesamiento de datos son el encaminamiento basado en procesamiento de datos coherente y no coherente. En el encaminamiento basado en procesamiento de datos no coherentes, los nodos procesan localmente los datos antes de que sean enviados a otros nodos para su posterior procesamiento. Los nodos que realizan el procesado se denominan “agregadores”. En el encaminamiento coherente, los datos se enviarán

a los “agregadores” después de un procesamiento mínimo. El procesamiento mínimo típicamente incluye tareas como creación de marcas de tiempo y supresión de duplicados.

De las clasificaciones mencionadas anteriormente existen algoritmos híbridos que mezclan más de una categoría. En la Tabla 2.2 se muestran las categorías donde caen los algoritmos ya mencionados y también se comparan diferentes técnicas de encaminamiento acorde a los indicadores ya mencionados.

Tabla 2.2: Clasificación y comparación de algoritmos de encaminamiento para RIS [1].

	Clasificación	**Movilidad**	**Basado en posición**	**Uso de energía**	**Basado en negociaciones**	**Agregación de datos**	**Localización**	**QoS**	**Complejidad**	**Escalabilidad**	**Multiruta**	**Basado en consultas**
SPIN [25, 26]	Plano	Posible	No	Limitado	Si	Si	No	No	Baja	Limitada	Si	Si
Difusión Directa [27]	Plano	Limitada	No	Limitado	Si	Si	Si	No	Baja	Limitada	Si	Si
Por Rumor [28]	Plano	Muy limitada	No	No	No	Si	No	No	Baja	Buena	No	Si
MCFA [29]	Plano	No	No	No	No	No	No	No	Baja	Buena	No	No
CADR [30]	Plano	No	No	Limitado	No	Si	No	No	Baja	Buena	No	No
EAR [31]	Plano	Limitada	No	No	No	No	Si	No	Baja	Limitada	No	Si
LEACH [32]	Jerárquico	Limitada	No	Máximo	No	Si	Si	No	Baja	Buena	No	No
APTEEN [33, 34]	Jerárquico	Limitada	No	Máximo	No	Si	Si	No	Baja	Buena	No	No
PEGASIS [35]	Jerárquico	Limitada	No	Máximo	No	No	Si	No	Baja	Buena	No	No
SMECN [36]	Jerárquico	No	No	Máximo	No	No	No	No	Baja	Baja	No	No
SOP [37]	Jerárquico	No	No	No	No	No	No	No	Baja	Baja	No	No
HPAR [38]	Jerárquico	No	No	No	No	No	No	No	Baja	Buena	No	No
GAF [24]	Ubicación	Limitada	No	Limitado	No	No	No	No	Baja	Buena	No	No
GEAR [39]	Ubicación	Limitada	No	Limitado	No	No	No	No	Baja	Limitada	No	No
SPAN [23]	Ubicación	Limitada	No	No	Si	No	No	No	Baja	Limitada	No	No
GOAFR [40]	Ubicación	No	No	No	No	No	No	No	Baja	Buena	No	No

2.3. Optimización por colonia de hormigas (ACO)

En esta sección se comentan las características de la metaheurística ACO y se da la definición de un algoritmo basado en ACO.

2.3.1. Antecedentes

En el lenguaje coloquial optimizar significa poco más que mejorar; sin embargo, en el contexto científico la optimización es el proceso de tratar de encontrar la mejor solución posible para un determinado problema. En un problema de optimización existen diferentes soluciones y un criterio para discriminar entre ellas. De forma más precisa, estos problemas se pueden expresar como el procedimiento de encontrar el valor de

unas variables de decisión para los que una determinada función objetivo alcanza su valor máximo o mínimo. El valor de las variables en ocasiones está sujeto a algunas restricciones.

La idea intuitiva de un problema de optimización "difícil de resolver" puede ser: Un problema de optimización para el que no podemos garantizar el encontrar la mejor solución posible en un tiempo razonable. La existencia de una gran cantidad y variedad de problemas difíciles, que aparecen en la práctica y que necesitan ser resueltos, impulsó el desarrollo de procedimientos eficientes para encontrar buenas soluciones aunque no fueran óptimas. Estos métodos, en los que la rapidez del proceso es tan importante como la calidad de la solución obtenida, se denominan métodos heurísticos aproximados.

Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, donde la estructura del problema se utiliza de forma inteligente para obtener una buena solución.

En contraposición a los métodos exactos que proporcionan una solución óptima del problema, los métodos heurísticos se limitan a proporcionar una buena solución no necesariamente óptima. El tiempo invertido por un método exacto para encontrar la solución óptima de un problema difícil, si es que existe tal método, es de un orden de magnitud muy superior al del heurístico, capaz de llegar a ser tan grande, que sea inaplicable.

Existen muchos métodos heurísticos de naturaleza muy diferente, por lo que es complicado dar una clasificación completa. Además, muchos de ellos han sido diseñados para un problema específico sin posibilidad de generalización o aplicación a otros problemas similares. En [41] se proponen unas categorías amplias, no excluyentes, en donde ubicar a las técnicas heurísticas más conocidas:

Métodos de descomposición. El problema original se descompone en subproblemas más sencillos de resolver.

Métodos inductivos. La idea de estos métodos es generalizar versiones pequeñas o más sencillas al caso completo.

Métodos de reducción. Consisten en identificar propiedades que se cumplen por las buenas soluciones que se introducen como restricciones del problema. El objeto es restringir el espacio de soluciones y simplificar el problema.

Métodos constructivos. Consisten en construir paso a paso una solución del problema. Usualmente son métodos deterministas y suelen estar basados en la mejor elección en cada iteración.

Métodos de búsqueda local. A diferencia de los métodos anteriores, los procedimientos de búsqueda o mejora local comienzan con una solución del problema y la mejoran progresivamente.

Si bien todos estos métodos han contribuido a ampliar nuestro conocimiento para la resolución de problemas reales, los métodos constructivos y los de búsqueda local constituyen la base de los procedimientos metaheurísticos.

En estos momentos existe un gran desarrollo y crecimiento de estos métodos. Algunos ejemplos de técnicas ya consolidadas y que han probado su eficacia sobre una colección significativa de problemas son: La búsqueda tabú, el recocido simulado, métodos evolutivos, redes neuronales, optimización por colonia de hormigas, etcétera [42] [43] [44].

Uno de los algoritmos heurísticos que han sido seleccionados para solucionar el problema del encaminamiento en RIS, es el algoritmo de optimización por colonia de hormigas.

La metaheurística de optimización por colonia de hormigas es una técnica que permite encontrar soluciones factibles a problemas de optimización basados el uso de agentes móviles llamados hormigas, los cuales se inspiran en el comportamiento real de las colonias de hormigas.

2.3.2. Comportamiento de las hormigas reales

Las hormigas son insectos sociales que viven en colonias y que debido a su colaboración mutua, son capaces de mostrar comportamientos complejos y realizar tareas difíciles.

Un aspecto interesante del comportamiento de muchas especies de hormigas es su habilidad para encontrar los caminos más cortos entre su hormiguero y las fuentes de alimento. Este hecho es especialmente interesante si se tiene en cuenta que muchas de las especies de hormigas son casi ciegas, lo que evita el uso de pistas visuales.

Durante el proceso en que la colonia de hormigas se desenvuelve y explota la fuente de alimento, las hormigas establecen caminos desde el hormiguero hasta la comida, con un número de hormigas especialistas en la detección de dichos caminos esparcidas en los lugares entre el hormiguero y la fuente de alimento.

Mientras se mueven entre el hormiguero y la fuente de alimento, algunas especies de hormigas, como por ejemplo las hormigas argentinas [45], depositan una sustancia química denominada feromona (una sustancia que puede "olerse").

Si no se encuentra ningún rastro de feromona, las hormigas se mueven de manera aleatoria, pero cuando existe feromona depositada en el camino, tienen tendencia a seguir el rastro.

En la práctica, la elección entre distintos caminos toma lugar cuando varios caminos se cruzan. Las hormigas eligen el camino a seguir basándose en la cantidad de feromona del mismo: Cuanto más fuerte es el rastro de feromona, mayor es la probabilidad de elegirlo. Puesto que las hormigas depositan feromona en el camino que siguen, este comportamiento lleva a un proceso de retroalimentación que concluye con la formación de rastros señalados por una concentración de feromona elevada [46].

Este comportamiento permite a la colonia de hormigas encontrar los caminos más cortos entre su hormiguero y la fuente de alimento. Las hormigas sólo se comunican de manera indirecta, a través de modificaciones del espacio físico que perciben. Esta forma de comunicación se denomina estimergia [47]. Esta idea se ilustra en la Figura 2.6 [48].

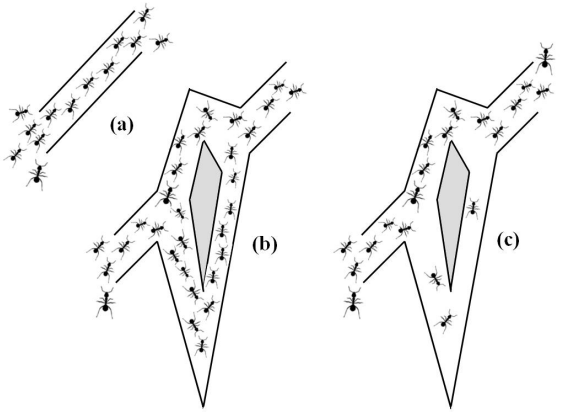

Figura 2.6: Comportamiento de las hormigas reales.

Debido a la gran persistencia de feromona es difícil que las hormigas "olviden" un camino que tiene un alto nivel de feromona aunque hayan encontrado un camino aún más corto [49].

2.3.3. Definición del algoritmo de ACO

La metaheurística de optimización por colonia de hormigas permite que sus propios miembros (hormigas) tengan una memoria basada en su experiencia y en su comportamiento (debida al depósito de feromona), lo que les permite en conjunto tener un comportamiento dinámico que les facilite encontrar rutas más cortas rápidamente [50].

Inspirado en el comportamiento anteriormente explicado y su semejanza con el problema del agente viajero (*Travelling salesman problem*, TSP), el primer algoritmo ACO propuesto fue el sistema de hormigas (*Ant system*) para resolver pequeñas instancias del TSP [51, 52]. El sistema de hormigas fue propuesto por Marco Dorigo y es el primer ejemplo de un algoritmo ACO en la literatura [6]. Aunque los primeros resultados fueron prometedores, el sistema de hormigas no resultó competitivo con el estado del arte en algoritmos de optimización combinatoria diseñados para el TSP. Con la motivación de obtener algoritmos competitivos, ACO ha recibido una creciente atención de la comunidad científica.

Con el objetivo de proveer un marco de referencia para el diseño de nuevas versiones de algoritmos, la metaheurística ACO fue propuesta [53]. Aun cuando existen varias versiones de ACO, se presenta a continuación lo que se considera un enfoque estándar [54].

En cada iteración del algorimo, cada hormiga k construye un recorrido completo M_k, creándose por lo tanto un conjunto de recorridos $M = \{M_k\}$. Para la construcción de cada recorrido se realiza el procedimiento siguiente. Inicialmente es elegida una ciudad al azar, luego es seleccionada una ciudad iterativamente hasta completar un recorrido válido. Mientras es visitada una ciudad i, N_i representa el conjunto de ciudades todavía no visitadas; la probabilidad de elegir una ciudad j desde la ciudad i es definida por:

$$P_{ij}^{k} = \left\{ \begin{array}{ll} \frac{\psi_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{\forall g \in N_i} \psi_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} & \text{si } j \in N_i \\ \\ 0 & \text{en otro caso} \end{array} \right\} \tag{2.1}$$

ACO usa una matriz de feromonas $\psi = \{\psi_{ij}\}$ para almacenar el nivel de feromonas de cada arista (i, j) del TSP. Esta matriz de feromonas es inicializada con los valores $\psi_{ij} = \psi_{init} \forall (i, j)$, donde $\psi_{init} > 0$. También explota información de costos de distancia con el uso de $\eta_{ij} = \frac{1}{d(i,j)}$, donde $d(i, j)$ es la distancia entre la ciudad i y j. Los parámetros α y β definen la influencia relativa entre los rastros de feromonas y la información de costo.

Después de que todas las hormigas han construido sus rutas, se aplica evaporación de feromonas para toda arista (i, j) de acuerdo con $\psi_{ij} = (1 - \rho) \cdot \psi_{ij}$, donde el parámetro $\rho \in (0, 1]$ determina la tasa de evaporación. Finalmente cada hormiga k deposita una cantidad de feromona en las aristas del recorrido construido por ella, de longitud $l(M_k)$, de acuerdo con $\psi_{ij} = \psi_{ij} + \frac{1}{l(M_k)}$. Esto significa que mientras más corto es el recorrido de una hormiga, más cantidad de feromonas recibirán las aristas de ese recorrido. En general, las aristas que son utilizadas por muchas hormigas y las aristas de recorridos de longitud pequeña recibirán más feromona y por lo tanto serán más frecuentemente elegidas en las iteraciones futuras del algoritmo.

En el Algoritmo 1 se muestra el pseudocódigo del enfoque estándar de ACO:

Algoritmo 1 Enfoque estándar de ACO.

1: $u = 0$ // Inicialización del contador de iteraciones
2: $\psi =$ Inicializar la matriz de feromona // $\psi_{ij} = \psi_{init} \forall (i, j)$
3: **mientras** condición de fin no se alcance **hacer**
4: $M =$ Construir soluciones (ψ)
5: $\psi =$ Actualizar la matriz de feromonas (M) // Actualización de ψ con el uso de M
6: $u = u + 1$ // Incremento del contador de iteraciones
7: **fin mientras**

Algunas investigaciones sobre ACO han demostrado que puede obtenerse un mejor rendimiento por medio de la explotación de las mejores soluciones encontradas durante la ejecución, aún cuando este procedimiento agrava el problema de la convergencia prematura de la búsqueda. La clave para lograr un mejor rendimiento de los algoritmos ACO es combinar la explotación de las mejores soluciones encontradas con un mecanismo para evitar la convergencia temprana. El sistema de hormigas MAX-MIN (*MAX-MIN ant system*, MMAS), diseñado por Stutzlen y Hoos [55], ha sido especialmente desarrollado para cumplir con estos requerimientos.

En la búsqueda de herramientas de análisis de ACO se pensó en desarrollar un algoritmo muy simple que intentara mantener ciertas características de ACO y más específicamente de MMAS. De esta manera fue concebido el Ómicron ACO (OA). Su nombre proviene del principal parámetro de entrada utilizado, Ómicron (O). OA evolucionó hasta convertirse en un algoritmo, conceptualmente simple y competitivo [56].

Esta simplicidad conceptual, sumada al estudio del espacio de soluciones del TSP, permitió utilizarlo para analizar las razones del éxito de ACO en el TSP [54].

En los últimos 10 años algunas aplicaciones basadas en ACO, han mostrado la efectividad de la optimización por colonia de hormigas. Sin embargo, existe muy poca teoría que explique las razones por las cuales esta técnica logra resultados factibles. Birattari propuso un tipo de interpretación de ACO dentro del ámbito de control y aprendizaje reforzado. Meuleau y Dorigo mostraron que los algoritmos basados en ACO y los métodos de optimización de descenso de gradiente estocástico están fuertemente relacionados y, además que un conjunto particular de algoritmos basados en ACO converge con una probabilidad de 1 a un óptimo local [2].

ACO ha sido utilizada para resolver el problema del encaminamiento de datos en RIS; algunos ejemplos de estos algoritmos de encaminamiento serán descritos en el siguiente capítulo.

Capítulo 3

Estado del arte

3.1. Esquema general de un algoritmo de encaminamiento basado en ACO para RIS

La principal característica de un algoritmo de encaminamiento basado en ACO para RIS consiste en la obtención continua de información de encaminamiento de la red. Dicha información se obtiene a través de la prueba de rutas con la ayuda de paquetes de información llamados *hormigas*. Las hormigas son colocadas originalmente en el nodo origen con la tarea de encontrar rutas entre los demás nodos hasta el nodo destino llamado base. Una hormiga camina desde el nodo origen hasta el nodo destino. El inicio del algoritmo de encaminamiento se puede observar en la Figura 3.1.

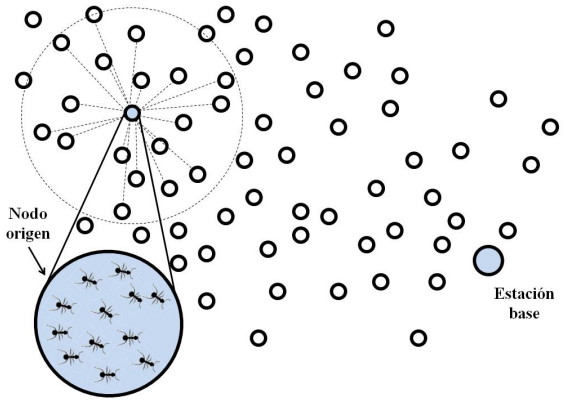

Figura 3.1: Inicio del algoritmo de encaminamiento basado en ACO.

Cada una de las hormigas recolecta información referente a la calidad de la ruta encontrada, para posteriormente utilizarla para actualizar los niveles de feromona de los nodos intermedios de la ruta, con lo que se incrementan los niveles de feromona de las mejores rutas de la RIS y crea una forma de aprendizaje reforzado distribuido en la red [57].

La dinámica de los niveles de feromona en la red se ilustra en la Figura 3.2. En a) se observa la distribución de los nodos, con niveles iniciales de feromona nulos entre los nodos de la red. El conjunto de hormigas viajan a través de los nodos, desde el nodo origen s_o hasta el nodo destino s_b y encuentran rutas entre dichos nodos. En b) y c) cada hormiga al llegar al nodo destino s_b, regresa sobre sus pasos hasta el nodo origen s_o y deposita una cantidad de feromona sobre la ruta. Después de varias iteraciones se observa la convergencia del sistema hacia la mejor ruta, la que puede observarse en d).

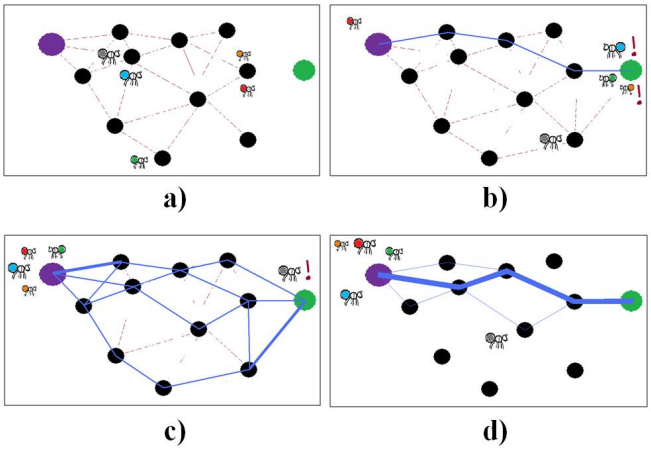

Figura 3.2: Dinámica de los niveles de feromona de la RIS.

3.2. Algoritmos de encaminamiento basados en ACO para RIS

En esta sección se describen algunos ejemplos de algoritmos de encaminamiento basados en ACO.

3.2.1. Algoritmo de encaminamiento para el cuidado de la calidad en el servicio (QoS) basado en ACO

En [58] se propone un algoritmo de encaminamiento de datos llamado ACO-QoSR (por sus siglas en inglés *ACO based QoS routing algorithm*) que utiliza un enfoque del algoritmo ACO modificado para resolver los problemas de encaminamiento, que toma en cuenta, tanto el retardo de la información, las limitantes de energía y el ancho de banda de la comunicación inalámbrica. ACO-QoSR busca las mejores rutas que satisfagan los requerimientos mínimos de la QoS y el equilibrio entre el cuidado de las restricciones de QoS y su complejidad computacional.

El encaminamiento basado en la QoS requiere una ruta que satisfaga los requerimientos de punto a punto utilizados en las redes *ad-hoc*, frecuentemente dados en términos de

ancho de banda, retardo de la información y pérdida de datos. Las aplicaciones que corren dentro de las RIS son normalmente aplicaciones sensibles al retardo, por lo que los indicadores de desempeño de las RIS principalmente se refieren al retardo de la información y a un factor de balance de energía.

Cuando un nodo origen ha capturado información del medio, ésta debe ser enviada a la estación base. Primero, el nodo revisa su tabla de encaminamiento para encontrar la ruta apropiada. La fase de prueba de una nueva ruta se realiza solo si no existen rutas que aún no han expirado desde el nodo actual hasta la estación base, mientras que el nodo requiere mantener la información guardada en memoria hasta que se pueda transmitir. Se requieren m hormigas para hacer una prueba de encaminamiento. Después de haber realizado el proceso de descubrimiento de la ruta nueva, el dato que se guardaba en memoria se manda inmediatamente a la estación base. Para reducir el retardo en la fase de búsqueda de la nueva ruta, se requiere que el algoritmo ACO-QoSR empiece el proceso de descubrimiento de rutas al mismo tiempo en que se inicializa la red.

Lo que se debe realizar al momento en el que un nodo sensor recibe a una hormiga, se describe en la Figura 3.3. Hay tres fases en el algoritmo ACO-QoSR: La fase de las hormigas que van, cuando las hormigas vienen de regreso y la fase de mantenimiento de las rutas.

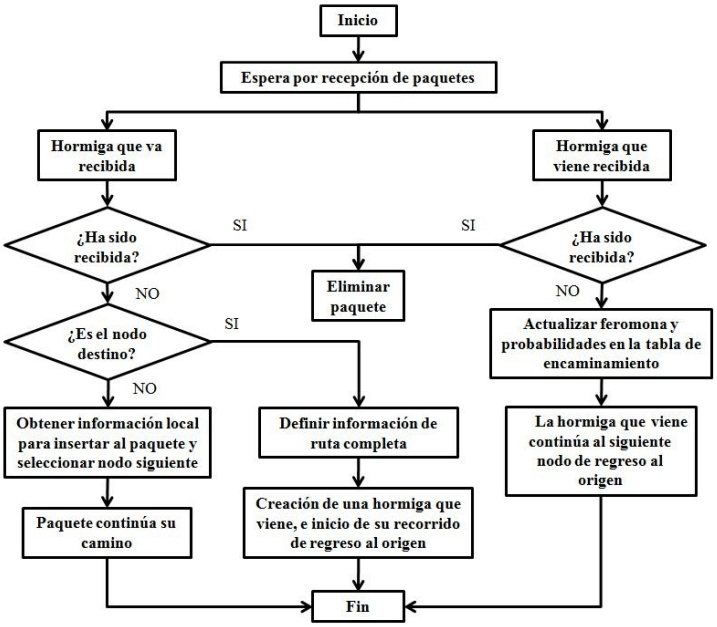

Figura 3.3: ACO-QoSR - Proceso de descubrimiento de rutas.

3.2.2. Algoritmo de difusión directa con cuidado en la energía basado en feromona

El algoritmo de difusión directa consciente de la energía basado en feromona (*Pheromone based energy aware directed diffusion algorithm*, PEADD) se basa en el algoritmo

de difusión directa (DD), donde el nodo destino propaga intereses (tareas de captura de datos del medio ambiente), con base en un método de inundación y con la ayuda de los gradientes establecidos. El gradiente se basa en la frecuencia de solicitud de información del interés. Un gradiente comúnmente se deriva de la frecuencia a la cual el nodo destino solicita información de un interés dado e, indica la frecuencia de actualización deseada y el vecino al cual se debe enviar la información. El nodo destino periódicamente transmite un mensaje que indica el interés en turno a cada uno de los nodos vecinos. En el paso de la propagación de datos el nodo selecciona el gradiente solamente basado en la velocidad en la que recibe los intereses. De manera natural, se espera que la feromona de las hormigas pueda ser utilizada para reflejar la energía restante en la búsqueda de nuevas rutas, por ejemplo, todo nodo tiene un nivel de feromona variable que se reduce con base en la frecuencia de transmisión en una sola ruta. En el algoritmo DD cada nodo selecciona el siguiente nodo a saltar de forma probabilística con ayuda del nivel de feromona. La Figura 3.4 muestra la idea básica de los niveles de feromona de la RIS. El primer paso es para propagar los intereses desde el nodo destino. El segundo paso es para inicializar la información de los gradientes en los nodos intermedios y para inicializar también los niveles de feromona. El tercer paso es para seleccionar la ruta apropiada con base en el nivel de feromona de la ruta.

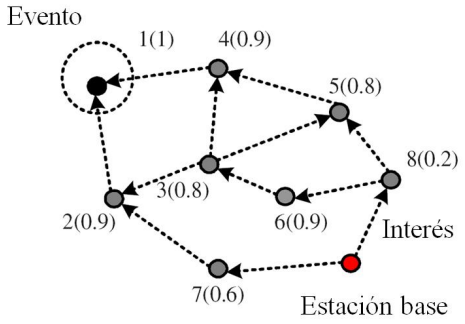

Figura 3.4: Difusión directa - Idea básica.

No es una buena idea para la feromona de los nodos que sean inicializadas con respecto a la energía de los nodos adyacentes, los cuales pueden ser detectados por la propagación del interés. Es posible que un nodo con un nivel de energía bajo quede dentro de la ruta establecida. Este caso se muestra en la Figura 3.4. El nodo 2 tiene dos nodos para seleccionar el siguiente nodo a saltar: el nodo 3 y el nodo 7. El nivel de feromona para el nodo 3 es 0.8 y para el nodo 7 es 0.6 (el nivel de energía original es 1). Luego entonces el nodo 2 selecciona el nodo 3 como el siguiente nodo a saltar. Sin embargo cuando la información es transmitida al nodo 3 el nodo 8 se encuentra en la ruta de transmisión y la feromona del nodo 8 es 0.2. Por lo tanto es muy probable que el nodo 8 muera rápidamente.

Una buena idea es establecer el nivel de energía restante en la ruta seleccionada por la propagación del interés, como el nivel de feromona mínimo encontrado en dicha ruta, por ejemplo en el nodo 2 la feromona para el nodo adyacente 3 debería ser 0.2, puesto que el mínimo nivel de energía encontrado en la ruta del interés es el nivel del nodo 8. De la misma forma la feromona del nodo 7 debería ser 0.6.

En el algoritmo PEADD la cantidad de feromona se reduce por la transmisión de información, porque la feromona está ligada a la energía restante del nodo. El nivel de feromona se actualiza con base en la cantidad de la información transmitida.

3.2.3. Algoritmo de encaminamiento adaptativo mejorado

Los autores Reza y Abdur de [59] proponen el algoritmo de encaminamiento adaptativo mejorado (*Improved adaptive routing algorithm*, IAR) basado en ACO para arquitecturas planas basadas en la localización de los nodos sensores. Esta propuesta intenta maximizar el tiempo de vida de la RIS y controlar, reaccionar y adaptarse a los cambios de topología de la red.

IAR utiliza una distribución de probabilidad en cada nodo sensor de la RIS, donde una tabla de encaminamiento se mantiene guardada como se muestra en la Tabla 3.1. Cada tabla de encaminamiento contiene $|N_k| \cdot N$ entradas, donde $|N_k|$ corresponde al conjunto de vecinos del nodo k y N es el número de nodos sensores en la RIS. Las hormigas utilizan las tablas de encaminamiento para escoger el siguiente nodo a saltar.

Tabla 3.1: IAR - Estrutura de la tabla de encaminamiento para el nodo k.

	Destinos			
Nodos vecinos	1	2	...	N
1	$P_{1,1}$	$P_{1,2}$	...	$P_{1,N}$
2	$P_{2,1}$	$P_{2,2}$	...	$P_{2,N}$
...	...	...	...	...
$\|N_k\|$	$P_{\|N_k\|,1}$	$P_{\|N_k\|,2}$	...	$P_{\|N_k\|,N}$

Cada fila en la tabla corresponde a un vecino y cada columna a un destino. Se define una probabilidad P_{ij} de un nodo k como la probabilidad de escoger el nodo i como el vecino a saltar desde el nodo k para ir hasta el nodo destino j. Basado en esta distribución de probabilidad las hormigas exploran nuevas y mejores rutas. Una vez que las nuevas rutas han sido encontradas las probabilidades para escoger el siguiente nodo a saltar son actualizadas en las tablas de encaminamiento, para que el siguiente conjunto de hormigas puedan seguir estas rutas. La siguiente restricción de probabilidad debe ser satisfecha por todas las entradas de las tablas de encaminamiento [59]:

$$\sum P_{i,d} = 1, \forall i \in vecinos(k), d \in [1, N]. \tag{3.1}$$

Se inicializan todas las tablas de encaminamiento con probabilidades iguales. En esta propuesta se utilizan dos tipos de hormigas, las hormigas que van al nodo destino (d) (F_{ant}) y las hormigas que van de regreso al nodo origen s (B_{ant}).

Desde cada nodo sensor k que tiene $|N_k|$ vecinos, la F_{ant} selecciona el siguiente nodo a saltar i con una probabilidad de $P_{i,d}$, la cual se define como sigue [59]:

$$P'_{i,d} = \frac{P_{i,d} + \beta \cdot c_i}{1 + \beta \cdot (|N| - 1)} \tag{3.2}$$

Donde $P'_{i,d}$ es la suma de la entrada probabilística $P_{i,d}$ de la tabla de encaminamiento, con un factor de corrección c_i y el coeficiente β. c_i es el costo desde el nodo actual k hasta el nodo vecino i, el cual se calcula de la siguiente forma [59]:

$$c_i = 1 - \frac{D_{k,i}}{\sum_{j=1}^{|N_k|} D_{k,j}} \tag{3.3}$$

Donde $D_{k,i}$ es la distancia entre el nodo actual k y su vecino el nodo i.

Como se muestra en la ecuación 3.3, el factor de corrección c_i es proporcional a la distancia $D_{k,i}$. El valor de β en la ecuación 3.2 da un peso al nivel del factor de corrección c_i con respecto al valor de probabilidad ($P_{i,d}$) almacenado en la tabla de encaminamiento. El coeficiente β tiene un valor entre cero y uno.

3.2.4. Algoritmo de retardo y energía

En [60] el algoritmo de retardo de energía (*Energy Delay Algorithm*, E&D) tiene como principal objetivo encontrar la mejor ruta entre el nodo sensor origen y la estación base, maximizar el tiempo de vida de la red y también proporcionar servicios de transmisión de datos en tiempos lo más cortos posibles.

La clave principal del algoritmo E&D es tomar ventaja de la conjunción de la energía y el retardo en la RIS, con el objetivo de actualizar la feromona de los nodos.

En esta propuesta se considera que la red inalámbrica está modelada como un grafo dirigido $G(N, A)$, donde N es el conjunto de todos los nodos de la red, los cuales pueden solicitar y enviar paquetes de información entre ellos y A es el conjunto de todas las aristas dirigidas (i, j) que pertenecen a N. L_i es el conjunto de todos los nodos que pueden ser alcanzados por la comunicación del nodo i.

Se asume que el enlace (i, j) existe si y sólo si $j \in L_i$. Cada nodo i tiene un nivel de energía residual llamada e_i (cuyo valor inicial es E_0). La transmisión de una unidad de información entre dos nodos requiere e_{ij} energía. Se asume que la tabla de encaminamiento del nodo i tiene la siguiente estructura:

$$A_i = [a^i_{jd}(t)] = \begin{Bmatrix} a^i_{11}(t), a^i_{12}(t), ..., a^i_{1N}(t) \\ a^i_{21}(t), a^i_{22}(t), ..., a^i_{2N}(t) \\ ... \\ a^i_{L1}(t), a^i_{L2}(t), ..., a^i_{LN}(t) \end{Bmatrix} \tag{3.4}$$

Donde $i, d \in N$, $j \in L$ y cada renglón debe respetar la siguiente restricción [60]:

$$\sum_{j \in L_i} a^i_{jd} = 1; d \in [1, N] \tag{3.5}$$

a^i_{jd} representa la probabilidad de seleccionar desde el nodo actual i hacia el nodo destino d el nodo j y se define por [60]:

$$a^i_{jd} = \frac{\omega \psi_{jd}(t) + (1-\omega)\eta_j}{\omega + (1-\omega)(|N_i| - 1)} \tag{3.6}$$

$\omega \in [0, 1]$ es el factor de importancia y el denominador es un término normalizado. Las tablas de encaminamiento A_i son obtenidas por la integración parcial de los valores de feromona en los enlaces $\psi_{jd}(t)$ y los valores heurísticos η_j. Los valores de la feromona en los enlaces son calculados por [60]:

$$\psi_{ij}(t+1) = \rho \psi_{ij}(t) + \Delta \psi^{best}_{ij} \tag{3.7}$$

Donde $\rho \in [0, 1]$ y $\Delta \psi^{best}_{ij} = f^{best}(t)$. La función $f^{best}(t)$ es la mejor solución de la iteración. Esta función es el objeto crucial que necesita alcanzar un equilibrio entre el consumo de energía y el retardo de la información en las RIS. Los valores heurísticos se calculan como se indica a continuación [60]:

$$\eta_i = \frac{e_i}{\sum_{n \in N_i} e_n} \in [0, 1] \tag{3.8}$$

Un nodo tendrá menos oportunidad de ser seleccionado cuando contenga una pequeña cantidad de energía en él.

En este algoritmo todas las hormigas son identificadas dentro de dos tipos, las que van hacia la estación base (F) y las que van de regreso al nodo origen (B). Cada hormiga tiene una memoria tab_k la cual contiene la información de los nodos visitados en el recorrido de la misma. La memoria $L_i - \{tab_k\}$ es utilizada para definir, para cada hormiga k, el conjunto de nodos que una hormiga que inició desde el nodo i aun debe visitar.

3.2.5. Algoritmo de agregación de datos basado en ACO

Los autores Ning y Jie [61] proponen el algoritmo basado en ACO con agregación de datos (*Data Aggregation Based on ACO algorithm*, ACAR) cuya idea es la optimización del encaminamiento de datos con base en la agregación de datos por medio de agentes cooperativos llamados hormigas. Dentro de la heurística para la selección del próximo nodo a saltar se toman en cuenta tres factores: la energía de los nodos, la distancia entre dispositivos y la agregación de datos.

Durante el ciclo de agregación de datos, cada hormiga obtiene información cuantitativa acerca del costo de la ruta e información cualitativa sobre el nivel de tránsito en la RIS.

Con la consideración de la transmisión de datos bidireccional entre el nodo origen y el nodo destino, se utilizan dos tipos de agentes móviles llamados hormigas (AMA), las cuales son: las **AMA activas (AAMA)** que viajan desde el nodo origen hasta el nodo destino, con el objetivo de explorar nuevas rutas y obtener información de las mismas y, las **AMA pasivas (PAMA)** que son creadas por la estación base y viajan de vuelta al nodo origen y actualizan la información en cada nodo sensor en cada salto que dan.

Una hormiga AMA se coloca inicialmente en cada uno de los nodos antes de que los nodos hayan sido distribuidos en la RIS. Una vez recibida la tarea de encaminamiento, el AMA intenta buscar una ruta de costo mínimo en la red. La nueva ubicación de la AMA se implementa de acuerdo con una probabilidad de salto al siguiente nodo vecino. Similar al algoritmo ACO, la posición de AMA se puede actualizar con la siguiente ecuación [61]:

$$P_{i,j}^{k} = \begin{cases} \frac{[\tau_{i,j}]^{\alpha} \cdot [\eta_{i,k}]^{\beta}}{\sum_{s \in k_{libre}} [\tau_{i,s}]^{\alpha} \cdot [\eta_{i,s}]^{\beta}} & \forall j \in k_{libre} \\ 0 & \text{en otro caso} \end{cases} \tag{3.9}$$

Donde la $P_{i,j}^{k}$ es la probabilidad de salto del nodo i al nodo j para la AMA k. k_{libre} indica el nodo siguiente que está libre para ser seleccionado por AMA k. $\eta_{i,k}$ es el valor de la ganancia de agregación. $\tau_{i,j}$ es el valor de feromona. Los valores de feromona son guardados en la memoria del nodo. Cada nodo tiene información acerca de los niveles de feromona en los enlaces hacia los nodos vecinos. En el algoritmo ACO, se utiliza la evaporación de la feromona de los enlaces. La actualización de la feromona se realiza al final de cada viaje. Esta actualización se realiza con las siguiente regla [61]:

$$\tau_{i,j} = (1 - \rho) \cdot \tau_{i,j} + \rho \cdot \Delta\tau_{i,j} \tag{3.10}$$

Donde ρ es un coeficiente de control llamado tasa de evaporación de la feromona. $\rho \in (0, 1)$ es utilizada para medir la rapidez con que evolucionan los niveles de feromona. El incremento $\Delta\rho_{i,j}$ está dado por [61]:

$$\Delta\tau_{i,j}(t) = \begin{cases} \frac{Q}{d_{i,j}} & \text{sí la AMA } k \text{ pasó a través del } enlace(i,j) \\ 0 & \text{en otro caso} \end{cases} \tag{3.11}$$

Q representa la intensidad de la feromona y afecta la velocidad de convergencia de este algoritmo. En la ecuación 3.9 las cantidades α y β son dos parámetros que se mantienen constantes en todo el algoritmo.

3.2.6. Algunos otros ejemplos

En [62] se propone un algoritmo de encaminamiento para RIS basado en ACO, cuyo mecanismo se basa en la calidad de los enlaces o rutas, energía y algunos parámetros de velocidad. La arquitectura adoptada por esta propuesta ayudó a la RIS a mejorar el rendimiento general de las transmisiones de datos en la red, especialmente para el caso de tráfico en tiempo real.

En [63] se muestra un algoritmo de encaminamiento multiruta basado en *clustering* dinámico y ACO. Dicha propuesta pretende maximizar el tiempo de vida de la red y reducir el consumo de energía de la misma.

En [64] se presenta el algoritmo de encaminamiento de datos para sistemas de posicionamiento global llamado GPSAL, el cual está basado en GPS y agentes móviles que imitan el comportamiento de las hormigas con el fin de resolver el problema del encaminamiento en una red móvil *ad-hoc* (MANET). El sistema de la red está compuesta por dispositivos que tienen la capacidad de comunicación entre ellos de forma inalámbrica.

En [57] se muestran las propiedades de algunos algoritmos bioinspirados encargados del encaminamiento de datos para redes alambradas e inalámbricas. Además, se revisan las características principales de los mismos, cuyo diseño se ha inspirado en el comportamiento colectivo de grupos de insectos como las hormigas y las abejas.

En [65] se propone un enfoque novedoso llamado *Chip* encaminador basado en ACO (*Router Chip ACO based*), que consiste principalmente en la implementación de un componente de hardware pequeño como un *chip*, encargado de la parte de encaminamiento de datos de la RIS.

El algoritmo de encaminamiento consciente de la ubicación basado en ACO (*Ant Colony Optimization-Based Location-Aware Routing*, ACLR) propuesto en [66] es un algoritmo de encaminamiento de datos plano con base en la localización de los nodos sensores. Esta propuesta fusiona la energía residual de los nodos y la información local y global de la ubicación de los nodos de la red con el objetivo de definir la probabilidad de salto al siguiente vecino, con la cual las hormigas pueden moverse entre los nodos de la red.

En [10] está propuesto el algoritmo de encaminamiento basado en hormigas con eficiencia de energía (*Energy-Efficient Ant-Based Routing Algorithm*, EEABR) dirigido a infraestructuras planas basado en ubicación. En esta propuesta las hormigas buscan las rutas con menor consumo de energía, mientras que se utiliza una técnica de reducción de tamaño del paquete de información que contiene a la hormiga durante la comunicación entre nodos.

Tanto el algoritmo ACLR como el algoritmo EEABR serán explicados en detalle en el capítulo siguiente.

Capítulo 4

Metodología

En este capítulo se describen dos algoritmos de encaminamiento de datos para RIS seleccionados del estado del arte, que serán utilizados para llevar a cabo la comparación de su desempeño en términos de la maximización de la energía y minimización de la latencia.

Antes de entrar de lleno en la descripción de los algoritmos seleccionados, se darán un conjunto de definiciones con las que se trabajará en lo consecuente del documento.

4.1. Definiciones

Asumiremos que una RIS consiste de m nodos sensores estáticos y homogeneos. Los nodos se encuentran desplegados uniformemente en una región plana que es el área de interés. El área de comunicación que se encuentra dentro de la cobertura de cada nodo sensor es representada por una área circular cuyo centro es el nodo sensor y cuyo radio es r (figura 4.1).

Una RIS se describe formalmente como un grafo ponderado no dirigido[1] *G(V, E, L)*. V es el conjunto de nodos sensores y $V = \{s_1, s_2, s_3, ..., s_m\}$. L es el conjunto de los costos de las aristas del grafo. E es el conjunto de bordes, $E \subset V \times V \times L$, por ejemplo, para cualquier $s_i, s_j \in V$, $i \neq j$, $(s_i, s_j, \psi_{ij}(t)) \in E$, donde $\psi_{ij}(t))$ es el costo de enviar un paquete de datos desde s_i hasta s_j en el tiempo t; en este caso $\psi_{ij}(t)$ es la feromona entre s_i y s_j.

Cualquier nodo s_i tiene un conjunto de nodos vecinos $N(s_i)$ el cuál se define por:

$$N(s_i) = \{s_j | s_j \in V, d_{ij} \leq r\} \tag{4.1}$$

Donde r es el radio de cobertura de la comunicación inalámbrica de los nodos sensores. d_{ij} es la distancia entre s_i y s_j. Las coordenadas del nodo s_i son x_i y y_i y las coordenadas del nodo s_j son x_j y y_j, d_{ij} es calculada por:

[1]No dirigido significa que no es necesario asignar un sentido a las aristas del grafo

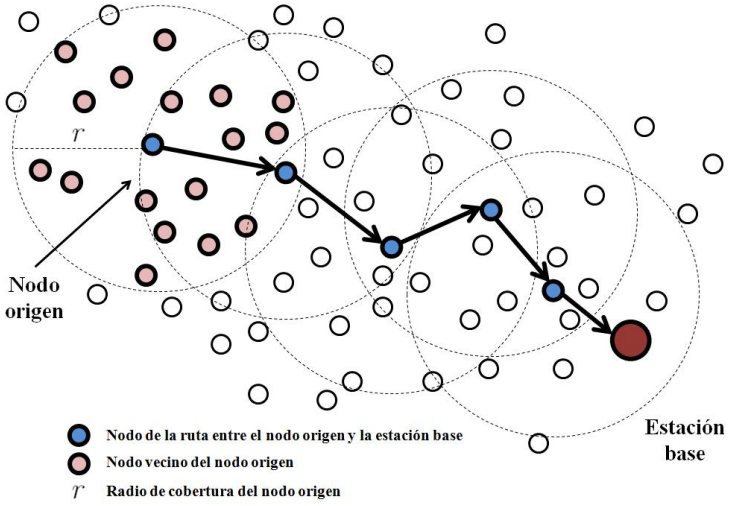

Figura 4.1: Radio de cobertura de los nodos sensores.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{4.2}$$

El algoritmo está compuesto por dos fases. La **primera fase** comienza con un conjunto de hormigas colocadas en el nodo origen s_o. Cuando la hormiga k está en el nodo s_i en el instante t, el siguiente nodo a saltar $s_j \in N(s_i)$ será seleccionado con probabilidad $P_{ij}^k(t)$. Cuando la hormiga k alcanza el nodo destino s_b la fase dos comienza.

En la **segunda fase** la hormiga k regresa al nodo origen s_o a lo largo de la misma ruta por la que llego al nodo destino, a la vez que deposita un incremento de feromona ψ_{ij} en la ruta.

4.2. Algoritmos seleccionados

A continuación se describen los algoritmos de encaminamiento que serán comparados.

4.2.1. Algoritmo de encaminamiento consciente de la ubicación basado en ACO (ACLR)

En [66] los autores Wang y Li describen un algoritmo para el encaminamiento de datos para RIS basado en ACO, llamado ACLR (por sus siglas en inglés *ACO Based Location Aware Routing for WSN*). El algoritmo ACLR intenta encontrar un equilibrio entre el tiempo de vida de la RIS y el retardo de la información en las transmisiones de datos.

ACLR supone que cada nodo en la RIS tiene un bloque de memoria donde se almacena tanto la energía residual del nodo actual como de sus vecinos, la información acerca

de la ubicación de los mismos y la información de la ubicación de la estación base. Cada hormiga es un agente móvil que tiene una lista de contraindicación T^k, en la que almacena los nodos por los que ha pasado en su ruta actual. La lista de contraindicación puede ayudar a cada hormiga a evitar seleccionar nodos por los que anteriormente haya pasado.

En ACLR no todos los nodos vecinos del nodo s_i son candidatos a ser seleccionados para ser el nodo siguiente en la ruta que la hormiga k va construyendo, sólo los nodos vecinos más cercanos al nodo destino son candidatos. Este conjunto es definido por [66]:

$$C(s_i) = \{s_j | s_j \in N(s_i), d_{jb} \leq d_{ib}\}. \tag{4.3}$$

La información de ubicación de los nodos tiene una influencia significativa en las probabilidades con las que la hormiga k selecciona al nodo vecino s_j como el siguiente nodo a saltar. Por lo anterior se define la función de ubicación ξ_{ij} como sigue [66]:

$$\xi_{ij} = \left(\frac{d_{ob}}{d_{oi} + d_{ij} + d_{jb}}\right) \cdot \left(1 - \frac{d_{ij}}{\sum_{s_l \in C(s_i)} d_{il}}\right) \tag{4.4}$$

Donde d_{ij} es la distancia desde el nodo s_i hasta el nodo s_j, d_{ob} es la distancia desde s_o hasta s_b, d_{oi} es la distancia desde s_o hasta s_i y d_{jb} es la distancia desde s_j hasta s_b. $0 \leq \xi_{ij} \leq 1$. Mientras más grande es el valor de ξ_{ij} más grande es la probabilidad con la que las hormigas seleccionan a s_j como el siguiente nodo a saltar. Si no hay ningún vecino al cual saltar, lo que significa que $C(s_i)$ está vacio, entonces la hormiga k regresa un salto al nodo previo de s_i. Sea s_l el nodo anterior del nodo s_i. Antes de que la hormiga k haga una reselección en el nodo s_l, s_i se suma a la lista de contraindicación de la hormiga k, con lo que la hormiga k no seleccionará a s_i como el siguiente nodo nuevamente.

ACLR establece que la energía residual de los nodos tiene una gran influencia sobre la probabilidad con la cual la hormiga k selecciona a s_j como el siguiente nodo a saltar. La función de energía $\eta_{ij}(t)$ se define como sigue [66]:

$$\eta_{ij}(t) = \frac{e_j(t)}{\sum_{s_l \in C(s_i)} e_l(t)} \tag{4.5}$$

Donde $e_l(t)$ es la energía residual de s_l en el instante t. Para considerar la información de la ubicación y la energía residual de los nodos se define la función de probabilidad de salto como sigue [66]:

$$P_{ij}^k(t) = \frac{[\psi_{ij}(t)]^\alpha \times [\xi_{ij}]^\beta \times [\eta_{ij}(t)]^\gamma}{\sum_{s_l \in C(s_i)} [\psi_{il}(t)]^\alpha \times [\xi_{ij}]^\beta \times [\eta_{il}(t)]^\gamma} \tag{4.6}$$

Donde α, β y γ son los pesos ajustables de la importancia de la feromona $\psi_{ij}(t)$, ξ_{ij}

y $\eta_{ij}(t)$ respectivamente. La selección del encaminamiento de las hormigas puede ser ajustado acorde a los diferentes valores de α, β y γ.

Las hormigas inician su ruta desde s_o hasta s_b y cuando una hormiga k alcanza el nodo destino s_b, la hormiga k ha encontrado una ruta R^k desde s_o hasta s_b, entonces la hormiga k inmediatamente inicia el viaje de regreso hacia s_o desde s_b a lo largo de la ruta R^k, mientras que al mismo tiempo la hormiga k actualiza los niveles de feromona en la ruta. Con n hormigas en la RIS una ronda completa se da cuando las n hormigas llegan a s_b desde s_o y luego regresan a s_o desde s_b a lo largo de las mismas rutas con las cuales las hormigas llegaron a s_b respectivamente. Supondremos que toma una unidad de tiempo para las hormigas terminar una ronda completa. La regla utilizada para actualizar la feromona en la ruta R_{ij} (el segmento entre s_i y s_j) está definido como se muestra a continuación [66]:

$$\psi_{ij}(t+1) = (1 - p(t)) \times \psi_{ij}(t) + \Delta\psi_{ij} \tag{4.7}$$

Donde $p(t)$ es la tasa de evaporación de feromona en el instante t, $0 \leq p(t) \leq 1$. $p(t)$ se calcula como a continuación se indica [66]:

$$p(t) = (1 - \eta_{ij}(t)) \cdot (1 - \xi_{ij}). \tag{4.8}$$

La fórmula anterior implica que la tasa de evaporación de la feromona $p(t)$ está en función de la energía residual y de la información de la ubicación de los nodos en lugar de una constante. Este esquema tiene una mejor adaptación a los cambios frecuentes de topología de la RIS. En la fórmula 4.7 $\Delta\psi_{ij}$ es el incremento de feromona en la ruta entre s_i y s_j en la ronda actual. $\Delta\psi_{ij}$ se calcula con la siguiente fórmula [66]:

$$\Delta\psi_{ij} = \sum_{k=1}^{n} \Delta\psi_{ij}^{k} \tag{4.9}$$

Donde $\Delta\psi_{ij}^{k}$ es la feromona que la hormiga k deposita en la ruta entre s_i y s_j en la ronda actual. $\Delta\psi_{ij}^{k}$ se calcula con [66]:

$$\Delta\psi_{ij}^{k} = \left\{ \begin{array}{cc} \frac{d_{ob} \times Q}{(d_{oi} + d_{ij} + d_{jb}) L^k} & \text{sí la hormiga } k \text{ pasó de } s_i \text{ a } s_j \\ \\ 0 & \text{en otro caso} \end{array} \right\} \tag{4.10}$$

Donde Q es una constante, d_{oi}, d_{ob}, d_{ij} y d_{jb} tienen el mismo significado que en la fórmula 4.4 respectivamente. L^k es la longitud de la ruta que ha sido encontrada por la hormiga k en la ronda actual.

En los experimentos realizados en [66] se puede observar que los autores utilizaron un valor arbitrario para el consumo de energía de los nodos sensores de la RIS. Ellos determinan que se consume una unidad de energía por la transmisión de un paquete de información entre dos nodos, no se toma en cuenta el tamaño del paquete de datos

mandado en la transmisión, el cual es vital para calcular el consumo de energía de la RIS. Puesto que los paquetes que se utilizan en este algoritmo son de tamaño variable, el consumo de energía que ellos calculan no es correcto. Para poder realizar una comparación justa y lo más realista posible entre diferentes algoritmos de encaminamiento de datos, se debe tener mucho cuidado en la manera de calcular el consumo de energía de cada algoritmo.

El algoritmo ACLR se describe en Algoritmo 2:

Algoritmo 2 Algoritmo ACLR.

```
1: mientras la condición de término de iteraciones no se alcance hacer
2:   si el nodo origen no tiene batería entonces
3:     termina
4:   fin si
5:   Inicializar hormigas en el nodo origen
6:   para k = 1 hasta n hacer
7:     mientras nodo actual ≠ nodo destino hacer
8:       Obtener vecinos elegibles
9:       si vecinos elegibles ≠ NULL entonces
10:        Saltar al vecino siguiente con base en la probabilidad de salto
11:      si no, si vecino anterior no tiene batería entonces
12:        termina
13:      si no
14:        Regresar al vecino anterior
15:      fin si
16:    fin mientras
17:    Calcular distancia recorrida
18:    Calcular promedio de retardo
19:  fin para
20:  Actualizar feromona (ψ)
21:  Obtener mejor solución
22:  Calcular resultados promedio (consumo de energía, retardo de envío de paquetes
     y eficiencia de energía)
23: fin mientras
```

4.2.2. Algoritmo de encaminamiento basado en hormigas con eficiencia de energía (EEABR)

En [10] los autores Camilo y Carreto proponen el algoritmo EEABR (por sus siglas en inglés *Energy Efficient Ant Based Routing Algorithm*), que optimiza el manejo del consumo de energía con el objetivo de maximizar el tiempo de vida de la RIS.

EEABR propone que las tablas de encaminamiento sólo contengan los identificadores de los nodos vecinos que estén en la dirección del nodo destino. Ésto reduce considerablemente el tamaño de la tabla de encaminamiento y, en consecuencia la memoria necesitada por los nodos.

La probabilidad con que las hormigas escogen al siguiente nodo esta definida por [10]:

$$P_{ij}^{k}(t) = \frac{[\psi_{ij}(t)]^{\alpha} \times [\xi_{ij}(t)]^{\beta}}{\sum_{s_l \in N(s_i)} [\psi_{il}(t)]^{\alpha} \times [\xi_{il}(t)]^{\beta}} \tag{4.11}$$

Donde α y β son los pesos ajustables de $\psi_{ij}(t)$ y $\xi_{ij}(t)$ respectivamente. ξ_{ij} es la función de ubicación definida por [10]:

$$\xi_{ij} = \frac{1}{d_{ij}} \tag{4.12}$$

Para considerar la calidad de la energía de las rutas en el algoritmo, la función propuesta para determinar la cantidad de feromona que la hormiga que viene depositará durante su viaje de regreso está definida como [10]:

$$\Delta\psi_k = \frac{1}{C - (avg(\eta_k) - \frac{1}{min(\eta_k)})} \tag{4.13}$$

Donde η_k es un vector llevado por la hormiga que va k con el nivel de energía de los nodos de la ruta, C es el nivel de energía inicial de los nodos, $avg(\eta_k)$ es el promedio de los valores de η_k y $min(\eta_k)$ es el mínimo valor de η_k.

Se ha probado que los trabajos realizados por el nodo sensor que son relacionados con comunicaciones (transmisión y recepción de información) gastan mucha más energía que esos relacionados con el procesamiento de datos y el manejo de la memoria [10].

EEABR propone que la memoria M_k de cada hormiga sea reducida a dos registros, los últimos dos nodos visitados. Puesto que el camino seguido por las hormigas no está en su memoria, una memoria debe ser creada en cada nodo para guardar los registros de cada hormiga que fue recibida y enviada. Cada registro guarda tanto el nodo previo como el nodo siguiente de la hormiga, su identificación y un temporizador del registro. Cada vez que una hormiga que va es recibida en el nodo busca dentro de su memoria la identificación de la hormiga, para evitar posibles ciclos infinitos, si no existe dicho registro, el nodo guarda la información requerida de la hormiga, reinicia el temporizador del registro y se envía a la hormiga al siguiente nodo, de lo

contrario si un registro contiene la identificación de la hormiga, entonces la hormiga es eliminada. Cuando un nodo recibe a una hormiga que viene, éste busca en su memoria para encontrar el siguiente nodo a donde la hormiga debe ser enviada. Si por alguna razón la hormiga no llega al nodo en el tiempo definido en el temporizador, se borra el registro de la memoria del nodo que identifica a la hormiga que viene.

El vector η_k fue borrado de las hormigas que van, las cuales ahora sólo llevan el promedio de energía hasta el nodo actual (ηavg_k) y el mínimo nivel de energía registrada (ηmin_k). Estos valores son actualizados por cada nodo que recibe a las hormigas que van. Cuando la hormiga que va alcanza el nodo destino estos valores son utilizados para calcular la cantidad de feromona que depositará la correspondiente hormiga que viene [10]:

$$\Delta\psi_k = \frac{1}{C - \left[\frac{\eta min_k - Fd_k}{\eta avg_k - Fd_k}\right]}. \tag{4.14}$$

Calcular $\Delta\psi_k$ solamente en función del nivel de energía de la ruta puede traer consigo rutas no optimizadas, puesto que un camino con quince nodos puede tener el mismo promedio de energía que una ruta con sólo cinco nodos. $\Delta\psi_k$ debe ser calculada como una función de los parámetros de nivel de energía y la distancia del camino. Ésto puede lograrse con la ayuda del parámetro Fd_k en la ecuación 4.14, el cual representa el número de nodos que la hormiga que va ha visitado.

La ecuación utilizada para actualizar las tablas de encaminamiento en cada nodo ha cambiado a [10]:

$$\psi_k(r,s) = (1-\rho)\cdot\psi_k(r,s) + \left[\frac{\Delta\psi_k}{\varphi Bd_k}\right] \tag{4.15}$$

Donde φ es un coeficiente y Bd_k es la distancia viajada (el número visitado de nodos) por la hormiga que viene k hasta el nodo r, estos dos parámetros forzarán a la hormiga a perder parte de la fuerza de la feromona durante su camino al nodo origen, la idea detrás de este comportamiento es construir una mejor distribución de la feromona (nodos cerca del nodo destino tendrán más niveles de feromona) y forzarán a nodos lejanos a encontrar un mejor camino. Tal comportamiento es extremadamente importante cuando el nodo destino es capaz de moverse, puesto que la adaptación de la feromona será más rápida. El Algoritmo 3 muestra el algoritmo EEABR.

Es importante notar que el punto clave del algoritmo de encaminamiento EEABR es minimizar el tamaño de los paquetes de datos (hormigas) transmitidos entre nodos, por lo tanto para poder realizar una comparación justa y realista entre los algoritmos ACLR y EEABR se debe tener mucho cuidado en la manera en que se lleva a cabo el cálculo del consumo de energía de la RIS.

Algoritmo 3 Algoritmo EEABR.

```
1: mientras la condición de término de iteraciones no se alcance hacer
2:   si el nodo origen no tiene batería entonces
3:     termina
4:   fin si
5:   Inicializar hormigas en el nodo origen
6:   para k = 1 hasta n hacer
7:     mientras nodo actual ≠ nodo destino hacer
8:       Obtener vecinos elegibles
9:       si vecinos elegibles ≠ NULL entonces
10:        Saltar al vecino siguiente con base en la probabilidad de salto
11:        El nodo se encarga de guardar la información de la hormiga que va
12:      fin si
13:    fin mientras
14:    Crear hormiga que viene e iniciar su viaje al nodo origen
15:    Calcular distancia recorrida
16:    Calcular promedio de retardo
17:  fin para
18:  Actualizar feromona (ψ)
19:  Obtener mejor solución
20:  Calcular resultados promedio (consumo de energía, retardo de envío de paquetes
     y eficiencia de energía)
21: fin mientras
```

4.2.3. Comparación entre ACLR y EEABR

En esta sección se presenta la comparación entre los algoritmos de encaminamiento para RIS llamados ACLR y EEABR.

4.2.3.1. Experimentos

Para realizar los experimentos se consideran los nodos Mica2 de la compañía Crossbow.

La energía requerida para transmitir un bit entre dos nodos es de 4.28 $\mu joules$ y la energía requerida para recibir un bit es de 2.36 $\mu joules$ [67]. Es un hecho que los nodos consumen energía simplemente por estar encendidos, e incluso apagados o en estado dormido, sin embargo en la simulación los nodos sólo se encuentran despiertos en los momentos que se requiere enviar información, la energía que se consume por estar en modo dormido puede ser ignorada y sólo ser tomada en cuenta la energía consumida en la transmisión y recepción de paquetes de datos. Con estos valores de consumo de energía queda establecido que la comparación entre los algoritmos ACLR y EEABR será más realista y justa que aquellas realizadas en sus propuestas originales [66, 10].

Los experimentos se ejecutaron con el uso de tres diferente escenarios que son definidos en [10] y tres indicadores para RIS usados en [66] para determinar el desempeño de los algoritmos seleccionados. Estos escenarios e indicadores están definidos a continuación:

Escenarios. ***Primer escenario:*** Todos los nodos sensores se inicializan con el mismo nivel de energía, sólo hay un nodo origen s_o y el nodo destino s_b es fijo. El nodo origen toma las coordenadas $x = 1$ y $y = 1$, mientras que el nodo destino toma $x = 299$ y $y = 199$, por lo que las rutas que se deben encontrar, tendrán que atravesar la RIS completa de lado a lado, mientras que el resto de nodos se despliegan aleatoriamente. En la Figura 4.2 se muestran las posiciones del nodo origen s_o y del nodo destino s_b.

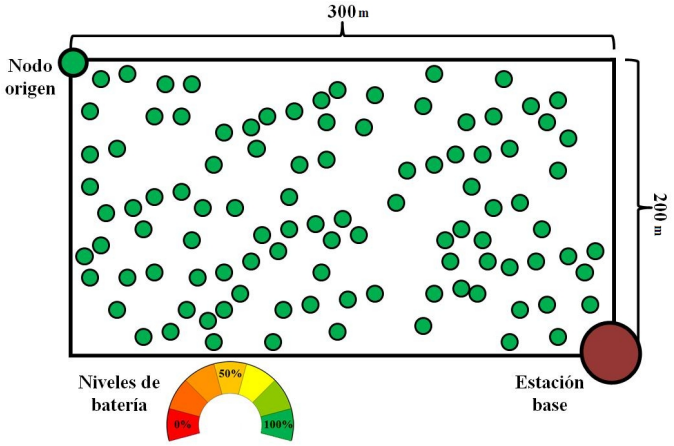

Figura 4.2: Primer escenario - Posiciones de s_o y s_b.

Segundo escenario: Todos los nodos se inicializan con el mismo nivel de energía, el nodo origen s_o cambia aleatoriamente en cada iteración y el nodo destino s_b está fijo. En la Figura 4.3 se muestran la posibles posiciones de nodo origen aleatorio s_o y la posición de la estación base s_b.

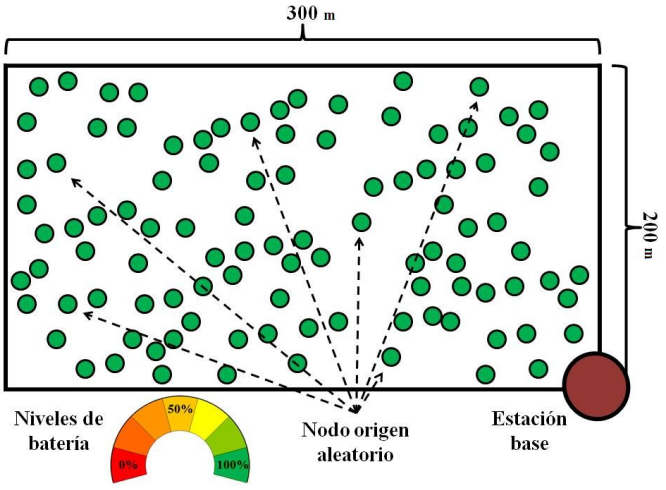

Figura 4.3: Segundo escenario - Posiciones de s_o y s_b.

Tercer escenario: El nivel de energía inicial de los nodos es seleccionado aleatoriamente, el nodo origen s_o cambia aleatoriamente en cada iteración y el nodo destino s_b es fijo. En la Figura 4.4 se observa la topología de la RIS en este escenario.

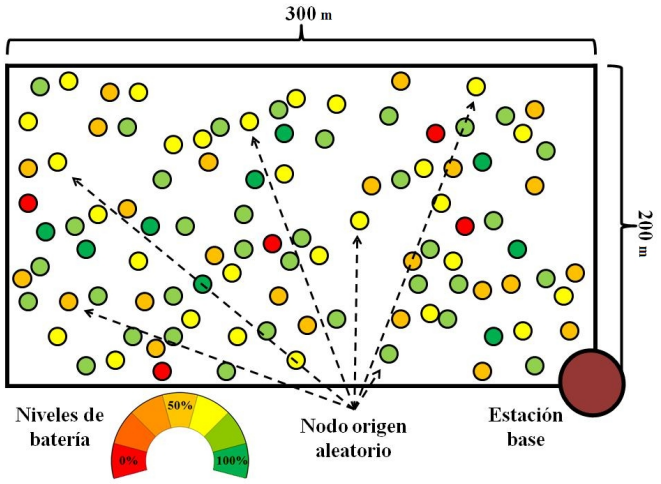

Figura 4.4: Tercer escenario - Posiciones de s_o y s_b.

Indicadores. Los indicadores que a continuación se describirán son los que normalmente se utilizan en el estado del arte para medir el desempeño del algoritmo de encaminamiento de datos de las RIS. Dichos indicadores son:

Consumo de energía: Este indicador se refiere al total de energía utilizada en la RIS en el proceso de encontrar las mejores rutas desde el nodo origen s_o hasta el nodo destino s_b. Para ser justos en la comparación de los algoritmos de encaminamiento, se utiliza el consumo de energía bit por bit que se muestra en el estudio realizado en [67], para la transmisión y recepción de datos en cada nodo. Como se mencionó anteriormente, la energía requerida para transmitir un bit entre dos nodos es 4.28 $\mu joules$ y la energía requerida para recibir un bit es 2.36 $\mu joules$.

Latencia o retardo: La latencia se refiere al tiempo que toma mandar un paquete de datos desde el nodo origen s_o hasta el nodo destino s_b. Normalmente la latencia es la suma de los retardos temporales dentro de la red. Este indicador se calcula como el promedio del número de nodos visitados por ruta.

Eficiencia de energía: Este indicador se refiere a la relación del número de paquetes recibidos por la estación base s_b entre el total de energía consumida.

Valores de los parámetros. Los valores de los parámetros son los mismos para ambos algoritmos, indicados a continuación:

- El área de despliegue de la RIS mide 300 x 200 m^2.
- 500, 2500, 5000 y 10000 nodos respectivamente se distribuyen en el entorno de interés de manera aleatoriamente uniforme.
- El número de hormigas utilizadas es igual a 20.
- El radio de comunicación inalámbrica de los nodos sensores es $r = 30$m.
- $\alpha = 1, \gamma = 1$. Para el algoritmo ACLR $\beta = 1$ y $Q = 1$.

- Para el algoritmo EEABR $\varphi = 1$ y la tasa de evaporación de feromona $p(t)$ igual a 0.95.
- El nivel de feromona inicial para cada enlace entre nodos adyacentes igual a $\psi_{ij}(0) = 0.01$.
- El consumo de energía por bit transmitido (definido en [67]) es 4.28 $\mu joules$.
- El consumo de energía por bit recibido es 2.36 $\mu joules$.
- Se realizaron 50 iteraciones en cada experimento.
- Para ambos algoritmos se corrieron 30 experimentos independientes para cada uno de los tres escenarios para 500, 2500, 5000 y 10000 nodos respectivamente.

4.2.3.2. Resultados

Análisis de los tamaños de las rutas. Los tamaños de las rutas (ruta menor y ruta mayor) encontrados durante los experimentos se muestran en la Tabla 4.1 para el escenario 1 tanto para el algoritmo ACLR como para el algoritmo EEABR, en la Tabla 4.3 para el escenario 2 y en la Tabla 4.4 para el escenario 3.

Tabla 4.1: Tamaño de rutas escenario 1.

	ACLR (número de nodos)								EEABR (número de nodos)							
	500		2500		5000		10000		500		2500		5000		10000	
Experimento	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor
1	19	38	18	36	18	35	18	37	19	42	21	40	20	44	20	42
2	19	36	17	36	19	39	18	39	20	40	22	41	21	44	21	43
3	18	36	19	38	19	37	18	36	20	40	21	45	21	42	22	44
4	19	39	19	37	19	37	18	37	22	43	19	41	19	43	20	44
5	19	39	18	37	19	37	18	36	21	41	20	39	18	43	20	43
6	19	36	19	36	17	37	16	38	21	38	21	42	20	41	21	45
7	19	40	18	38	18	38	19	36	19	39	21	40	19	43	22	44
8	18	37	19	41	19	37	19	36	19	40	21	43	20	44	22	45
9	19	35	18	40	18	37	18	38	21	44	20	42	20	43	20	44
10	18	35	18	38	18	38	17	37	19	41	20	41	21	41	21	44
11	18	35	18	38	18	39	17	38	18	39	19	41	21	42	19	41
12	18	35	19	37	18	36	18	38	19	40	21	43	22	43	20	42
13	19	37	18	36	18	37	19	35	20	39	21	43	21	42	21	48
14	19	37	18	37	18	39	19	35	20	38	21	42	21	43	22	43
15	17	36	18	38	18	37	18	37	22	40	21	41	21	40	21	42
16	19	36	18	37	18	43	19	40	20	40	20	42	20	42	21	42
17	19	36	18	37	19	35	18	37	20	39	21	40	21	41	20	46
18	18	35	18	38	18	37	19	39	21	40	18	44	20	43	21	44
19	19	36	18	39	19	37	18	37	21	45	19	40	21	43	21	42
20	19	36	18	38	18	38	19	38	21	41	21	45	21	41	21	43
21	17	35	18	38	18	38	18	40	21	40	21	42	22	43	21	44
22	19	34	19	38	18	37	19	39	20	40	20	43	20	40	20	49
23	18	36	18	36	18	38	19	37	20	41	21	42	21	43	21	46
24	17	36	19	39	18	37	18	36	22	44	21	42	21	46	21	43
25	19	36	18	39	19	40	19	37	21	43	20	42	19	40	19	41
26	17	42	18	38	18	38	18	38	21	45	20	47	22	41	22	44
27	18	37	18	40	19	36	16	39	19	41	20	41	20	41	22	42
28	19	37	18	37	18	37	17	37	21	40	19	41	20	44	21	42
29	18	38	19	37	18	37	19	36	21	44	21	44	20	44	21	51
30	19	38	19	37	19	38	18	36	20	40	21	43	19	45	20	42
Promedio	**18**	**36**	**18**	**37**	**18**	**37**	**18**	**37**	**20**	**40**	**20**	**42**	**20**	**42**	**20**	**43**

Los promedios totales son mostrados en la Tabla 4.2, donde se puede observar que el número de nodos desplegados en el área de interés, no afecta tanto al número de nodos que se encuentran por ruta. Lo anterior significa, que los algoritmos ACLR y EEABR buscan consistentemente las rutas más cortas entre el nodo origen s_o y la estación base s_b y evitan saltos innecesarios.

Con base en los valores obtenidos de todos los experimentos juntos (500, 2500, 5000 y 10000 nodos) se encontró que la ruta más corta fue de 16 saltos y la mayor de 51. En promedio en todos los experimentos realizados se tuvieron rutas de 29 nodos.

Tabla 4.2: Promedio del tamaño de rutas por número de nodos desplegados.

	Escenario 1				**Escenario 2**				**Escenario 3**			
Número de nodos	**500**	**2500**	**5000**	**10000**	**500**	**2500**	**5000**	**10000**	**500**	**2500**	**5000**	**10000**
ACLR	27.53	27.98	27.92	27.72	18.57	19.92	20.87	21.58	18.53	19.77	21.02	21.83
EEABR	30.60	31.23	31.45	32.32	19.28	21.07	23.05	24.10	19.55	20.28	21.00	21.97

Tabla 4.3: Tamaño de rutas escenario 2.

	ACLR (número de nodos)								**EEABR (número de nodos)**							
	500		**2500**		**5000**		**10000**		**500**		**2500**		**5000**		**10000**	
Experimento	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor
1	1	35	1	36	1	40	2	39	4	38	2	41	3	44	3	43
2	2	35	1	37	2	39	1	43	3	37	2	42	2	45	2	48
3	1	35	2	37	1	40	2	41	2	36	3	40	2	44	3	45
4	2	37	3	34	2	41	1	44	4	38	2	39	3	40	2	48
5	2	38	1	39	2	41	2	42	3	36	2	40	1	41	3	47
6	1	36	2	35	1	41	1	42	3	35	2	42	3	53	3	46
7	2	36	1	38	1	38	2	41	2	39	2	38	2	40	3	44
8	2	34	1	38	1	42	1	45	4	36	3	39	3	40	3	46
9	1	35	2	37	1	37	3	42	2	37	2	38	3	42	2	46
10	2	35	2	43	3	40	2	38	1	38	2	39	4	41	2	40
11	1	35	1	41	1	39	2	41	1	37	2	40	2	45	3	44
12	2	35	2	36	1	37	3	42	2	34	2	36	3	49	3	41
13	1	37	1	38	1	40	1	46	2	36	2	39	2	47	2	53
14	1	35	1	38	1	37	2	43	2	37	3	39	3	41	2	44
15	2	34	2	43	1	40	2	39	4	35	3	45	3	43	2	46
16	1	38	1	38	1	45	1	40	3	36	2	39	2	45	3	49
17	1	35	1	37	1	40	1	42	3	37	4	42	4	41	2	45
18	2	38	2	38	2	43	2	42	2	37	3	42	2	44	2	50
19	1	34	1	39	3	39	2	42	1	34	2	37	3	44	3	46
20	3	38	1	40	1	40	1	42	3	35	3	43	2	41	2	46
21	1	33	1	41	2	40	2	39	1	38	3	39	2	45	3	43
22	2	37	1	40	2	41	2	42	4	34	2	37	2	41	3	42
23	1	33	3	39	1	44	2	41	2	36	3	38	3	43	3	45
24	1	40	2	37	1	41	1	45	3	34	2	41	4	42	3	48
25	2	35	2	41	1	38	2	41	4	35	1	40	3	42	2	46
26	1	34	1	37	1	42	2	41	1	38	3	42	3	45	4	44
27	3	37	2	37	2	48	1	40	2	33	2	41	3	45	2	46
28	2	35	2	44	2	41	1	43	2	36	2	39	3	41	2	45
29	2	35	2	36	2	38	1	38	2	35	3	35	3	41	3	45
30	1	35	1	35	1	37	1	40	1	37	2	41	3	47	3	47
Promedio	**1**	**35**	**1**	**38**	**1**	**40**	**1**	**41**	**2**	**36**	**2**	**39**	**2**	**43**	**2**	**45**

En el escenario 2 (tabla 4.1) donde todos los nodos empiezan con el mismo nivel de energía, el nodo origen s_o cambia aleatoriamente en cada iteración y el nodo destino s_b está fijo, se encontró que la ruta menor fue de 1 salto, debido a que el nodo origen s_o cayó cerca de la estación base, por otra parte la ruta mayor encontrada fue de 53 nodos. En promedio en todos los experimentos realizados se tuvo rutas de 21 nodos.

En la Tabla 4.2 se observa el mismo fenomeno que en el escenario anterior, el número de nodos desplegados en la RIS no afecta a los algoritmos ACLR y EEABR en el caso de los tamaños de rutas encontradas.

En el escenario 3 (tabla 4.1) cuando el nivel de energía inicial de los nodos es seleccionado aleatoriamente, el nodo origen s_o cambia aleatoriamente en cada iteración y el nodo destino s_b es fijo, se encontró la menor ruta de 1 salto, debido también a que el nodo origen s_o cae de forma aleatoria cerca de la estación base s_b y la mayor ruta encontrada fue de 46 nodos. En promedio en todos los experimentos realizados se tuvieron rutas de 20 nodos, un valor muy cercano al escenario 2.

En este escenario los valores obtenidos entre los algoritmos son más cercanos entre sí, sin embargo se establece que ACLR trabaja mejor respecto al tamaño de rutas encontradas que EEABR. En la Tabla 4.2 se observa que el algoritmo se mantiene constante en términos de tamaños de rutas, sin importar el número de nodos desplegados en la RIS.

Tabla 4.4: Tamaño de rutas escenario 3.

	ACLR (número de nodos)								EEABR (número de nodos)							
	500		2500		5000		10000		500		2500		5000		10000	
Experimento	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor	Menor	Mayor
1	3	36	2	36	2	45	2	44	2	35	3	35	2	40	2	41
2	1	36	2	40	1	41	1	42	3	36	2	36	3	41	2	38
3	2	36	2	36	2	39	1	41	3	37	2	41	3	41	3	41
4	2	39	1	39	2	44	1	42	3	36	3	37	2	41	2	40
5	1	34	1	37	1	41	2	41	3	38	3	38	3	38	3	40
6	1	33	1	40	1	38	2	41	2	38	2	36	2	43	3	43
7	4	38	2	38	2	43	1	42	2	36	3	41	3	38	3	43
8	2	34	1	39	1	43	2	43	2	34	3	36	2	39	2	42
9	1	36	1	36	2	38	2	43	4	37	3	35	2	38	2	44
10	2	34	2	39	3	42	3	41	3	37	1	38	3	41	2	41
11	1	38	1	38	1	37	1	42	3	37	2	37	3	37	4	39
12	1	34	2	37	2	45	1	39	4	39	3	40	3	42	2	42
13	2	33	2	38	1	39	3	41	3	36	3	39	3	38	3	42
14	1	35	2	36	2	43	1	43	3	34	3	39	4	41	2	38
15	2	37	1	40	2	39	1	43	2	37	3	41	2	40	3	44
16	1	35	2	36	1	41	2	44	4	35	2	38	2	39	3	41
17	1	39	2	37	1	38	2	38	3	40	3	36	2	40	3	40
18	1	37	1	39	1	44	1	45	2	36	2	36	2	41	4	43
19	2	37	1	41	2	36	3	43	2	34	3	39	2	36	2	42
20	2	34	1	37	1	40	1	41	4	37	3	44	2	41	3	44
21	1	36	2	42	2	43	1	43	1	37	2	38	2	36	3	41
22	2	32	2	40	1	42	3	40	4	34	3	39	2	37	2	43
23	3	35	1	39	1	39	1	41	1	34	2	41	2	41	4	39
24	1	35	2	39	2	42	1	42	3	37	2	38	2	38	2	42
25	2	34	1	37	1	39	1	42	2	43	2	39	2	42	2	42
26	1	35	1	38	2	40	2	44	2	34	2	38	3	38	4	43
27	1	35	1	35	1	38	2	42	4	37	3	38	2	40	2	40
28	1	36	2	36	2	39	2	42	1	33	3	35	3	41	2	39
29	2	38	1	39	2	40	2	40	5	38	2	36	3	39	3	42
30	2	36	2	37	1	37	1	46	2	35	3	37	3	39	2	40
Promedio	**1**	**35**	**1**	**38**	**1**	**40**	**1**	**42**	**2**	**36**	**2**	**38**	**2**	**39**	**2**	**41**

Indicadores del primer escenario. Los valores reportados en los indicadores siguientes son los que se obtuvieron en los experimentos donde se desplegaron 10000 nodos de manera aleatoria en toda el área de interés.

Los valores de los indicadores obtenidos en el primer escenario se muestran en las Figuras 4.5, 4.6 y 4.7, para el consumo de energía, la eficiencia de energía y la latencia de la información, respectivamente.

Claramente se observa que el algoritmo EEABR supera en el consumo de energía al algoritmo ACLR. Este resultado es debido al tamaño de los paquetes de información (hormigas) que utiliza EEABR (las hormigas sólo recuerdan los últimos dos nodos por donde han pasado).

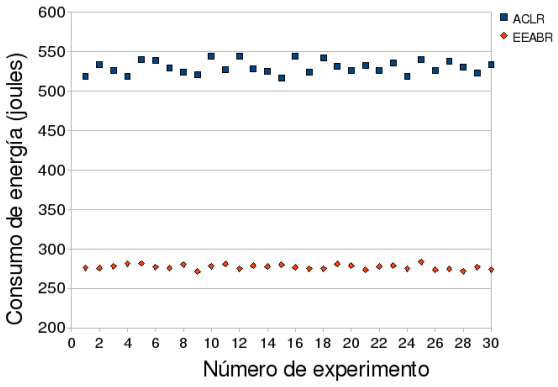

Figura 4.5: Primer escenario. Consumo de energía.

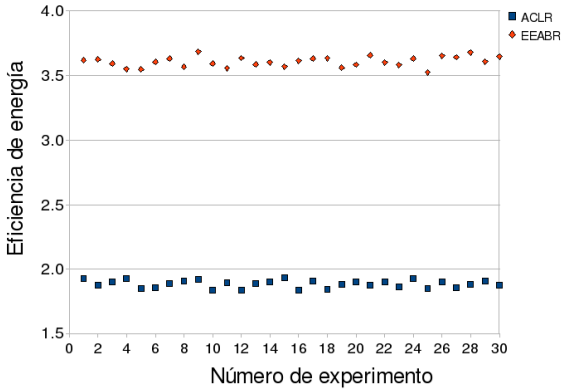

Figura 4.6: Primer escenario. Eficiencia de energía.

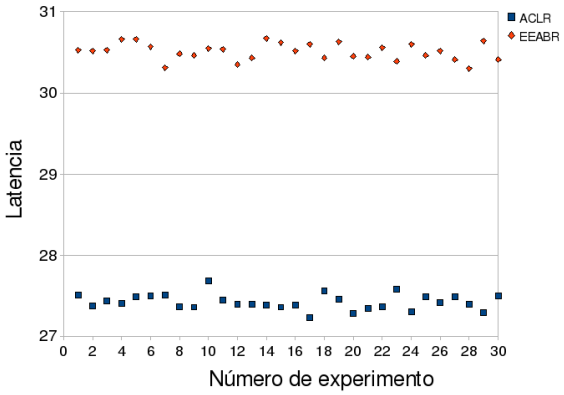

Figura 4.7: Primer escenario. Latencia.

El algoritmo ACLR supera por algunos nodos (2 o 3 nodos) al algoritmo EEABR en todos los 30 experimentos en el indicador de latencia, lo que significa que ACLR se concentra más en minimizar el retardo de la información.

En general en el indicador de eficiencia de energía EEABR es más eficiente que el algoritmo ACLR.

En las Figuras 4.8, 4.10 y 4.12 se muestran los valores del consumo de energía, latencia y eficiencia de energía obtenidos en los experimentos realizados con distintas cantidades de nodos desplegados en la red (500, 2500, 5000 y 10000) para el algoritmo ACLR en el escenario uno. De igual manera las Figuras 4.9, 4.11 y 4.13 muestran los resultados mencionados para el algoritmo EEABR.

Se observa que con las características que define el escenario uno los indicadores de consumo de energía y de latencia parecen tener una correlación muy grande entre sí, en este caso parecen ser directamente proporcionales, además el indicador de eficiencia de energía parece ser inversamente proporcional al consumo de energía.

Lo anterior se debe a que con las características indicadas por el escenario 1 todos los paquetes que se capturan del medio ambiente por el nodo origen s_o llegan al nodo destino s_b, no existe pérdida de paquetes en las transmisiones, por lo que el consumo de energía tiene relación directa con el número de paquetes enviados y la latencia.

Como el indicador de eficiencia de energía es la razón entre el número de paquetes recibidos por s_b entre el consumo total de energía de la RIS, luego entonces la eficiencia de energía es inversamente proporcional al consumo de energía.

Hay que mencionar que el escenario uno representa a una red en condiciones iniciales de funcionamiento, es decir, cuando todos los nodos sensores tienen sus baterías al mismo nivel de energía, razón por la cual se dan los valores de los indicadores de desempeño con una gran correlación entre sí.

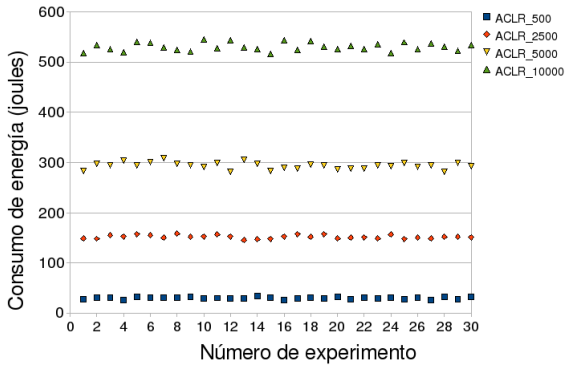

Figura 4.8: Primer escenario. Consumo de energía de ACLR con diferente número de nodos desplegados.

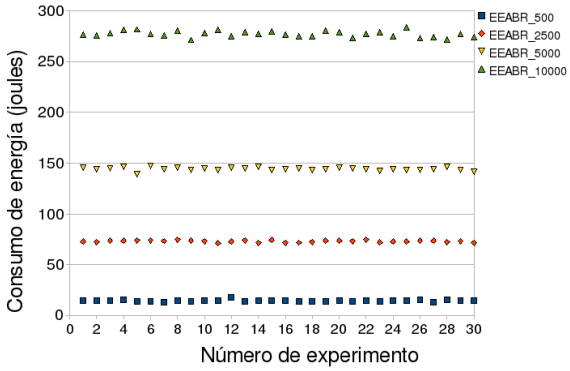

Figura 4.9: Primer escenario. Consumo de energía de EEABR con diferente número de nodos desplegados.

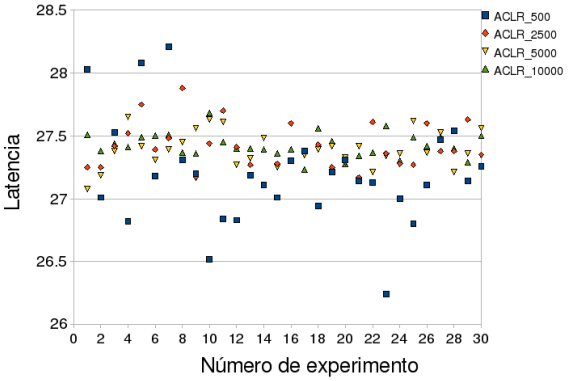

Figura 4.10: Primer escenario. Latencia de ACLR con diferente número de nodos desplegados.

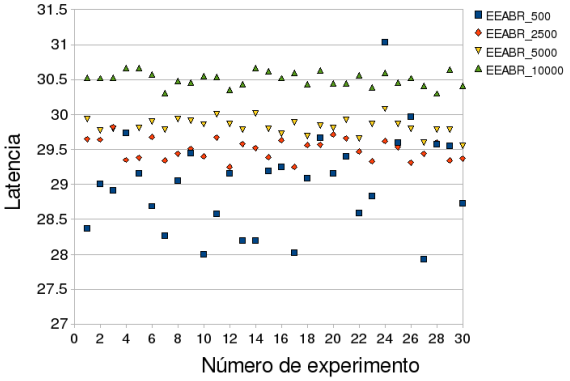

Figura 4.11: Primer escenario. Latencia de EEABR con diferente número de nodos desplegados.

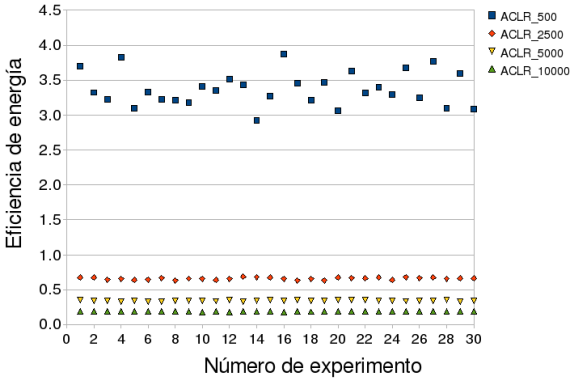

Figura 4.12: Primer escenario. Eficiencia de energía de ACLR con diferente número de nodos desplegados.

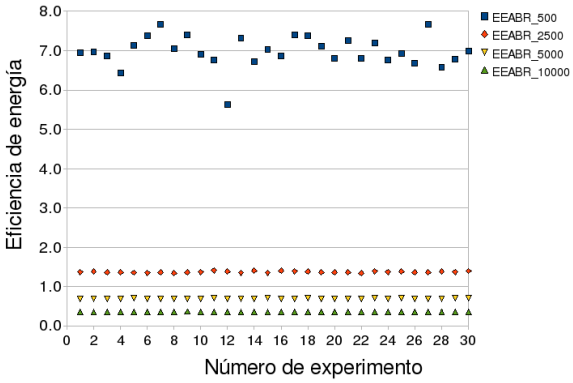

Figura 4.13: Primer escenario. Eficiencia de energía de EEABR con diferente número de nodos desplegados.

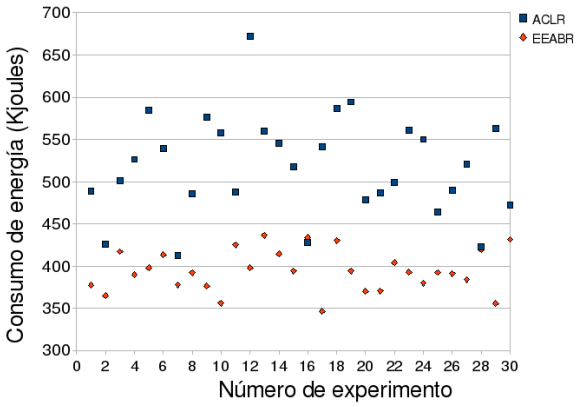

Figura 4.14: Segundo escenario. Consumo de energía.

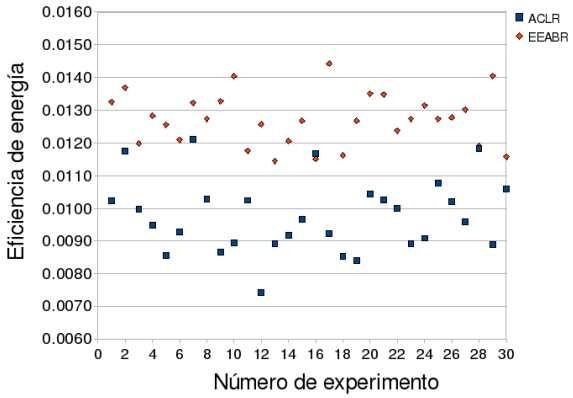

Figura 4.15: Segundo escenario. Eficiencia de energía.

Indicadores del segundo escenario. Los resultados para los indicadores de consumo de energía, eficiencia de energía y latencia son mostrados en las Figuras 4.14, 4.15 y 4.16, respectivamente para el segundo escenario.

Para el indicador de consumo de energía en el segundo escenario el algoritmo EEABR mejora el rendimiento del algoritmo ACLR en la mayoría de los casos (29 de 30 experimentos), con una diferencia menor que la observada en el escenario anterior. Lo anterior se puede explicar porque el nodo origen cambia constantemente su posición, por lo que las rutas encontradas son mucho más diversas.

Para el indicador de latencia ACLR supera a EEABR en la mayoría de los experimentos (25 de 30).

En el indicador de eficiencia de energía se puede ver nuevamente que el algoritmo EEABR es mejor que el algoritmo ACLR. Los resultados para este segundo escenario son consistentes con los obtenidos en el primer escenario.

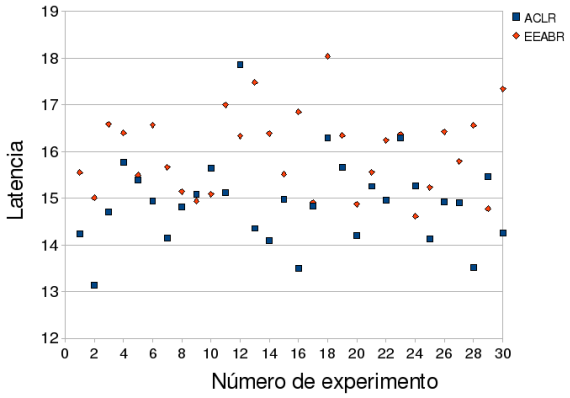

Figura 4.16: Segundo escenario. Latencia.

En las Figuras 4.17, 4.19 y 4.21 se muestran los indicadores obtenidos en los experimentos realizados con las cantidades de 500, 2500, 5000 y 10000 nodos desplegados en la red para el algoritmo ACLR, con las características indicadas por el segundo escenario.

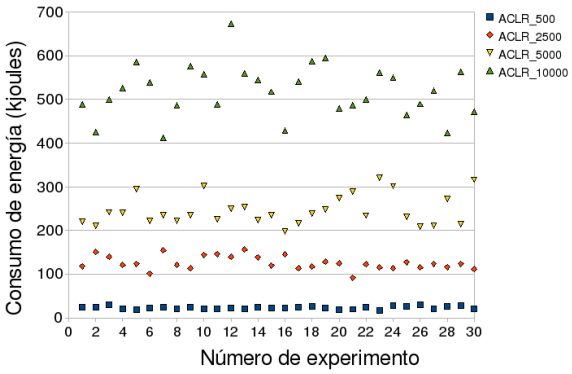

Figura 4.17: Segundo escenario. Consumo de energía de ACLR con diferente número de nodos desplegados.

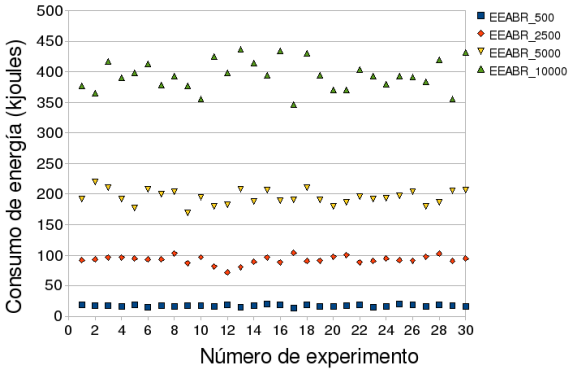

Figura 4.18: Segundo escenario. Consumo de energía de EEABR con diferente número de nodos desplegados.

De igual manera, las Figuras 4.18, 4.20 y 4.22 muestran los resultados del algoritmo EEABR.

El algoritmo EEABR pierde un poco de la tendencia que mostró en el escenario anterior, donde el número de nodos marcaba una importante influencia en el consumo de energía en la red, pues en este escenario no parece haber una relación directa entre el número de nodos de la red y el consumo de energía de la misma.

La correlación tan alta entre los valores de los indicadores en el algoritmo EEABR se debe principalmente, por su propuesta de encaminamiento de datos, donde establece que el paquete que lleva a la hormiga nunca cambia de tamaño, lo que significa que la cantidad de energía consumida por llevar una hormiga a través de una ruta, es directamente proporcional a la cantidad de saltos de nodo en dicha ruta.

Lo anterior puede explicar también dicha correlación en el algoritmo ACLR, puesto que en este algoritmo los paquetes que contienen y transportan a las hormigas sí cambian de tamaño conforme la memoria de la hormiga agrega más nodos a la ruta. Sin embargo los números promedio de nodos por ruta afectan muy poco el cambio o diferencia entre la variable de consumo de energía y la latencia. Al final la característica principal de ACLR (encontrar rutas cortas) es la causante de que el consumo de energía sea casi directamente proporcional a la latencia.

Indicadores del tercer escenario. Para el tercer escenario, los valores de los indicadores para consumo de energía, eficiencia de energía y latencia están mostrados en las Figuras 4.23, 4.24 y 4.25 respectivamente.

Se puede observar que los valores del consumo de energía para el caso de ambos algoritmos tiene un desempeño muy parecido.

Los valores de la latencia para ambos algoritmos son similares en este escenario y similares al segundo escenario también, sin embargo ACLR supera a EEABR. En el indicador de eficiencia de energía EEABR obtuvo mejores resultados sobre ACLR en todos los experimentos.

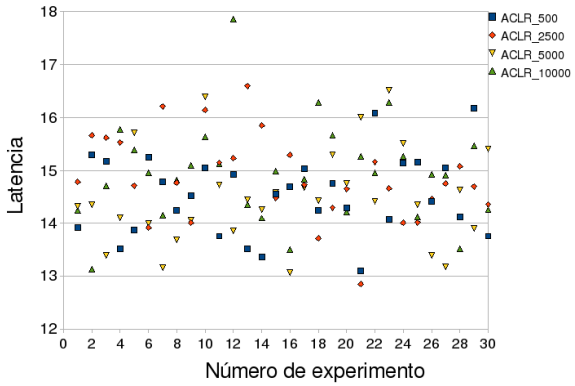

Figura 4.19: Segundo escenario. Latencia de ACLR con diferente número de nodos desplegados.

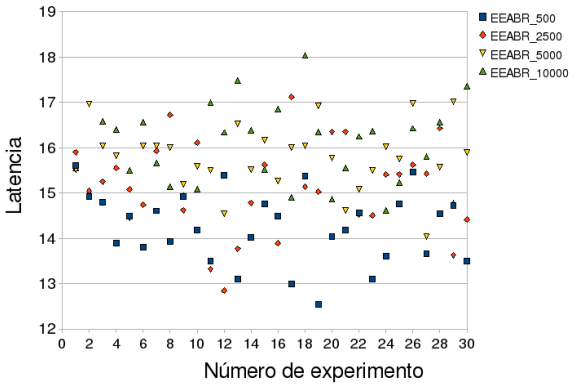

Figura 4.20: Segundo escenario. Latencia de EEABR con diferente número de nodos desplegados.

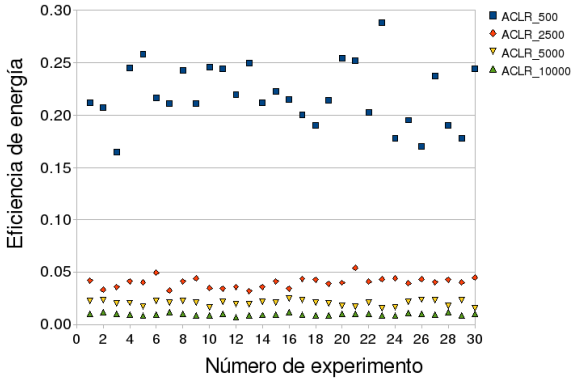

Figura 4.21: Segundo escenario. Eficiencia de energía de ACLR con diferente número de nodos desplegados.

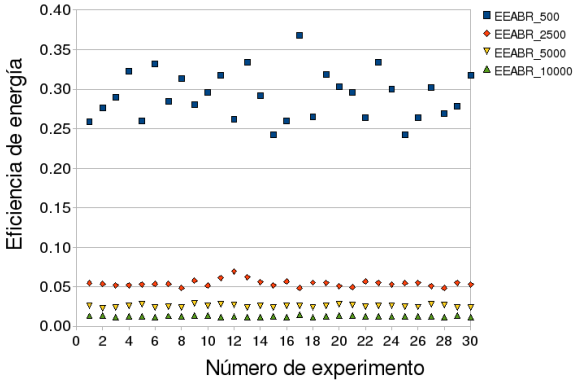

Figura 4.22: Segundo escenario. Eficiencia de energía de EEABR con diferente número de nodos desplegados.

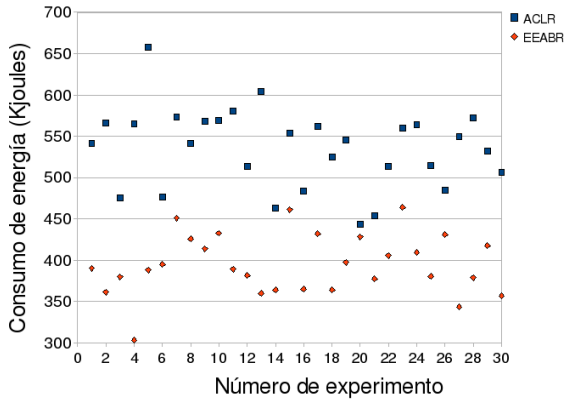

Figura 4.23: Tercer escenario. Consumo de energía.

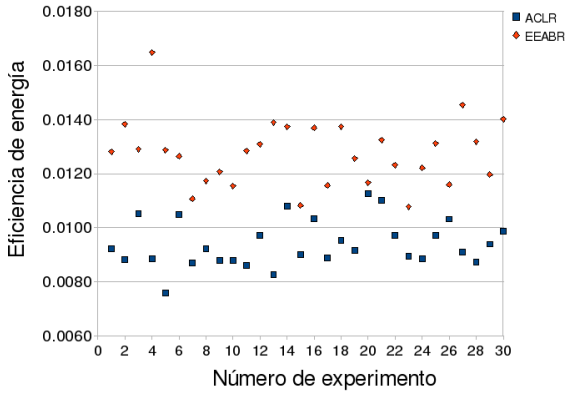

Figura 4.24: Tercer escenario. Eficiencia de energía.

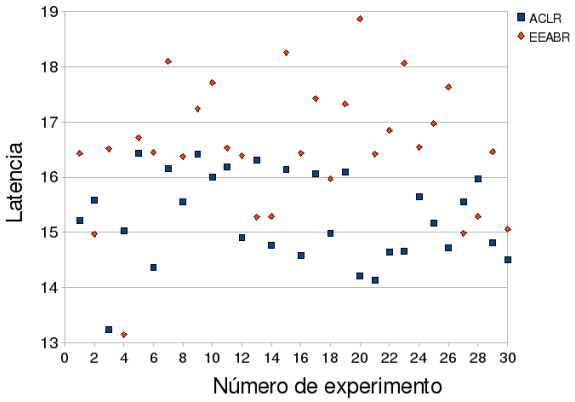

Figura 4.25: Tercer escenario. Latencia.

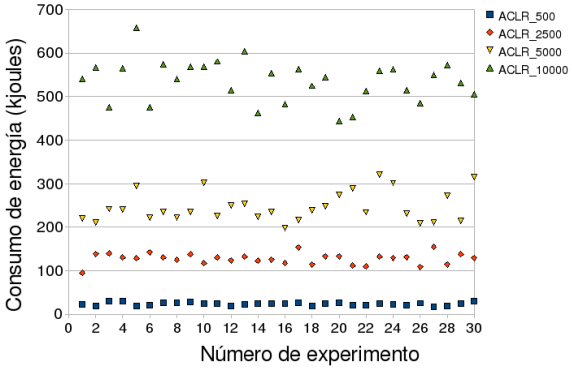

Figura 4.26: Tercer escenario. Consumo de energía de ACLR con diferente número de nodos desplegados.

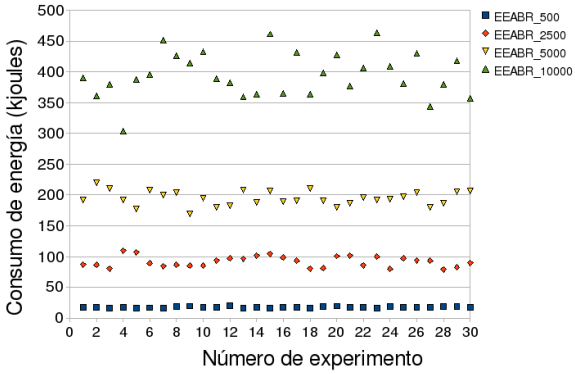

Figura 4.27: Tercer escenario. Consumo de energía de EEABR con diferente número de nodos desplegados.

Las Figuras 4.26, 4.28 y 4.30 muestran los valores de los indicadores en los experimentos con 500, 2500, 5000 y 10000 nodos desplegados en la red para el algoritmo ACLR, con las características indicadas por el tercer escenario. De igual manera, las Figuras 4.27, 4.29 y 4.31 muestran los resultados del algoritmo EEABR.

El tercer escenario indica que ya se han consumido cantidades de energía importantes en todos o casi todos los nodos de la red, se puede decir que en este escenario es donde se debe observar de mejor manera la forma de trabajo del algoritmo de encaminamiento de la RIS, puesto que el nivel de energía de los nodos pasa a ser un criterio muy importante en la probabilidad de salto de los algoritmos. En este escenario la relación entre las variables disminuye, manteniéndose claro está, la correlación directa entre el consumo de energía y la latencia en el algoritmo EEABR (por lo explicado en el escenario anterior).

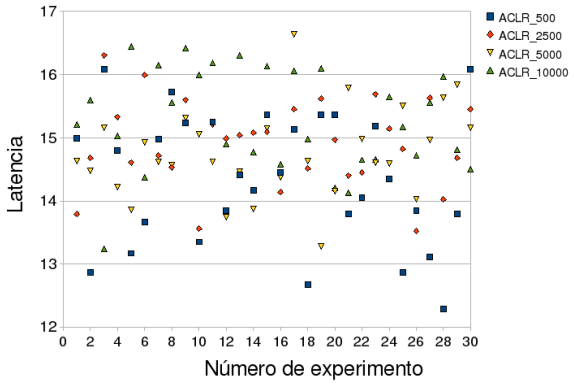

Figura 4.28: Tercer escenario. Latencia de ACLR con diferente número de nodos desplegados.

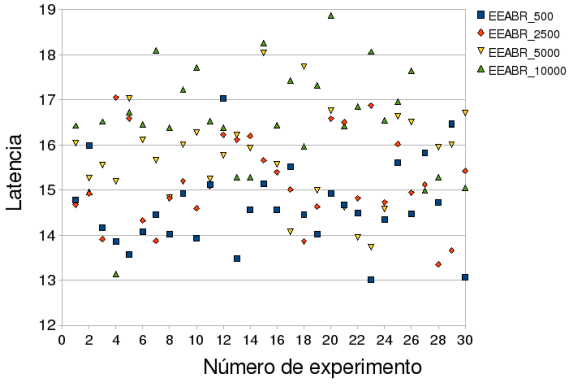

Figura 4.29: Tercer escenario. Latencia de EEABR con diferente número de nodos desplegados.

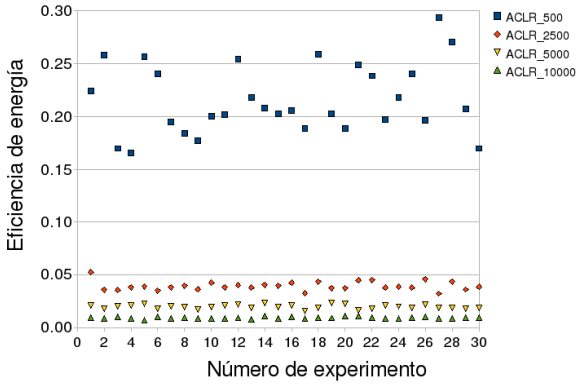

Figura 4.30: Tercer escenario. Eficiencia de energía de ACLR con diferente número de nodos desplegados.

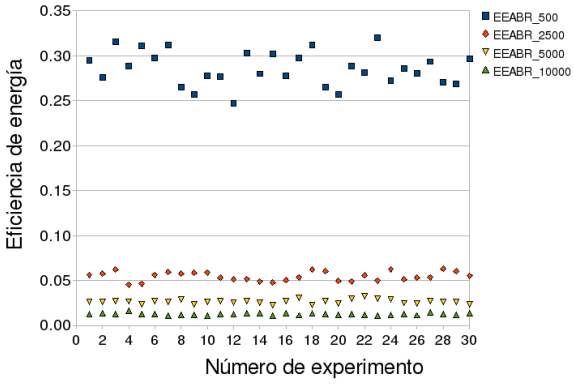

Figura 4.31: Tercer escenario. Eficiencia de energía de EEABR con diferente número de nodos desplegados.

4.2.3.3. Conclusiones de la comparación entre ACLR y EEABR

Tanto ACLR y EEABR son algoritmos de encaminamiento de datos para RIS los cuales están basados en ACO. Dichos algoritmos se han programado con el objetivo de realizar una simulación del comportamiento tanto del ACLR como del EEABR, con el objetivo de observar el desempeño de ambos. Con base en los resultados mostrados en la sección anterior, se concluye lo siguiente.

El número de nodos desplegados en la RIS no es un factor que influya de gran manera al desempeño de los algoritmos comparados en cuestión de búsqueda de rutas cortas. Por parte del algoritmo ACLR este factor no incide en que las rutas que se encuentren varíen en tamaño. En el caso del primer escenario el algoritmo ACLR fue el que encontró las rutas más cortas, por lo anterior se establece que el algoritmo ACLR tiene mejor desempeño que el algoritmo EEABR, respecto a la cuestión de encontrar rutas más cortas, lo cual lo logra por el tipo de heurística utilizada en la selección del siguiente nodo a saltar en la ruta de las hormigas.

Debido a la propuesta del algoritmo EEABR respecto al manejo del control del movimiento de las hormigas a través de la RIS, este algoritmo resulto ser mejor respecto al consumo de energía que el algoritmo ACLR.

No obstante, que las rutas encontradas por EEABR tienen más saltos que las rutas que encuentra el algoritmo ACLR, EEABR supera al ACLR en el nivel de consumo de energía global de la RIS. Lo anterior se logra con el uso de la técnica de reducción de tamaño de paquetes de información descrita por el algoritmo EEABR, donde se establece que la memoria de las hormigas sólo deben recordar los últimos dos nodos de su ruta y, los encargados de controlar el tránsito de hormigas a través de la RIS son los nodos sensores.

El algoritmo EEABR proporciona mayores niveles en la eficiencia de energía que el algoritmo ACLR, por lo que EEABR obtiene mayores tiempos de vida útil de la red.

Como se ha mencionado cada uno de los algoritmos tiene puntos fuertes y puntos débiles, según la aplicación para la que la RIS esté diseñada, entonces será mejor uno u otro algoritmo de encaminamiento. ACLR es el mejor para la aplicación que requiera la entrega de la información capturada del ambiente en el menor tiempo posible y EEABR para cuando se necesita que la RIS sea útil el mayor tiempo de vida posible.

4.3. Algoritmo de encaminamiento propuesto para mejorar el tiempo de vida de la RIS (IACAR)

Basados en los experimentos y resultados obtenidos en la sección anterior, se propone el algoritmo mejorado de encaminamiento basado en ACO para WSN (*Improve Ant Colony Based Routing Algorithm for WSN*, IACAR) que toma las partes principales de cada uno de los algoritmos comparados anteriormente (ACLR y EEABR), para lograr un algoritmo con lo mejor de ambos.

La característica que IACAR toma de ACLR es la heurística utilizada en el cálculo de la probabilidad de salto al siguiente nodo, con el fin de obtener la rutas más cortas entre el nodo origen que captura la información del medio y la estación base. A su vez IACAR toma de EEABR el método de control de movimiento de hormigas a través de los nodos de la red, donde se tiene un tamaño pequeño y constante de los paquetes de información que se transmiten en la red.

A continuación se describe el algoritmo IACAR y se compara con los algoritmos ACLR y EEABR.

Un nodo sensor tiene una cobertura de comunicación r. Cada nodo i conoce cuáles son sus nodos vecinos C_{s_i} dentro de su radio de cobertura r, sus ubicaciones (x, y) y cuales son los niveles de feromona ψ_{ij} en los enlaces con cada nodo vecino j donde $j \in C_{s_i}$.

En IACAR se hace una diferencia importante entre las hormigas que van desde el nodo origen s_o hasta el nodo destino s_b y las hormigas que vienen desde el nodo s_b hasta el nodo s_o.

Cuando se requiere encontrar rutas factibles entre un nodo origen s_o y la estación base s_b, se colocan en s_o un conjunto de hormigas que son capaces de encontrar dichas rutas. Una hormiga k tiene la capacidad de escoger a cual nodo vecino j moverse, con base en la probabilidad de salto establecida en el algoritmo ACLR, la cual está definida como sigue:

$$P_{ij}^{k}(t) = \frac{[\psi_{ij}(t)]^{\alpha} \times [\xi_{ij}]^{\beta} \times [\eta_{ij}(t)]^{\gamma}}{\sum_{s_l \in C(s_i)} [\psi_{il}(t)]^{\alpha} \times [\xi_{ij}]^{\beta} \times [\eta_{il}(t)]^{\gamma}} \tag{4.16}$$

Donde $\psi_{ij}(t)$ es el nivel de feromona en el enlace entre el nodo i y el nodo j, la cual está definida por:

$$\psi_{ij}(t+1) = (1-\rho) \cdot \psi_{ij}(t) + \left[\frac{\sum \Delta\psi_{ij}^{k}(t)}{\varphi B d_k} \right] \tag{4.17}$$

Donde ρ es un coeficiente tal que $(1-\rho)$ representa la evaporación de la feromona desde la última vez que ψ_{ij} fue actualizado. φ es un coeficiente y Bd_k es la distancia viajada (el número visitado de nodos) por la hormiga que viene k hasta el nodo j. Estos dos parámetros forzarán a la hormiga a perder parte de la fuerza de la feromona durante su camino al nodo origen. La idea detrás de este comportamiento es construir una mejor distribución de la feromona (nodos cerca del nodo destino tendrán mayores niveles de feromona) y forzarán a nodos lejanos a encontrar un mejor camino.

$\Delta\psi_{ij}^{k}$ es la feromona que la hormiga k deposita en la ruta entre s_i y s_j en la ronda actual. $\Delta\psi_{ij}^{k}$ se definirá en la segunda fase.

Además la información de ubicación de los nodos tiene una influencia significativa en las probabilidades con las que la hormiga k selecciona al nodo vecino s_j como el siguiente nodo a saltar. Por lo anterior se define la función de ubicación ξ_{ij} como sigue:

$$\xi_{ij} = \left(\frac{d_{ob}}{d_{oi} + d_{ij} + d_{jb}}\right) \cdot \left(1 - \frac{d_{ij}}{\sum_{s_l \in C(s_i)} d_{il}}\right) \tag{4.18}$$

Donde $0 \leq \xi_{ij} \leq 1$. Mientras más grande el valor de ξ_{ij} es más grande es la probabilidad con la que las hormigas seleccionan a s_j como el siguiente nodo a saltar.

Además, la energía residual de los nodos tiene una gran influencia sobre la probabilidad con la cual la hormiga k selecciona a s_j como el siguiente nodo a saltar. Por lo tanto, la función de energía $\eta_{ij}(t)$ se define a continuación:

$$\eta_{ij}(t) = \frac{e_j(t)}{\sum_{s_l \in C(s_i)} e_l(t)} \tag{4.19}$$

Donde $e_l(t)$ es la energía residual de s_l en el instante t. Entre más grande es el valor de $\eta_{ij}(t)$ más grande es la probabilidad con la cual la hormiga k selecciona al nodo s_j como el siguiente nodo a saltar.

α, β y γ son los pesos ajustables de la importancia de $\psi_{ij}(t)$, ξ_{ij} y $\eta_{ij}(t)$, respectivamente.

Cuando la hormiga k alcanza s_b, la hormiga k ha encontrado una ruta R^k desde s_o hasta s_b.

Para implementar estas ideas la memoria M_k de cada hormiga es reducida a sólo dos registros de los últimos dos nodos visitados. Puesto que el camino seguido por las hormigas no está en su memoria, una memoria debe ser creada para cada nodo donde se guarden los registros de cada hormiga que fue recibida y enviada. Cada registro de memoria guarda tanto el nodo previo como el nodo siguiente, la identificación de la hormiga y el temporizador del registro de la misma forma que lo establece el algoritmo EEABR.

Las hormigas que van sólo llevan el promedio de energía hasta el nodo actual ($Eavg_k$) y el mínimo nivel de energía registrada ($Emin_k$). Estos valores son actualizados por cada nodo que recibe a las hormigas que van. Cuando la hormiga que va alcanza el nodo destino se transforma en una hormiga que viene, cuya misión ahora es actualizar la feromona en la ruta que utilizó para llegar al nodo destino con ayuda del control de tránsito de hormigas implementado por los nodos sensores.

Cuando la hormiga que va alcanza el nodo destino, estos valores son utilizados para calcular la cantidad de feromona usada por la correspondiente hormiga que viene:

$$\Delta\psi_{ij}^k = \frac{1}{E_{init} - \left[\frac{Emin_k - Fd_k}{Eavg_k - Fd_k}\right]} \tag{4.20}$$

Donde Fd_k es la distancia viajada por la hormiga que va y E_{init} es el nivel de energía inicial de los nodos. Al calcular $\Delta\psi_{ij}^k$ solamente en función del nivel de energía de la ruta, puede traer consigo rutas no optimizadas, por lo que un camino con quince nodos

puede tener el mismo promedio de energía que una ruta con sólo cinco nodos. Además, $\Delta\psi_{ij}^{k}$ debe ser calculada como una función de ambos parámetros: Los niveles de energía y la distancia del camino. Ésto puede lograrse con la ayuda del parámetro Fd_k en la ecuación 4.20, el cual representa el número de nodos que la hormiga ha visitado.

Cuando la hormiga que viene alcanza el nodo donde fue lanzada, su misión ha terminado y la hormiga es eliminada. El Algoritmo 4 muestra el algoritmo IACAR.

Algoritmo 4 Algoritmo IACAR.

```
1: mientras la condición de término de iteraciones no se alcance hacer
2:   si el nodo origen no tiene batería entonces
3:     termina
4:   fin si
5:   Inicializar hormigas en el nodo origen
6:   para k = 1 hasta n hacer
7:     mientras nodo actual ≠ nodo destino hacer
8:       Obtener vecinos elegibles
9:       si vecinos elegibles ≠ NULL entonces
10:        Saltar al vecino siguiente con base en la probabilidad de salto P_ij^k(t)
11:        El nodo se encarga de guardar la información de la hormiga que va
12:      fin si
13:    fin mientras
14:    Crear hormiga que viene e iniciar su viaje al nodo origen
15:    Calcular distancia recorrida
16:    Calcular promedio de retardo
17:  fin para
18:  Actualizar feromona (ψ)
19:  Obtener mejor solución
20:  Calcular resultados promedio (consumo de energía, retardo de envío de paquetes
     y eficiencia de energía)
21: fin mientras
```

4.4. Encaminamiento por inundación dirigida

Con el objetivo de tener una referencia con los métodos de encaminamiento de datos para RIS no basados en ACO, se ha seleccionado el algoritmo llamado sistema de encaminamiento por inundación dirigida (*Directed flood-routing framework*, DFRF) el cual se describe en [68].

Los algoritmos de encaminamiento por inundación son métodos probabilísticos que se encargan del encaminamiento de datos en las RIS, son particularmente resistentes a fallos de nodos y de comunicaciones puesto que los paquetes de información pueden llegar a la estación base a través de diferentes rutas.

Sin embargo, los algoritmos convencionales de encaminamiento, como el de inundación, desperdician energía y ancho de banda cuando envían, de forman adicional e

innecesaria, paquetes de datos de sensores que cubren en general las mismas áreas de interés. Los inconvenientes de la inundación incluyen la duplicidad de los mensajes enviados a la estación base. Esto ocurre cuando dos nodos monitorean la misma región y envían paquetes similares a los mismos vecinos, dando lugar a un consumo de grandes cantidades de energía sin considerar las limitaciones de la misma.

Este algoritmo consiste en inundar la red con un paquete de datos capturado del medio ambiente para que llegue a todos los nodos de la red. Esta estrategia se basa en que cada nodo envía el paquete a todos sus nodos vecinos para garantizar que llegue a la estación base, donde esa información pueda ser procesada, almacenada o consumida. Para evitar que los mensajes estén circulando indefinidamente, un nodo reenviará sólo los mensajes que no ha recibido previamente (requiere para ello la identificación única de los nodos y de los mensajes) o, también pueden incorporar un tiempo de expiración a los paquetes.

El algoritmo DFRF está construido bajo un enfoque de inundación que controla el encaminamiento de datos en todos los nodos de la red. Cada uno de los nodos trabaja bajo este enfoque y puede enviar y recibir paquetes de datos de acuerdo a una política de inundación (principalmente donde se reduce la redundancia y la dirección de la inundación se controla). Las politicas de inundación especifican la dirección de la inundación y el como los nodos intermedios debe transmitir los paquetes. Este algoritmo mantiene los paquetes de datos recibidos recientemente en una tabla junto a su prioridad de transmisión, luego se seleccionan los paquetes con mayor prioridad y se transmiten a los nodos vecinos del nodo actual hasta que el paquete llega a la estación base.

La política de inundación establece que cada nodo tiene un nivel, el cual se refiere a que tan cerca se encuentra de la estación base. Este nivel dirige la transmisión del paquete de datos y es utilizado para establecer que hacer con paquetes que arrivan al nodo actual. Con lo anterior cuando un nodo recibe un paquete de información, compara su propio nivel con el nivel del nodo que le envió el paquete y, si tiene un menor nivel (se encuentra más cerca de la estación) entonces recibe el paquete y lo transmite a sus nodos vecinos, de lo contrario no recibe el paquete de datos.

Con esta técnica esta claro que la estación base recibirá un número de copias del mismo paquete capturado del ambiente, por lo que la estación base debe llevar un control de paquetes recibidos y evitar aceptar paquetes que habían llegado con anterioridad.

Para poder comparar de forma justa este algoritmo de encaminamiento de datos para RIS con los otros algoritmos basados en ACO estudiados, se establece que una iteración del algoritmo se cumple cada que todo el conjunto de copias del paquete que generó un nodo origen llegan a la estación base.

Capítulo 5

Experimentos y resultados

5.1. Experimentos

Para poder comparar el algoritmo propuesto IACAR con los algoritmos ACLR, EEABR y DFRF se ha agregado un cuarto escenario en el que el número de iteraciones se establece lo suficientemente grande para permitir que nodos mueran al consumir el total de su energía y por lo tanto, paquetes de información no llegen hasta la estación base.

Con este escenario se debe observar que el indicador de consumo de energía no es directamente proporcional al indicador de latencia. En la Tabla 5.1 se observan las características del escenario cuatro.

Tabla 5.1: Escenario cuatro.

Características
Todos los nodos empiezan con el mismo nivel de energía, el nodo origen s_o cambia aleatoriamente en cada iteración y el nodo destino s_b está fijo. El número de iteraciones es suficientemente grande para permitir que nodos sensores mueran.

Los valores de los parámetros iniciales son los mismos para los tres algoritmos a comparar, indicados en la Sección 4.2.3.1.

5.2. Resultados

Los resultados obtenidos tanto para el consumo de energía, como para la eficiencia de energía y la latencia (para los 30 experimentos realizados) para cada uno de lo cuatro escenarios se presentan a continuación.

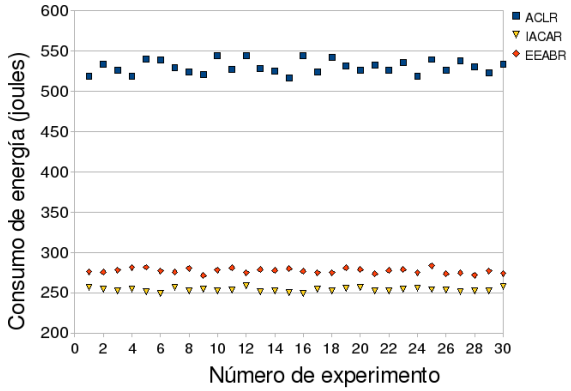

Figura 5.1: Primer escenario. Consumo de energía.

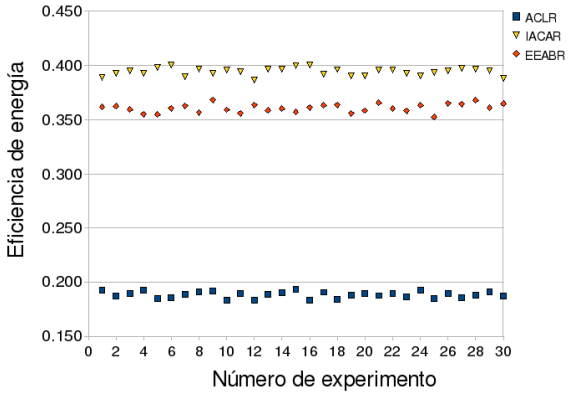

Figura 5.2: Primer escenario. Eficiencia de energía.

5.2.1. Primer escenario

En las Figuras 5.1, 5.2 y 5.3 se pueden observar los resultados obtenidos para los indicadores de consumo de energía, eficiencia de energía y latencia para los algoritmos comparados IACAR, ACLR y EEABR. Los resultados del algoritmo DFRF no se muestran en las Figuras, porque no permiten observar la verdadera diferencia entre los otros tres algoritmos comparados, puesto que los valores que obtuvo son en comparación, mucho mayores que los conseguidos por IACAR, ACLR y EEABR.

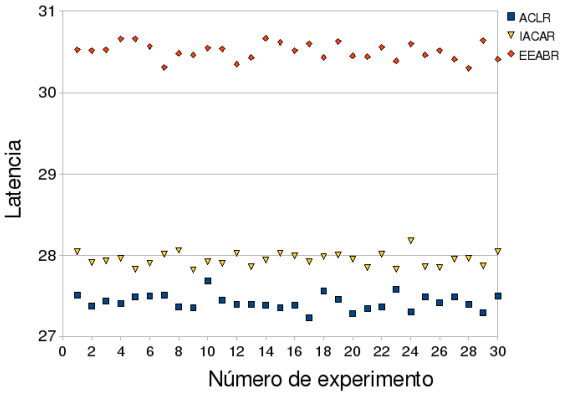

Figura 5.3: Primer escenario. Latencia.

En este primer escenario donde el nodo origen está fijo, el algoritmo IACAR en cuestión de consumo de energía estuvo mucho mejor que los algoritmos ACLR y DFRF, en todos los 30 experimentos, casi con los mismos valores que el algoritmo EEABR, que en este caso es el mejor en este indicador. Lo anterior señala que el tipo de administración que realizan los nodos respecto al flujo de hormigas propuesto por EEABR ayuda en gran medida al algoritmo IACAR.

En cuestión de latencia, IACAR es mejor que EEABR, en promedio dos o tres nodos menor por ruta, mientras que el algoritmo ACLR es el mejor, aunque con valores muy

cercanos a los obtenidos por la propuesta IACAR, con una diferencia de un nodo por ruta en promedio. DFRF es el peor en este rubro, encontrando en promedio rutas de arriba de 400 nodos.

Aunque se tomaron las partes principales de cada algoritmo comparado, IACAR no mejora el desempeño de sus contrapartes en los indicadores de consumo de energía y de latencia, sin embargo en el indicador de eficiencia de energía, el algoritmo propuesto IACAR tiene un mejor desempeño que ACLR e incluso que EEABR, con resultados casi del doble de diferencia con ACLR y un poco menos de un punto con su similar EEABR. Los resultados conseguidos por DFRF son mucho menores que los de IACAR, ACLR y EEABR, puesto que la redundancia de información en este algoritmo provoca una eficiencia de energía muy reducida.

Con lo anterior se establece que el algoritmo propuesto se acerca mucho al desempeño del ACLR al encontrar rutas cortas y, respecto al consumo de energía se acerca considerablemente al algoritmo EEABR, por lo que se establece que el algoritmo IACAR es el que mejor funciona para cuando se requiere que se maximice el tiempo de vida útil de la RIS.

5.2.2. Segundo escenario

Los resultados del segundo escenario tanto para el consumo de energía como la latencia y la eficiencia de energía están mostrados en las Figuras 5.4, 5.6 y 5.5 respectivamente para los algoritmos comparados IACAR, ACLR y EEABR. Los resultados que consiguió el algoritmo DFRF se encuentran en rangos mucho mayores que los otros tres algoritmos, por lo que no se presentan en las Figuras.

En este escenario el desempeño del algoritmo propuesto IACAR supera tanto al algoritmo ACLR como al EEABR y DFRF. En el consumo de energía, de los 30 experimentos IACAR mejora en todos al ACLR y al DFRF y en 24 al EEABR, recordando que el mejor en este indicador en el escenario pasado fue el EEABR. Lo anterior indica que el algoritmo propuesto funciona de una manera más destacada cuando la RIS no se encuentra en condiciones iniciales de funcionamiento (todos los nodos con baterías llenas). La inundación utilizada por DFRF provoca que los consumos de energía en los experimentos realizados sean muy grandes.

En el indicador de latencia IACAR encontró en promedio las menores rutas, puesto que de los 30 experimentos en 19 ha sido mejor que el ACLR. Con respecto al algoritmo EEABR IACAR tuvo mejor desempeño en la mayoría de los experimentos (con sólo cuatro donde fue mejor el EEABR). DFRF obtuvo los peores resultados, en promedio rutas de 276 nodos.

El indicador de eficiencia de energía es ganado por la propuesta en 24 de los 30 experimentos.

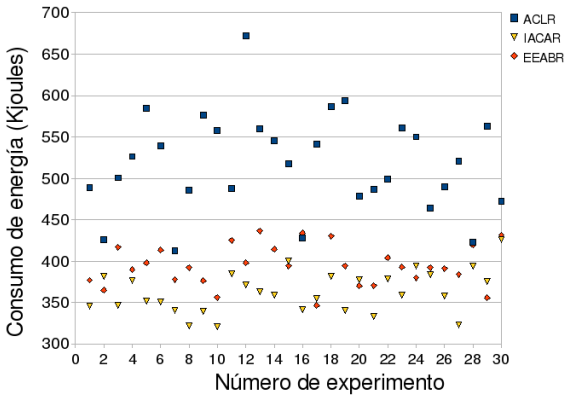

Figura 5.4: Segundo escenario. Consumo de energía.

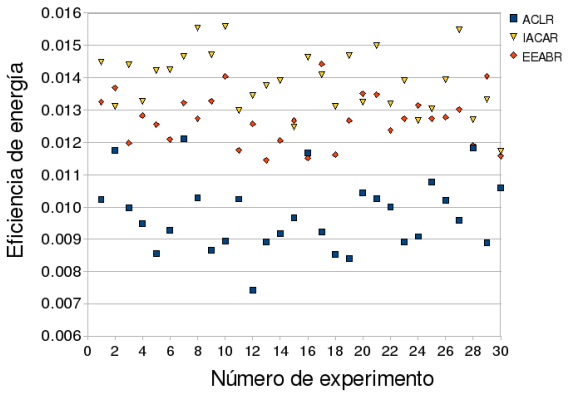

Figura 5.5: Segundo escenario. Eficiencia de energía.

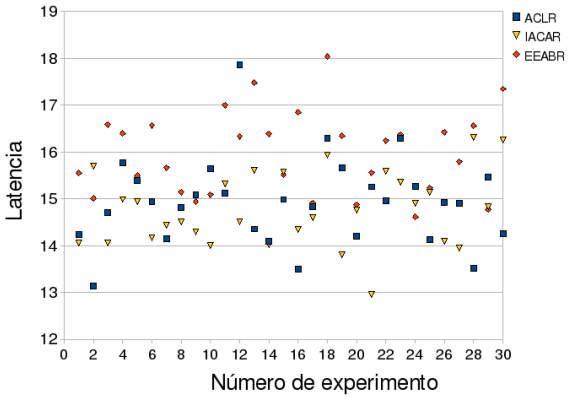

Figura 5.6: Segundo escenario. Latencia.

5.2.3. Tercer escenario

Los resultados obtenidos de los algoritmos comparados IACAR, ACLR y EEABR se muestran en las Figuras 5.7, 5.9 y 5.8, de los indicadores de consumo de energía, latencia y eficiencia de energía respectivamente. Al igual que en los escenarios anteriores, los resultados del algoritmo DFRF no se muestran el las Figuras por ser en proporción mucho mayores que los obtenidos por IACAR, ACLR y EEABR.

Para este escenario los resultados muestran un fenómeno interesante en el indicador de consumo de energía, pareciera que el algoritmo IACAR mejora con respecto al escenario anterior, pues en el escenario anterior sus resultados estaban más cercanos a los del algoritmo EEABR, sin embargo no es cierto, puesto que IACAR se ha mantenido en los mismos niveles obtenidos en el escenario anterior y, EEABR ha sido el que ha obtenido valores mayores a los que obtuvo en el segundo escenario. ACLR es el algoritmo que se desempeña de peor manera con casi los mismos valores que obtuvo en el escenario anterior. En este caso IACAR obtuvo mejores resultados en 24 de los 30 experimentos comparado con ACLR y EEABR. Al igual que en los escenarios anteriores el algoritmo DFRF consiguió los peores resultados.

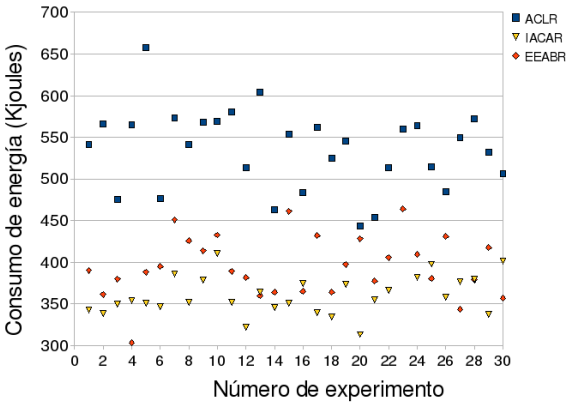

Figura 5.7: Tercer escenario. Consumo de energía.

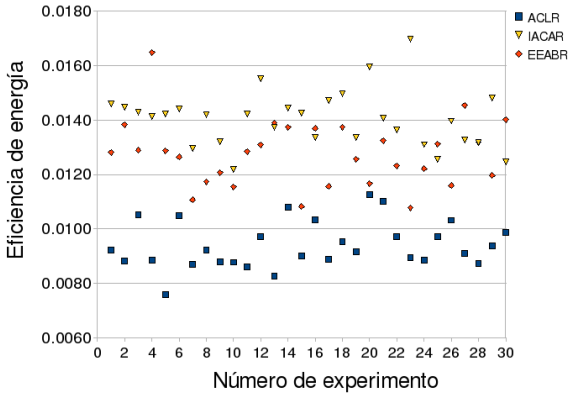

Figura 5.8: Tercer escenario. Eficiencia de energía.

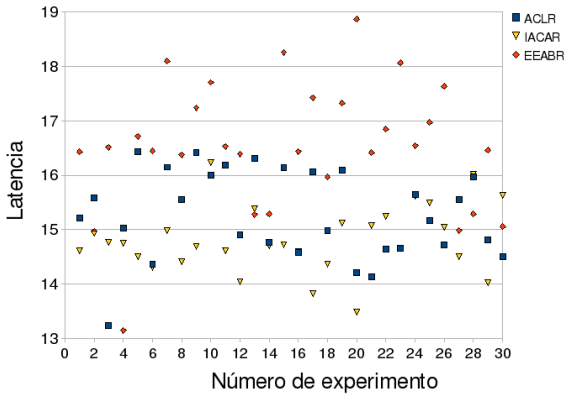

Figura 5.9: Tercer escenario. Latencia.

En el caso de la latencia se observa que el algoritmo propuesto se ha emparejado al ACLR, donde las rutas más pequeñas en promedio están muy parejas entre sí. En este rubro el algoritmo EEABR queda lejos del desempeño tanto de ACLR y de IACAR. En la eficiencia de energía, IACAR se mantiene como líder, incluso en algunos experimentos su desempeño supera por el doble al algoritmo ACLR.

5.2.4. Cuarto escenario

En las Figuras 5.10, 5.12 y 5.11 se pueden observar los resultados obtenidos para los indicadores de consumo de energía, latencia y eficiencia de energía respectivamente para el cuarto escenario. Los resultados conseguidos por el algoritmo DFRF no se muestran en las Figuras, por la misma razón que en los escenarios anteriores. DFRF obtuvo los peores resultados tanto para el consumo de energía, latencia y eficiencia de energía.

Debido a las características del escenario cuatro, el consumo de energía se disparó a cantidades mucho más altas que en los escenarios anteriores.

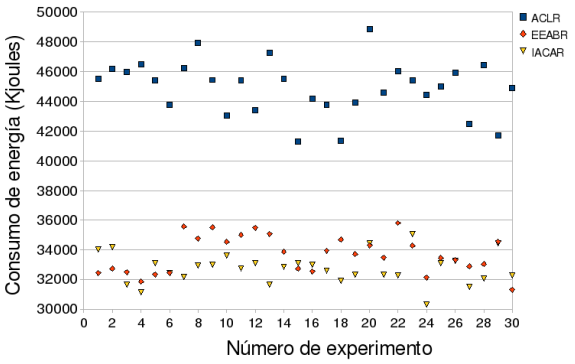

Figura 5.10: Consumo de energía en el cuarto escenario.

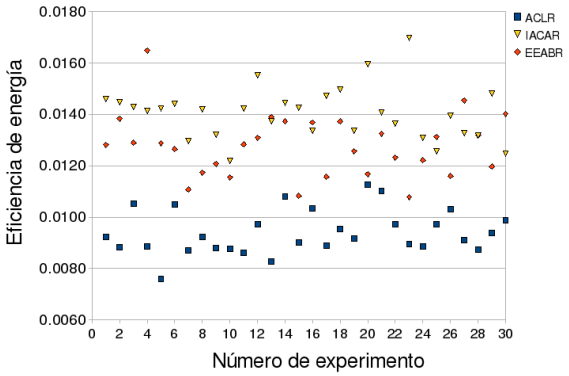

Figura 5.11: Eficiencia de energía en el cuarto escenario.

IACAR obtuvo los mejores resultados en 22 de los 30 experimentos realizados. Sin embargo lo importante en este escenario, es que no se observa una proporción directa entre los valores de los indicadores de consumo de energía y latencia, debido principalmente a que no todos los paquetes que se crearon en la RIS fueron recibidos por la estación base, puesto que varios nodos de la RIS murieron.

En cuestión de latencia, en este escenario se observa que ACLR es el que obtuvo peores resultados, mientras que IACAR mejoró de manera significativa al obtener mejores valores en 29 de los 30 experimentos. En el indicador de eficiencia de energía IACAR resultó mejor que ACLR y EEABR en 23 de los 30 experimentos realizados.

De igual manera se tiene que la correlación que se observaba entre el indicador de eficiencia de energía y el consumo de energía en los escenarios anteriores ya no aparece en éste, por la misma razón de que no todos los paquetes creados en la RIS llegan a su destino.

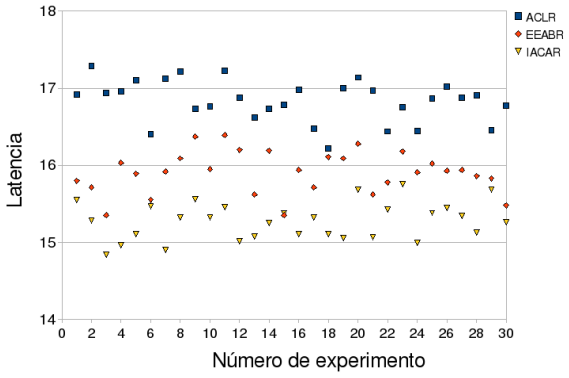

Figura 5.12: Latencia en el cuarto escenario.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Comentarios finales

Dirigimos esta tesis a la evaluación del desempeño de dos algoritmos de encaminamiento de datos para RIS basados en ACO. Dichos algoritmos son integrantes representativos en el área, tanto por su actualidad, como por los resultados reportados por sus autores. Las siglas de estos algoritmos son: ACLR y EEABR.

Estos algoritmos los programamos en lenguaje C con la finalidad de tener control de todos los parámetros utilizados en la simulación de los mismos. Los experimentos se realizaron bajo tres diferentes escenarios que representan tipos de RIS con distintas características de interés para el estudio que nos compete. Cada escenario refleja una situación muy particular de la RIS simulada, a grandes rasgos, diferentes cantidades de nodos desplegados en la RIS, iguales o diferentes niveles de energía inicial de los nodos sensores, posición del nodo origen fija o aleatoria, etcétera.

Para poder llevar a cabo la evaluación del desempeño de estos algoritmos de encaminamiento, hemos utilizado los indicadores clásicos del estado del arte, los cuales son: Consumo de energía, latencia y eficiencia de energía.

Cabe señalar que en los previos reportes de resultados de los algoritmos ACLR y EEABR se toman valores de consumo de energía equivocados, los cuales imposibilitan el efectuar una comparación justa y lo más real posible. Para evitar caer en el mismo error hemos investigado los valores reales, basados en experimentación, del consumo de energía para los nodos Mica2 de la compañía Crossbow.

Basados en los valores que obtuvimos de los experimentos, observamos que el algoritmo ACLR es mejor para encontrar rutas de menor tamaño, mientras que el algoritmo EEABR demostró más eficiencia en el uso de la energía de los nodos sensores en el proceso de la transmisión de datos en la RIS.

Hemos propuesto un algoritmo de encaminamiento basado en ACO para RIS llamado IACAR, que hemos diseñado basandonos en las principales características de los algoritmos ACLR y EEABR. El algoritmo IACAR toma como base, tanto la heurística

para la selección del siguiente nodo a saltar del algoritmo ACLR, como el tipo de administración de movimiento de paquetes dentro de la RIS propuesto por el algoritmo EEABR.

Además, con el objetivo de tener una referencia con los métodos de encaminamiento de datos para RIS no basados en ACO, se ha seleccionado el algoritmo DFRF para ser comparado con los algoritmos mencionados.

Con el objetivo de ejecutar la comparación entre los algoritmos IACAR, ACLR, EEABR y DFRF, hemos realizado una serie de experimentos con el uso de los tres distintos escenarios e indicadores ya mencionados, además de que hemos propuesto un cuarto escenario que trata de reflejar un comportamiento más acorde a la realidad de una RIS y, observamos que el algoritmo que hemos propuesto IACAR ha logrado conjuntar las partes fuertes de los algoritmos ACLR y EEABR de una manera correcta.

Por ejemplo en el indicador de consumo de energía IACAR obtuvo mejores valores en general que sus contrapartes en todos los escenarios establecidos, mientras que en el caso de la latencia demostró un desempeño muy cercano al del mejor de los otros dos algoritmos en este rubro (el algoritmo ACLR), al encontrar rutas en promedio 1 nodo más largas que las encontradas por éste, inclusive en el tercer y cuarto escenarios nuestra propuesta IACAR logró encontrar rutas más cortas en los experimentos que las obtenidas por el algoritmo ACLR. Los valores registrados en el indicador de eficiencia de energía muestran claramente que, el algoritmo que hemos propuesto supera en casi todos los experimentos de todos los escenarios a los otros tres algoritmos.

Con el objetivo de poder observar el comportamiento de la relación entre los indicadores utilizados, propusimos un cuarto escenario, donde basados en los resultados obtenidos podemos establecer que los valores del indicador de consumo de energía no son directamente proporcionales a los del indicador de latencia. Además también con ayuda del cuarto escenario mostramos que, los resultados del indicador de eficiencia de energía tampoco tiene una relación inversamente proporcional al los resultados del indicador de consumo de energía.

6.2. Conclusiones

Para poder realizar una comparación justa entre algoritmos de encaminamiento de datos para RIS, se debe tomar en cuenta con mucha atención los valores utilizados en la simulación para el consumo de energía.

El número de nodos desplegados no afecta de manera directa al desempeño del algoritmo ACLR en el caso de la obtención de rutas cortas, puesto que el promedio de nodos por ruta se mantiene en valores similares para cuando se despliegan 500, 2500, 5000 o 10000 nodos en la RIS.

Por su parte el algoritmo EEABR si mantiene una relación entre el promedio de nodos por ruta y el número de nodos desplegados en la RIS, por lo que es aconsejable desplegar el menor número de nodos en la RIS.

Mostramos que el algoritmo ACLR es el mejor en cuestión de búsqueda de rutas cortas en la RIS, puesto que obtuvo valores comparables con los obtenidos por el algoritmo que proponemos en los escenarios dos y tres.

Lo anterior implica que la mejor heurística del cálculo de la probabilidad de salto al siguiente nodo es la que propone el algoritmo ACLR, debido a que toma en cuenta tanto al nivel de energía de los nodos, los niveles de feromona de los enlaces como a la información de la ubicación de los nodos de la RIS.

El algoritmo IACAR es mejor en el caso del consumo de energía en general en todos los escenarios definidos, por lo tanto, IACAR es la mejor opción para el caso de aplicaciones que necesitan maximizar el tiempo de vida de la red.

El control del tránsito de paquetes propuesto por el algoritmo EEABR es el mejor respecto al consumo y eficiencia de energía de la RIS.

Los algoritmos IACAR y EEABR son más eficientes al maximizar el tiempo de vida de la RIS que los algoritmos ACLR y DFRF. Esta eficiencia se debe al uso de paquetes de tamaño reducido utilizados en su algoritmo.

Tanto nuestro algoritmo propuesto IACAR como el algoritmo ACLR, son los mejores algoritmos respecto a la latencia o retardo de la transmisión de información por la RIS.

Los resultados muestran que los algoritmos inspirados en ACO resuelven el problema del encaminamiento de datos para RIS de una manera eficiente, sin importar su naturaleza estocástica, por lo que este tipo de algoritmos son una opción viable en la solución de problemas de encaminamiento de datos en RIS.

6.3. Posibles aplicaciones

Este tipo de tecnología puede ser utilizado en nuestro país, por ejemplo:

- En cuestiones de criadero de animales y siembra, tanto para mantener las condiciones ambientales, como los niveles de materias primas en los puntos óptimos.
- En el área de ahorro de energía, tanto a nivel local como a nivel industrial, al crear instalaciones inteligentes y procesos automatizados en áreas de monitoreo, trayendo consigo reducciones importantes de costos de producción.
- En el ámbito de seguridad, para el control de equipo de armas de fuego, de comunicación, de personal, para facilitar la estrategia y táctica contra el crimen organizado, tener un mejor control del sistema de semáforos de la ciudad y/o para ayudar a sistemas de detección de ruido.
- Ayudar a los encargados de protección civil en siniestros, como detectores de movimientos, de ruidos, de fuerzas de presión, de fluido, etcétera.
- Para el ámbito de almacenes, al facilitar el control de inventario y condiciones ambientales necesarias para los productos guardados.

- En cuestiones médicas, como soporte de monitoreo de pacientes, como parte integral del equipo quirúrgico, de control de condiciones ambientales, de detección de accidentes médicos y/o ayuda en detección de enfermos.

6.4. Trabajo futuro

- La implementación del algoritmo en una RIS física.
- Uso o comparación con otras técnicas de encaminamiento de datos en RIS.
- Estudio de valores de parámetros del IACAR.
- Análisis y ampliación de las capacidades de ACO.
- Comparación de algoritmos de encaminamiento no basados en ACO.

Referencias

[1] Al-Karaki J. N. y Kamal-Ahmed E. , Routing Techniques in Wireless Sensor Networks A Survey, *IEEE Wireless Communications*, vol. 11, pp. 6–28, Dec. 2004.

[2] Stutzle T. y Dorigo M. , A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms, *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 358–365, Aug. 2002.

[3] Huo H. , Gao D. , Niu Y. y Gao S. , Asdp: an action-based service discovery protocol using ant colony algorithm in wireless sensor networks, in *Proceedings of the 3rd international conference on Mobile ad-hoc and sensor networks*, MSN'07, (Berlin, Heidelberg), pp. 338–349, Springer-Verlag, 2007.

[4] Iyengar S. S., Wu H. C., Balakrishnan N., y Chang S. Y., Biologically inspired cooperative routing for wireless mobile sensor networks, in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pp. 29–37, 2007.

[5] Heimfarth T. y Janacik P., Experiments with Biologically-Inspired Methods for Service Assignment in Wireless Sensor Networks, in *Biologically-Inspired Collaborative Computing* (M. Hinchey, A. Pagnoni, F. Rammig, and H. Schmeck, eds.), vol. 268 of *IFIP International Federation for Information Processing*, pp. 71–84, Springer Boston, 2008.

[6] Caro G. D. y Dorigo M., Antnet: A mobile agents approach to adaptive routing, tech. rep., 1997.

[7] Stallings W., *Comunicaciones y Redes de Computadoras*. Pearson Prentice Hall, 7th ed., 2004.

[8] Gómez C. y Paradells J., Redes Ad-hoc: el próximo reto, *Buran No. 21, Rama de estudiantes del IEEE de Barcelona*, pp. 30–37, June 2004.

[9] Chong C. y Kumar S. P., Sensor Networks: Evolution, Opportunities and Challenges, in *Proceedings of the IEEE*, pp. 1247–1256, Aug. 2003.

[10] Camilo T., Carreto C., Silva J. y Boavida F., An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks, in *Ant Colony Optimization and Swarm Intelligence* (M. Dorigo, L. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, eds.), vol. 4150 of *Lecture Notes in Computer Science*, pp. 49–59, Springer Berlin / Heidelberg, 2006.

[11] Holger K. y Andreas W., A short survey of wireless sensor networks, Tech. Rep. TKN-03-018, Technical University Berlin Telecommunication Networks Group, Oct. 2003.

[12] Ramos H., Macía F. y Marcos D., *Redes Inalámbricas de Sensores Inteligentes. Aplicación a la Monitorización de Variables Fisiológicas*, pp. 141–150. Desarrollo de Grandes Aplicaciones de Red. III Jornadas, JDARE 2006. Alicante, España: Editorial Club Universitario, May 2006.

[13] Raghunathan V., Kansal A., Hsu J., Friedman J. y Srivastava M., Design Considerations for Solar Energy Harvesting Wireless Embedded Systems, in *IPSN 05: Proceedings of the 4th international symposium on Information processing in sensor networks*, pp. 457–462, IEEE Press, 2005.

[14] Aakvaag N. y Frey J., Redes de sensores inalámbricos Nuevas soluciones de interconexión para la automatización industrial, *Revista técnica del Grupo ABB: Tecnologías spirits de sistemas integrados*, vol. 2, pp. 39–42, 2006.

[15] Serna J., Redes de sensores inalámbricas, tech. rep., Universidad de Valencia, Jan. 2007.

[16] Romer K. y Mattern F., The Design Space of Wireless Sensor Networks, *IEEE Wireless Comunications*, vol. 11, pp. 54–61, Dec. 2004.

[17] García C., González P., García J. y Pérez J., Wireless Sensor Networks and Applications: a Survey, *International Journal of Computer Science and Network Security*, vol. 7, No. 3, pp. 264–273, Mar. 2007.

[18] Agostino E., Giandomenico E., Corti R. y Martinez R., Ruteo en redes inalámbricas de sensores inteligentes, Tech. Rep. RT-ID-06/03, Facultad de Ciencias Exactas, Ingeniería y Agrimensura Universidad Nacional de Rosario, Dec. 2006.

[19] Chio C., Tibaduiza B., Aparicio A. y Ortiz C., Redes de Sensores Inalámbricos, *Congreso Internacional de Ingeniería Mecatrónica, Universidad Autónoma de Bucaramanga*, vol. 1, no. 1, 2009.

[20] Bulusu N., Heidemann J. y Estrin D., Gps-less low cost outdoor localization for very small devices, *IEEE Personal Communications Magazine*, vol. 7, pp. 28–34, 2000.

[21] Savvides A., Han C. y Srivastava M., Dynamic fine-grained localization in ad-hoc networks of sensors, in *Proceedings of the seventh annual international conference on Mobile computing and networking, Mobicom 2001*, pp. 166–179, 2001.

[22] Capkun S., Hamdi M. y Hubaux J., Gps-free positioning in mobile ad-hoc networks, in *Cluster Computing*, pp. 3481–3490, 2001.

[23] Chen B., Jamieson K., Balakrishnan H. y Morris R., Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, *Wireless Networks*, vol. 8, pp. 481–494, September 2002.

[24] Xu Y., Heidemann J. y Estrin D., Geography-Informed Energy Conservation for Ad hoc Routing, in *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, (New York, NY, USA), pp. 70–84, ACM, July 2001.

[25] Heinzelman W., Kulik J. y Balakrishnan H., Adaptive protocols for information dissemination in wireless sensor networks, in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '99, (New York, NY, USA), pp. 174–185, ACM, Aug. 1999.

[26] Kulik J., Heinzelman R. y Balakrishnan H., Negotiation-based protocols for disseminating information in wireless sensor networks, *Wireless Networks*, vol. 8, pp. 169–185, March 2002.

[27] Intanagonwiwat C., Govindan R. y Estrin D., Directed diffusion: a scalable and robust communication paradigm for sensor networks, in *MOBICOM*, pp. 56–67, ACM, 2000.

[28] Braginsky D. y Estrin D., Rumor routing algorithm for sensor networks, in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, (New York, NY, USA), pp. 22–31, ACM, Oct. 2002.

[29] Ye F., Chen A., Lu S., Zhang L. y Chen F. Y. A., A scalable solution to minimum cost forwarding in large sensor networks, pp. 304–309, Computer Communications and Networks, Oct. 2001.

[30] Chu M., Haussecker H., Zhao F., Chu M., Haussecker H. y Zhao F., Scalable information driven sensor querying and routing for ad hoc heterogeneous sensor networks, *International Journal of High Performance Computing Applications*, vol. 16, Aug. 2002.

[31] Shah C. y Rabaey J., Energy aware routing for low energy ad hoc sensor networks, *Wireless Communications and Networking Conference. WCNC2002*, vol. 1, pp. 350–355, Mar. 2002.

[32] Heinzelman W. y Balakrishnan H., Energy-Efficient Communication Protocol for Wireless Microsensor Networks, in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 8 of *HICSS '00*, (Washington, DC, USA), pp. 8020–8030, IEEE Computer Society, 2000.

[33] Manjeshwar A. y Agarwal D., Teen: a routing protocol for enhanced efficiency in wireless sensor networks, in *Proceedings 15th International In Proceedings of the 15th International Parallel and Distributed Processing Symposium*, vol. 3, pp. 2009–2015, Parallel and Distributed Processing Symposium, Apr. 2001.

[34] Manjeshwar A. y Agrawal D. P., Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, IPDPS '02, (Washington, DC, USA), pp. 48–56, IEEE Computer Society, 2002.

[35] Lindsey S., Raghavendra C. y Sivalingam K. M., Data Gathering Algorithms in Sensor Networks Using Energy Metrics, *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, pp. 1125–1130, September 2002.

[36] Rodoplu V. y Meng T. H., Minimum Energy Mobile Wireless Networks, *IEEE Journal on selected areas in communications*, vol. 17, pp. 1333–1344, Aug. 1999.

[37] Subramanian L. y Katz R., An architecture for building self configurable systems, in *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '00, (Piscataway, NJ, USA), pp. 63–73, IEEE Press, Aug. 2000.

[38] Li Q., Aslam J. y Rus D., Hierarchical power-aware routing in sensor networks, in *In Proceedings of the DIMACS Workshop on Pervasive Networking*, May 2001.

[39] Yu Y., Estrin D. y Govindan R., Geographical and energy-aware routing: A recursive data dissemination protocol for wireless sensor networks, *Technical report, UCLA Comp. Sci. Dept.*, May 2001.

[40] Kuhn F., Wattenhofer R. y Zollinger A., Worst-case optimal and average-case efficient geometric ad hoc routing, in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, (New York, NY, USA), pp. 267–278, ACM, June 2003.

[41] Martí R., Procedimientos Metaheurísticos en Optimización Combinatoria, *Matemátiques, Universidad de Valencia*, vol. 1, no. 1, pp. 3–62, 2003.

[42] Hoffman K., Combinatorial Optimization: Current Successes and Directions for the Future, *Journal of Computational and Applied Mathematics*, vol. 124, pp. 341–360, December 2000.

[43] Qishi W., Nageswara S., Barhen J., Iyengar S., Vijay K., Hairong Q. y Krishnendu C., On Computing Mobile Agent Routes for Data Fusion in Distributed Sensor Networks, *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 740–753, June 2004.

[44] Colorni A., Dorigo M., Maffioli F., Maniezzo V., Righini G. y Trubian M., Heuristics From nature for hard combinatorial optimization problems, *International Transactions in Operational Research*, vol. 3, no. 1, pp. 1–21, 1996.

[45] Dorigo M., Birattari M. y Stutzle T., Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique, *IEEE Comput. Intell. Mag*, vol. 1, pp. 28–39, Sept. 2006.

[46] Deneubourg J. y Pasteels J., Probabilistic Behaviour in Ants: A Strategy of Errors?, *Journal of Theoretical Biology*, vol. 105, pp. 259–271, Oct. 1983.

[47] Hopkin M., Las hormigas tienen un sistema propio para regular el tráfico, *Nature News*, vol. 10, pp. 1–6, Mar. 2004.

[48] Dorigo M. y Gambardella L. M., Ant Colony System: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[49] Alonso S., Cordón O., Fernández I. y Herrera F., La Metaheurística de Optimización Basada en Colonias de Hormigas: Modelos y Nuevos Enfoques, *Mejora de Metaheurísticas mediante Hibridación y sus Aplicaciones, Universidad de Granada España*, pp. 261–314, 2004.

[50] Cruz H. y Viteri V., Optimización de problemas combinatorios y multiobjetivo utilizando el método de colonia de hormigas (OCH), Master's thesis, Escuela Politécnica Nacional - Quito, Apr. 2007.

[51] Kylie B., Genetic Algorithms and the Traveling Salesman Problem, tech. rep., Department of Mathematics, Harvey Mudd College, Dec. 2000.

[52] Dorigo M., Maniezzo V. y Colorni A., The Ant System: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, vol. 26, no. 1, pp. 1–13, 1996.

[53] Dorigo M. y DiCaro G., Ant Colony Optimization: A New Meta-Heuristic, *In Proceedings of the Congress on Evolutionary Computation*, vol. 2, pp. 1470–1477, Sept. 1999.

[54] Gómez O., Razones del Exito de Optimización por Colonia de Hormigas en el Problema del Cajero Viajante, Master's thesis, Universidad Nacional de Asunción, Sept. 2004.

[55] Stutzle T. y Hoos H., MAX–MIN Ant System, *IRIDIA, Future Generation Computer Systems*, vol. 16, pp. 889–914, 2000.

[56] Gómez O. y Barán B., Omicron ACO. A New Ant Colony Optimization Algorithm, *CLEI Electronic Journal*, vol. 8, Aug. 2005.

[57] Farooq M. y Caro G. A., Routing Protocols for Next-Generation Networks Inspired by Collective Behaviors of Insect Societies An Overview, *Swarm Intelligence*, pp. 101–160, 2008.

[58] Cai W., Jin X., Zhang Y., Chen K. y Wang R., ACO Based QoS Routing Algorithm for Wireless Sensor Networks, *Springer-Verlag, LNCS*, vol. 4159, pp. 419–428, 2006.

[59] Reza G., Rahman A., Gueaieb W. y Saddik A., Ant Colony-Based Reinforcement Learning Algorithm for Routing in Wireless Sensor Networks, *Instrumentation and Measurement Technology Conference Proceedings, 2007 IEEE*, pp. 1–6, 2007.

[60] Wen Y., Chen Y. y Qian D., An Ant-based approach to Power-Efficient Algorithm for Wireless Sensor Networks, *World Congress on Engineering*, vol. 2, pp. 1546–1550, July 2007.

[61] Ye N., Shao J., Wang R. y Wang Z., Colony Algorithm for Wireless Sensor Networks Adaptive Data Aggregation Routing Schema, in *Proceedings of the Life system modeling and simulation 2007 international conference on Bio-Inspired computational intelligence and applications*, LSMS'07, (Berlin, Heidelberg), pp. 248–257, Springer-Verlag, 2007.

[62] Saleem K., Fisal N., Baharudin M., Ahmed A., Hafizah S. y Kamilah S., Ant Colony inspired Self-Optimized Routing Protocol based on Cross Layer Architecture for Wireless Sensor Networks, *WSEAS Transactions on Communications*, vol. 9, pp. 669–678, Oct. 2010.

[63] Yang J., Xu M., Zhao W. y Xu B., A Multipath Routing Protocol Based on Clustering and ACO for WSN, *Sensors 2010*, vol. 10, pp. 4521–4540, May 2010.

[64] Camara D. y Loureiro A., A gps/ant-like routing algorithm for ad hoc networks, *IEEE Wireless Commun. Netw. Conf.*, pp. 1232–1236, 2000.

[65] Okdem S. y Karaboga D., Routing in Wireless Sensor Networks Using an Ant Colony Optimization (ACO) Router Chip, *Sensors*, vol. 9, pp. 909–921, Feb. 2009.

[66] Wang X., Li Q., Xiong N. y Pan Y., Ant Colony Optimization-Based Location-Aware Routing for Wireless Sensor Networks, *Springer-Verlag, LNCS*, vol. 5258, pp. 109–120, 2008.

[67] Torres M. G., Energy Consumption in Wireless Sensor Networks Usig GSP, Master's thesis, Universidad Pontificia Bolivariana, Medellín, Colombia, 2006.

[68] Marote M., Directed Flood-Routing Framework for Wireless Sensor Networks, *In Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*, no. 5, pp. 99–114, 2004.