



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO

MAESTRÍA EN TELECOMUNICACIONES

TEMA:

**ANÁLISIS Y EVALUACIÓN DE UNA PLATAFORMA DE MONITOREO
Y CONTROL UTILIZANDO ALGORITMOS DE ENRUTAMIENTO
PARA OBTENER EL MENOR COSTO ENERGÉTICO EN REDES
ZIGBEE**

AUTOR:

Esparza Abad, Julio Armando

**Trabajo de titulación previo a la obtención del grado de
Magister en Telecomunicaciones**

TUTOR:

MSc. Manuel Romero Paz

Guayaquil, a los 24 días del mes Agosto del año 2017



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO
MAESTRÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por Esparza Abad, Julio Armando como requerimiento parcial para la obtención del Título de Magíster en Telecomunicaciones.

TUTOR

MSc. Manuel Romero Paz

DIRECTOR DEL PROGRAMA

MSc. Manuel Romero Paz

Guayaquil, a los 24 días del mes Agosto del año 2017



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO

MAESTRÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, Esparza Abad, Julio Armando

DECLARÓ QUE:

El trabajo de Titulación “**Análisis y evaluación de una plataforma de monitoreo y control utilizando algoritmos de enrutamiento para obtener el menor costo energético en redes Zigbee**”, previa a la obtención del Título de **Magíster en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 24 del mes Agosto del año 2017

EL AUTOR

Esparza Abad, Julio Armando



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO
MAESTRÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Esparza Abad, Julio Armando

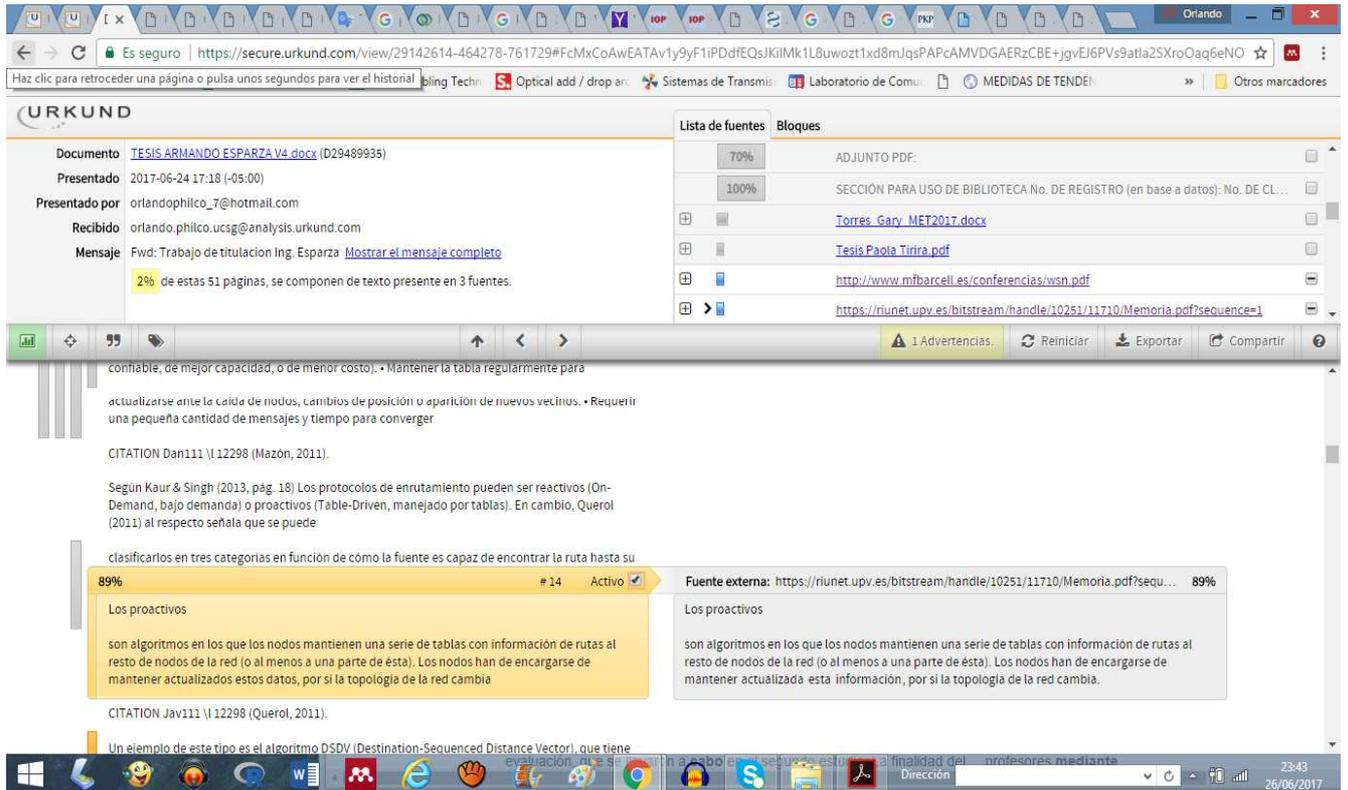
Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación**, en la biblioteca de la institución del Trabajo de Titulación, “**Análisis y evaluación de una plataforma de monitoreo y control utilizando algoritmos de enrutamiento para obtener el menor costo energético en redes Zigbee**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 24 del mes Agosto del año 2017

EL AUTOR

Esparza Abad, Julio Armando

REPORTE URKUND



Reporte Urkund trabajo de titulación; “Análisis y evaluación de una plataforma de monitoreo y control utilizando algoritmos de enrutamiento para obtener el menor costo energético en redes zigbee” del ing. Armando Esparza al 2% de coincidencias.

Atentamente.

MSc Orlando Philco Asqui

DEDICATORIA

Dedico esta tesis a Dios, a mi hermosa familia que he conformado con mi Esposa y mi hija y a mis padres y hermanos porque son los que han estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar con mi formación profesional, los mismos que a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ellos que soy lo que soy ahora.

Esparza Abad, Julio Armando

AGRADECIMIENTO

Este proyecto es el resultado de un gran esfuerzo profesional investigativo, por esto agradezco a mi Tutor MSc. Manuel Romero Paz, a mi compañera de aula la Ing. Natalia Romero. A mi familia, padres y hermanos quienes a lo largo de toda mi vida han apoyado y motivado mi formación académica, creyeron en mí en todo momento y no dudaron de mis habilidades. A mis profesores a quienes les debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza y finalmente un eterno agradecimiento a esta prestigiosa universidad, preparándonos para un futuro competitivo y formándonos como personas de bien.

Esparza Abad, Julio Armando



UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL
SISTEMA DE POSGRADO
MAESTRÍA EN TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

f. _____

MSc. Manuel Romero Paz
TUTOR

f. _____

MSc. Orlando Philco Asqui
REVISOR

f. _____

MSc. Luis Córdova Rivadeneira
REVISOR

ÍNDICE GENERAL

ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS.....	XIII
Resumen.....	XV
Abstract	XVI
INTRODUCCIÓN	1
CAPÍTULO I: GENERALIDADES DE LA INVESTIGACIÓN	2
1.1 Problema a resolver.....	2
1.2 Objeto de estudio	2
1.3 Objetivo general.....	2
1.4 Objetivos específicos	2
1.5 Justificación	2
1.6 Hipótesis	3
1.7 Metodología de la investigación	3
CAPÍTULO II: REDES ZIGBEE Y PROTOCOLOS DE ENRUTAMIENTO	4
2.1. Estructura del Stack o Pila de Zigbee	6
2.2 Operación de topologías en Zigbee.....	7
2.3 Red de Sensores Inalámbricos (WSN).....	10
2.3.1 Aplicaciones de WSN y Zigbee	11
2.4 Protocolos de enrutamiento para redes de sensores inalámbricos.....	12
2.5 Particularidades de la clasificación de protocolos de enrutamiento.....	13
2.5.1 Consumo de energía.....	16
2.6 Protocolos de enrutamiento tradicionales	17
2.6.1 Protocolo AODV	17
2.6.2 Protocolo SPIN (Sensor Protocol for Information Via Negotiation)	23
2.6.3 Protocolo LEACH (Low Energy Adaptive Clustering Hierarchy)	25
2.7 Protocolos de enrutamiento basados en Inteligencia Computacional	27

2.7.1 Algoritmos basados en Redes Neuronales	28
2.7.2 Algoritmos basados en Lógica Difusa	29
2.7.3 Algoritmos Evolutivos (AE)	30
2.7.4 Algoritmos basados en Inteligencia de Enjambres	32
2.7.5 Algoritmos basados en Aprendizaje por Refuerzos	33
2.7.6 FROMS (Feedback ROuting to Multiple Sinks).....	35
2.7.7 Comparación entre las técnicas de IC	36
2.8 Elección del Simulador	37
2.8.1 Simuladores de propósito general	37
2.8.2 Simuladores para plataformas específicas	38
CAPÍTULO III: DESEMPEÑO Y PRUEBAS DEL PROTOCOLO FROMS.....	40
3.1 Desempeño del algoritmo FROMS.....	40
3.1.1 Enrutamiento multicast con Q-Learning	41
3.1.2 Valores Q	42
3.1.3 Actualización de valores Q	43
3.1.4 Enrutamiento consciente de la energía con FROMS	44
3.1.5 Manejo de fallas	45
3.2.1 Estimación de la calidad del enlace	47
3.2.2 Selección de rutas y balanceo de cargas	48
3.3 NS-3	51
3.3.1 Estructura básica	51
3.3.3 Programación de módulos.....	52
3.4 Programación del algoritmo FROMS	53
3.4.1 Implementación del módulo de enrutamiento.....	53
3.4.2 Clase Tabla Enrutamiento.....	55
3.4.3 Formatos de paquetes.....	57
3.4.4 Implementación del comportamiento del módulo.....	58

3.5 Modificaciones en el simulador	63
3.6 Simulaciones de protocolos de enrutamiento.....	64
3.6.1 Configuración de los parámetros del simulador.....	65
3.6.2 Topología de la red	65
3.7 Escenarios de simulación	69
3.7.1 Comparación de las funciones de costo	70
3.7.2 Selección de la velocidad de aprendizaje.....	72
3.7.3 Selección de la tasa de exploración.....	74
3.8 Discusión de Resultados	78
CONCLUSIONES	80
RECOMENDACIONES.....	82
BIBLIOGRAFÍA	83
GLOSARIO DE TÉRMINOS.....	89
ANEXO 191	

ÍNDICE DE TABLAS

Tabla 2. 1 Valores de alcances (metros) según su velocidad de transmisión en red Zigbee.....	5
Tabla 2. 2 Ventajas y desventajas de protocolos de enrutamiento reactivos y reactivos	14
Tabla 2. 3 Comparación de los modelos de enrutamiento salto a salto y fijado en origen.....	15
Tabla 2. 4 Clasificación de protocolos de enrutamiento	15
Tabla 2. 5 Representación del formato del paquete solicitud de ruta	18
Tabla 2. 6 Formato del paquete de respuesta de ruta	19
Tabla 2. 7 Requerimientos de los diferentes paradigmas de IC.....	36
Tabla 3. 1 Estructura Sink Control para el nodo E	46
Tabla 3. 2. Módulos utilizados en la definición de los nodos	66
Tabla 3. 3. Niveles de potencia de salida del transceptor CC2420.....	67
Tabla 3. 4. Parámetros generales de simulación	68
Tabla 3. 5. Escenarios de simulación.....	69
Tabla 3. 6. Métricas de evaluación.....	69
Tabla 3. 7. Desviación estándar de la energía residual para cada función de costo.	72
Tabla 3. 8. Resultados obtenidos por cada protocolo.	77

ÍNDICE DE FIGURAS

Figura 2. 1. Ubicación de Zigbee según tipos de redes inalámbricas	5
Figura 2. 2 Estructura de pila de Zigbee	6
Figura 2. 3 Topología de Zigbee (a) Estrella, (b) Mesh y (c) Árbol.....	7
Figura 2. 4 Topología de árbol de clúster y tipo mesh o malla.....	9
Figura 2. 5, Ejemplo de redes Zigbee para monitoreo de calidad y nivel de agua	11
Figura 2. 6 Creación de rutas con AODV. Fuente: (Roselló, 2009).....	21
Figura 2. 7 (a) Desempeño Protocolo AODV (b) Desempeño Protocolo DYMO	22
Figura 2. 8 (a) Negociación SPIN pasos 1 y 2; (b) Negociación SPIN pasos 3 y 4 Fuente: (Garbarino, 2011).....	24
Figura 2. 9 Negociación SPIN pasos 5 y 6 Fuente: (Garbarino, 2011).....	24
Figura 2. 10 Esquema de red con protocolo LEACH	26
Figura 2. 11 Red con proceso de envío de un paquete en Q-Routing.....	34
Figura 3. 1 Topología de ejemplo con dos sumideros (sink) y una fuente	40
Figura 3. 2 Multiplicadores de estimados para diferentes funciones de costo.....	45
Figura 3. 3 Estimación de la calidad del enlace extremo a extremo	48
Figura 3. 4 Algoritmo de selección de rutas de REL	50
Figura 3. 5 Módulo compuesto para los nodos en NS-3	52
Figura 3. 6 Relación entre las clases programadas	54
Figura 3. 7 Estructura del módulo Froms Routing en NS-3.	54
Figura 3. 8 Estructura Entrada	56
Figura 3. 9 Estructura de la clase Tabla Enrutamiento	56
Figura 3. 10 Formato del paquete de anuncios del sumidero.....	57
Figura 3. 11 Formato del paquete de datos.	58
Figura 3. 12 Estructura de la clase Froms Routing	59

Figura 3. 13 Fragmento de código de ResourceManager.h.	61
Figura 3. 14. Topología de red de prueba.	65
Figura 3. 15. Tiempo de vida de la red para cada una de las funciones de costo .	71
Figura 3. 16. Tiempo de vida de la red para diferentes velocidades de aprendizaje	73
Figura 3. 17. Tasa de recepción de paquetes para diferentes velocidades de aprendizaje	73
Figura 3. 18. Tasa de recepción de paquetes para diferentes tasas de exploración.	75
Figura 3. 19. Paquetes recibidos en el sumidero por cada nodo en la red	76
Figura 3. 20 Histograma de la energía residual en baterías	78
Figura 3. 21 Esquema de aprendizaje de nodo en red Zigbee.....	79

Resumen

En el presente trabajo se caracteriza y evalúa el consumo de energía por comunicación en una red IEEE 802.15.4 conocida como Zigbee, dicha red opera en cortas distancias. Como problemática surge determinar o seleccionar el protocolo de enrutamiento eficiente para el cumplimiento del monitoreo y transmisión de datos entre diversos nodos que conforma la red Zigbee. La metodología es descriptiva-analítica, se emplea la metodología empírica por el uso de un programa simulador para valorar o cuantificar el protocolo de enrutamiento eficiente. Se comparan dos tipos de protocolos de enrutamiento, uno basado en Routing by Energy and Link Quality, REL y otro basado en Feedback ROuting to Multiple Sinks, FROMS, las evaluaciones indican que este último es más eficiente en un 17% en su tarea de censado, procesamiento y análisis de datos. Se concluye que los protocolos de ruteo basados en inteligencia computacional, puede permitir un desempeño o comportamiento inteligente a una red Zigbee o WSN.

Palabras Claves: AODV, REL, FROMS, Zigbee, WSN,

Abstract

In the present work, the energy consumption by communication in an IEEE 802.15.4 network known as Zigbee is characterized and evaluated, this network operates in short distances. As problematic it arises to determine or to select the efficient routing protocol for the fulfillment of the monitoring and transmission of data between diverse nodes that conforms the network Zigbee. The methodology is descriptive-analytical, empirical methodology is used by the use of a simulator program to evaluate or quantify the efficient routing protocol. We compare two types of routing protocols, one based on Routing by Energy and Link Quality, REL and another based on Feedback ROuting to Multiple Sinks, FROMS, evaluations indicate that the latter is more efficient at 17% in its census task, Data processing and analysis. It is concluded that routing protocols based on computational intelligence, can allow performance or intelligent behavior to a Zigbee or WSN network.

Keywords: AODV, REL, FROMS, Zigbee, WSN.

INTRODUCCIÓN

El estándar IEEE 802.15.4 de Zigbee proporciona la comunicación inalámbrica entre dispositivos de red de comunicaciones en área personal, la utilización de algoritmos para enrutamiento, y los servicios de apoyo para aplicaciones que operan en determinadas capas del modelo de referencia OSI (*Open Systems Interconnection*), establecen un enlace confiable, aunque por factores como distancia entre transmisor y receptor, frecuencias adyacentes iguales que perturban o ‘cortan’ la comunicación y otros más, son debilidades de las redes de área personal, que deben probarse o evaluarse a través de los diversos algoritmos de enrutamiento que utilizan la red Zigbee y comprobar una topología de red apropiada para la aplicación final deseada.

Una red basada en IEEE 802.15.4 especifica un conjunto de protocolos de comunicación de alto nivel que se utilizan para crear redes de área personal con bajo consumo de energía, Zigbee pretende ser más simple y menos costosa que otras redes inalámbricas de área personal (*Wireless Personal Area Network*, WPAN), tales como Bluetooth o Wifi. La relación entre IEEE 802.15.4 y Zigbee es similar a la existente entre IEEE 802.11 y la Alianza Wifi (Wang, Tao , & Qian , 2014, pág. 233).

Las aplicaciones incluyen la automatización del hogar, la recopilación de datos de dispositivos para telemedicina y monitoreo de la salud, entre otras necesidades de bajo ancho de banda y de baja potencia (Wilamowski & Irwin, 2011).

Zigbee es un estándar abierto, aunque la interoperabilidad es un factor que afecta un sistema de monitoreo y control diseñado bajo este estándar, es esencial evaluar protocolos confiables y eficientes en energía, es significativo para resistir el consumo energético por las diversas aplicaciones de WSN (*Wireless Sensor Network*); dependiendo de la aplicación una red puede tener decenas o cientos de nodos, y cada nodo hace uso de los protocolos para comunicarse con los otros nodos y con el nodo destino, o también llamado nodo *sink* (sumidero). El análisis de evaluaciones de los protocolos de ruteo, se utiliza para plantear una topología robusta para el consumo energético en este tipo de redes.

CAPÍTULO I: GENERALIDADES DE LA INVESTIGACIÓN

1.1 Problema a resolver

El patrón de tráfico alto y no uniforme presenta altas cargas de tráfico a los nodos de una red Zigbee y afecta su desempeño de transmisión y recepción para aplicaciones específicas de la red, esto provoca consumo energético no deseado y reduciría la capacidad de carga de batería en sus nodos.

1.2 Objeto de estudio

Protocolos de enrutamientos Zigbee

1.3 Objetivo general

Realizar el análisis de algoritmos de enrutamiento en una Zigbee de monitoreo y control para obtener el menor costo energético.

1.4 Objetivos específicos

1. Estudiar la operación de las redes Zigbee
2. Caracterizar los protocolos de enrutamiento para un entorno Zigbee
3. Determinar el desempeño de un protocolo de enrutamiento con menor consumo energético para una red Zigbee.
4. Analizar el desempeño de red Zigbee con protocolo FROMS para eficiencia en consumo de energía en red Zigbee.

1.5 Justificación

Las aplicaciones actuales de monitoreo y control utilizan decenas, centenas y miles de nodos sensores que se comunican con su nodo coordinador, este escenario aumenta el tráfico de datos y cuando el entorno dificulta la comunicación entre nodos, esto provoca alto consumo de energía de las baterías de los nodos. La

utilización de protocolos basado en inteligencia computacional permitirá que el flujo y tráfico de datos logre optimizar la transmisión de paquetes un aspecto técnico que confluye en una red Zigbee eficiente.

1.6 Hipótesis

El uso de algoritmos en protocolo de enrutamiento basado en inteligencia computacional puede mejorar el desempeño de comunicación en redes IEEE 802.15.4 estos algoritmos inducirán una tendencia de reducción del consumo energético de la batería de los nodos de la red.

1.7 Metodología de la investigación

La metodología a emplearse en este trabajo académico de posgrado para la obtención de la titulación en la Maestría de Telecomunicaciones, tiene como metodología principal, la descriptiva y analítica por cuanto se estudia y se analiza la literatura científica en redes del estándar IEEE 802.15.4 y su análisis de problemas en rendimiento de dichas redes en los últimos 4 años.

También se trata de una investigación de tipo Cuantitativa por cuanto se establecen resultados presentados en tablas y gráficos. En cuanto a metodología es Empírica-Analítica ya que se realizará un proceso de simulación de red Zigbee con el software NS-3, el manejo de este versátil software revelará la eficiencia de protocolos de enrutamiento y con aquello se plantea la propuesta de una topología robusta para redes bajo el estándar IEEE 802.15.4.

CAPÍTULO II: REDES ZIGBEE Y PROTOCOLOS DE ENRUTAMIENTO

En el estándar IEEE 802.15.4 se incluye a Zigbee, y ésta es una marca registrada de este grupo, al pertenecer al estándar forma también parte de la Zigbee Alliance, que publica estándares y normativas de aplicaciones que permiten a múltiples fabricantes de equipos crear productos interoperables. Los miembros de esta alianza justifican el desarrollo de este estándar para cubrir el vacío que se ocasiona por debajo de Bluetooth.

Se puede utilizar de forma similar a Bluetooth o Wifi para transferir datos, sin embargo, Zigbee se utiliza para enviar mucho menos datos a la vez. El objetivo principal de Zigbee es proporcionar una solución de bajo costo para redes WPAN. Está diseñado específicamente para redes WSN, para acoplar con microcontroladores de 8 bits, aunque se dispone de soluciones de 16 y 32 bits (González, 2011).

Zigbee logra transmitir con un simple protocolo de 20kbps hasta 250Kbps trabajando a una frecuencia de 2,4GHz, usa la modulación de espectro expandido (DSSS, *Direct Sequence Spread Spectrum*) bajo consumo y rangos de cobertura entre 10 y 75 metros, no obstante, dependerá de circunstancias físicas ambientales, que determinan las distancias de alcance (Mayné, 2005).

La tecnología Zigbee forma parte de la categoría de red inalámbrica, conocida como redes de área personal e incluye también parte de las redes de área local (LAN, *Local Area Network*.) esta tecnología de corto alcance o cobertura ha sido destinada para aplicaciones en el hogar como la seguridad y la automatización. (Moreno & Daniel, 2007).

En la figura 2.1 se muestra un esquema con la ubicación de la tecnología Zigbee entre los diversos tipos de redes con comunicación inalámbrica.

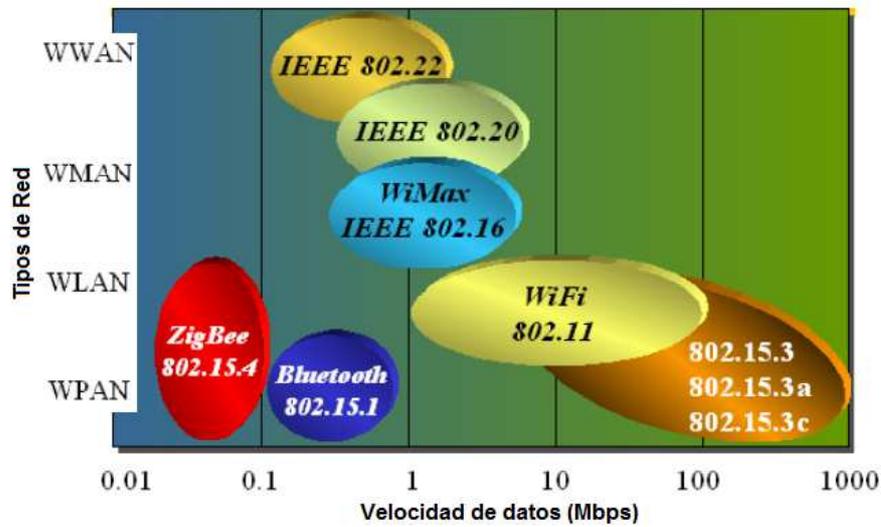


Figura 2. 1. Ubicación de Zigbee según tipos de redes inalámbricas

Fuente: (Moreno & Daniel, 2007)

Zigbee, además es conocida con otros nombres como "HomeRF Lite", que opera en bandas de frecuencias no licenciadas ISM (*Industrial, Scientific and Medical*) de 2,4 GHz, 868 MHz (Europa) y 915 MHz (EEUU).

En la tabla 2.1 se muestra rangos estimados de cobertura en metros, dependiendo de la velocidad de transmisión y su medida de potencia (expresada en dB, decibelios).

Tabla 2. 1 Valores de alcances (metros) según su velocidad de transmisión en red Zigbee

Velocidad de transmisión	0 dBm/1mW	10 dBm/10mW	20 dBm/100mW
250 kbps	13m	29m	67m
28 kbps	23m	54m	135m

Nota: 0 dBm es igual a 1 mW. Si se aumenta 3 dB representa más o menos el doble de potencia, es decir 3 dBm es casi igual a 2 mW.

Fuente: Elaborado por el autor

Se espera que los dispositivos inalámbricos compatibles con Zigbee transmitan entre 10 y 75 metros, dependiendo del ambiente RF (Radio Frecuencia) y el consumo de potencia requerido para una aplicación determinada, así la velocidad de transmisión de datos es 250kbps a 2.4GHz, 40kbps a 915MHz y 20kbps a 868MHz. (Morocho & Hipo, 2012).

Una red Zigbee puede estar formada por hasta 255 nodos, los cuales tienen el transceptor Zigbee en estado ‘dormido’ la mayor parte del tiempo, para mantener un bajo consumo. (Mayné, 2005, pág. 12).

2.1. Estructura del *Stack* o Pila de Zigbee

La pila es una lista estructurada de datos que permite almacenarlos y extraerlos, cuando se los requiera mediante una aplicación específica. La arquitectura del “*stack*” software de Zigbee es más simple que muchos “*stacks*” del protocolo así que requiere menos códigos de software (Mayné, 2005), (Murias, 2009).

En la figura 2.2, se muestra un diagrama o representación compacta de la pila de protocolo Zigbee completa, incluyendo capas inferiores 802.15.4 definidas por la IEEE.

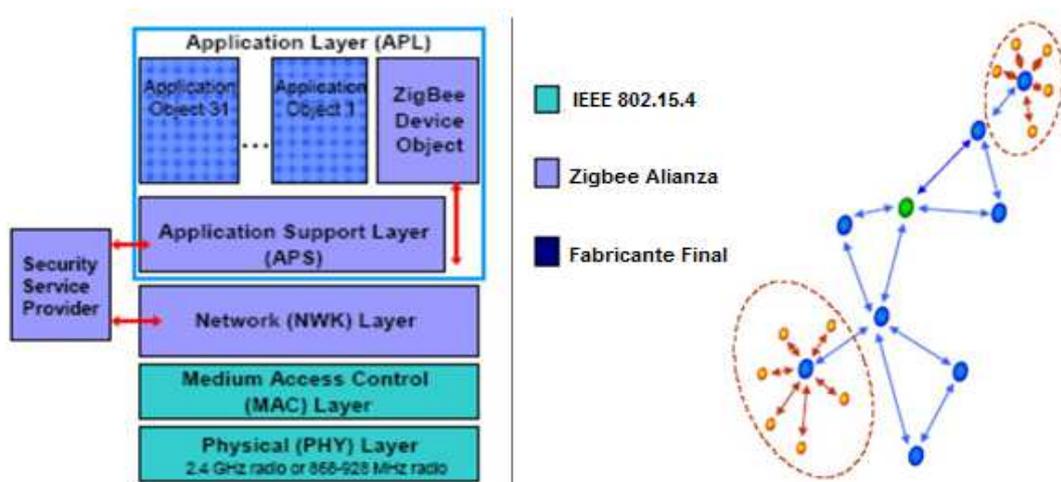


Figura 2. 2 Estructura de pila de Zigbee

Fuente: (Mayné, 2005)

La norma 802.15.4 define que la capa Física abreviada como PHY; precisa frecuencia, potencia, modulación, y otras condiciones inalámbricas del enlace. La capa MAC (Media Access Control) define el formato de la manipulación de datos.

Las capas restantes definen otras medidas para la entrega de los datos y mejoras de protocolos relacionados, incluyendo la aplicación final. (Soluciones Informáticas, 2012).

2.2 Operación de topologías en Zigbee

El estándar Zigbee puede conectarse o implementarse en 3 topologías básicas: estrella, árbol de clústeres y *mesh* o malla. Aparte, dicho estándar especifica tres tipos diferentes de nodos que podrían estar presentes en una red Zigbee (Figura 2.3).

1. Coordinador/Gateway
2. Enrutador/Router
3. Dispositivo final/Sensor

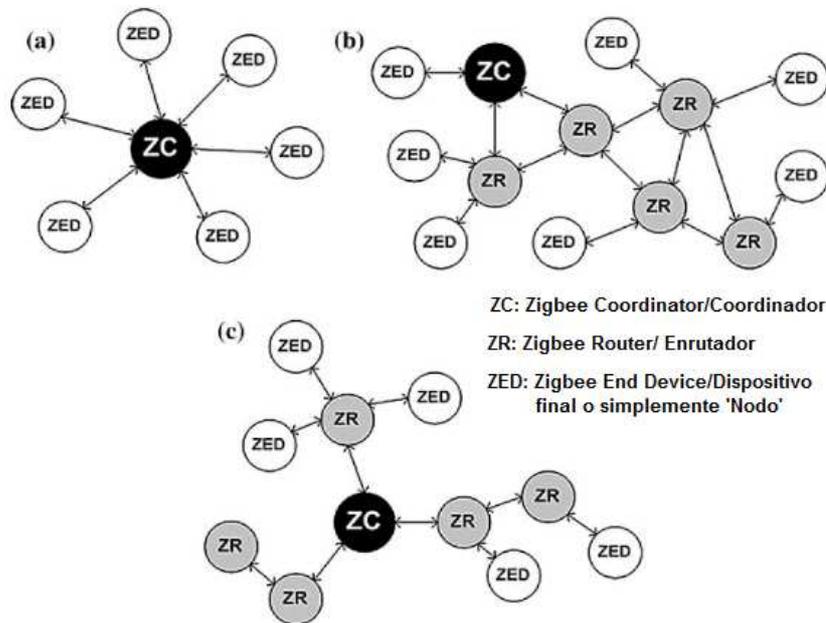


Figura 2. 3 Topología de Zigbee (a) Estrella, (b) *Mesh* y (c) Árbol

Fuente: (Tennina, 2013) y modificada por el autor

Coordinador. - Es el dispositivo responsable de configurar los parámetros clave de red, inicio de red, admisión de otros nodos y asignación de direcciones de red. Hay exactamente un coordinador de Zigbee en cada red. El coordinador debe conectarse a una fuente de suministro de energía confiable constante debido a su alto poder de procesamiento e incapacidad para 'dormir' (Kanjo & Trossen, 2015). Es el nodo que generalmente hace las funciones de pasarela (Gateway), ya sea con otra red, con una unidad central de procesado y en general con otro medio. Es el que crea la red y controla el acceso de nuevos nodos a ella (Roselló, 2009).

Si la red está operando en modo habilitado para el paquete que contiene información (*beacon*), el coordinador envía cuadros periódicos de éstos que servirán para

sincronizar el resto de los nodos. En una red de árbol de clústeres, todos los enrutadores recibirán un *beacon* de sus nodos ‘padres’ y enviarán sus propios ‘*beacons*’ para sincronizar los nodos pertenecientes a sus clústeres. (Buratti, Martaló, Verdone, & Ferrari, 2011).

Enrutador. - Pasa los mensajes de los dispositivos finales de Zigbee a otro enrutador o al coordinador de Zigbee. Es un equipo router utilizado para ampliar el área de cobertura de red y aumentar la confiabilidad de la red. (Garbarino, 2011). El investigador Tennina *et al* (2013) señala que el enrutador es un Dispositivo de funcionalidad completa (*Full-Function Device*, FFD) en terminología IEEE 802.15.4 puede actuar como un coordinador de red es decir actúa como enrutador una vez que se forma la red; gracias a la memoria adicional y capacidad de procesar y puede funcionar como Coordinador Zigbee, o puede ser usado en dispositivos de red que actúen de interfaz con los usuarios.

Dispositivo final. - Puede comunicarse sólo con el coordinador o un enrutador. No puede retransmitir datos de otros dispositivos. Esta relación permite que el nodo esté ‘dormido’ una cantidad significativa de tiempo, por lo tanto; proporcionará vida larga a la batería. Es decir, la mayor parte del tiempo están en modo durmiente, para minimizar así su consumo energético, despertándose periódicamente para ejecutar las tareas que le correspondan y comunicarse con su nodo ‘padre’ (Roselló, 2009).

El dispositivo final requiere la menor cantidad de memoria, por lo tanto, puede ser menos costoso de fabricar que un enrutador y un coordinador. Los dispositivos finales de Zigbee corresponden a dispositivos de función reducida (*Reduced Function Devices*, RFD) en estándar IEEE 802.15.4 (Buratti, Martaló, Verdone, & Ferrari, 2011) y (Revolvy, 2014).

En cuanto a la topología de conexión, la de tipo en estrella, es la más simple, pues necesita un solo Coordinador PAN, cada nodo se conecta directamente con el Coordinador central, todas las comunicaciones entre nodos se pasan a través del Coordinador. Además, en la topología estrella, el área de cobertura de red está limitada por el rango de transmisión del Coordinador, pero la red es sencilla en la configuración y la instalación. (Noorani, 2012).

En la figura 2.4 se aprecia los esquemas de conexión de topologías en árbol de clúster y de tipo *mesh* o malla.

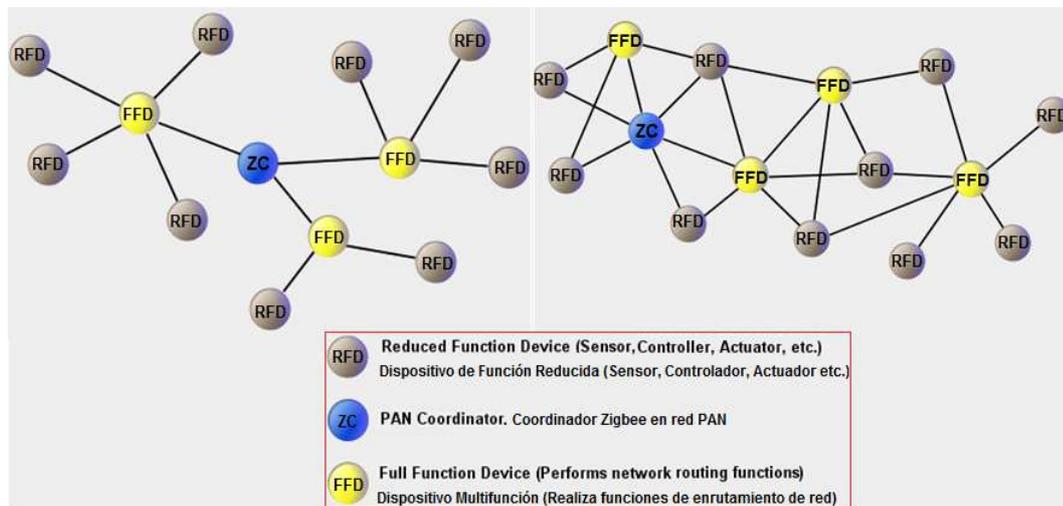


Figura 2. 4 Topología de árbol de clúster y tipo *mesh* o malla

Fuente: (Noorani, 2012) y modificada por el autor

En la jerarquía de la topología en árbol de clústeres cada nodo que se une a la red Zigbee, recibe una dirección de red temporal de 16 bits. La comunicación a nivel de red se realiza basándose en esta dirección mientras que la transmisión directa entre dos dispositivos vecinos se hace basándose en la dirección MAC (Syed, Piyush, & Tahsin, 2016).

La red de árbol de clústeres consiste en una serie de redes conectadas en estrella cuyos nodos centrales también están en comunicación directa con el único coordinador PAN. Usando un conjunto de *routers* y un único coordinador PAN, la red se forma en una malla interconectada de enrutadores y nodos finales que pasan información de nodo a nodo utilizando la ruta más provechosa. Si algún router individual se vuelve inaccesible, se pueden ‘descubrir’ y utilizar rutas alternativas, proporcionando una topografía de red robusta y fiable. (Buratti, Martaló, Verdone, & Ferrari, 2011).

En una red Mesh o malla, los nodos están interconectados con otros nodos de manera que múltiples vías conectan cada nodo. Las conexiones entre nodos se actualizan dinámicamente y se optimizan a través de una sofisticada tabla de enrutamiento de malla integrada (Wang, Tao, & Qian, 2014).

Las redes de malla están descentralizadas en la naturaleza; cada nodo es capaz de auto-descubrimiento en la red. Además, a medida que los nodos abandonan la red, la topología *Mesh* permite a los nodos reconfigurar rutas de enrutamiento basadas en la nueva estructura de red. Las características de la topología *Mesh* y el enrutamiento ad hoc proporcionan mayor estabilidad en condiciones cambiantes o fallas en nodos individuales. (Park, Stojmenovic, Young, & Yi, 2015).

Todas estas particularidades de Zigbee incorporadas a la reducción del consumo energético de los dispositivos, la creación de protocolos de comunicaciones adaptados a estos sistemas, así como las nuevas tecnologías de sensores y actuadores con sistemas micro electromecánicos (*Micro Electro-Mechanical Systems*, MEMS), han hecho que las WSNs se conviertan en una significativa opción para la adquisición y control en multitud de entornos y aplicaciones (Roselló, 2009, pág. 12)

2.3 Red de Sensores Inalámbricos (WSN)

La red WSN pueden integrar a su red, diversas plataformas basados en Zigbee, consiste normalmente en pequeños dispositivos espacialmente distribuidos para monitorear de forma cooperativa condiciones físicas o ambientales tales como temperatura, sonido, vibración, etc. Con conectividad WSN, los datos de sensores remotos de diferentes tipos son recolectados por la unidad central para procesamiento posterior y análisis. (Fernández, Ordieres, González, & Alba, 2009).

Las WSN son menos costosas y más flexibles que los sistemas de monitoreo por cable. Hay aplicaciones que se hacen viables sólo con WSNs porque el uso de cables entre dispositivos es demasiado caro o imposible en absoluto. Por ejemplo, en muchos problemas industriales, agrícolas, militares o ecológicos, el cableado físico es imposible o crearía disturbios extremos para otras operaciones. La WSN en comparación con otras tecnologías inalámbricas existentes, es la única tecnología que se dirige a la comunicación simple con bajas tasas de datos y bajo consumo de energía (Kanjo & Trossen, 2015).

2.3.1 Aplicaciones de WSN y Zigbee

Las aplicaciones incluyen interruptores de luz inalámbricos, medidores eléctricos en el hogar con pantallas, sistemas de gestión del tráfico, y otras de consumo y equipos industriales que requiere baja tasa de transferencia de datos inalámbrica de corto alcance.

La figura 2.5 muestra el uso de red WPAN que utiliza plataformas Zigbee para monitorear el nivel y la calidad de agua en una ciudad.

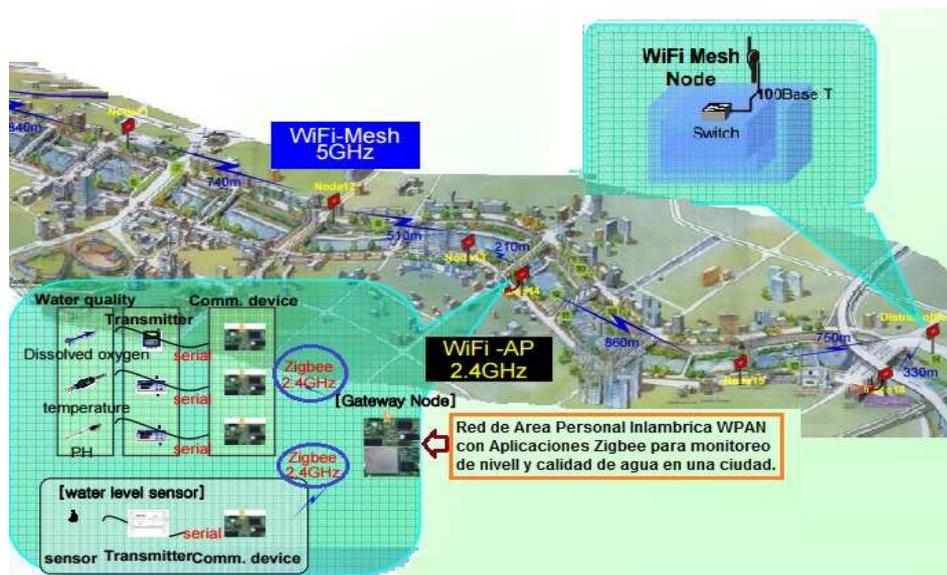


Figura 2. 5, Ejemplo de redes Zigbee para monitoreo de calidad y nivel de agua

Fuente: (Kanjó & Trossen, 2015)

En la gestión de emergencias, los nodos sensores pueden detectar y detectar el medio ambiente para pronosticar los desastres antes de que se produzcan. En aplicaciones biomédicas, Flores (2012, pág. 44) indica que los implantes quirúrgicos de sensores inalámbricos pueden ayudar a monitorear la salud de un paciente y juegan un papel importante en la detección temprana de una enfermedad.

Para la detección sísmica, el despliegue ad hoc de sensores en áreas volcánicas puede detectar la ocurrencia de terremotos y erupciones. Hoy en día en dispositivos móviles con aplicaciones sismo métricas se puede alertar con mínimo 8 segundos de anticipación un sismo de magnitud 5 Richter a una distancia de 10 Km del epicentro. (Amos, 2016).

2.4 Protocolos de enrutamiento para redes de sensores inalámbricos

Un algoritmo o protocolo de enrutamiento determina el camino o ruta de un mensaje desde una fuente hacia un destino. Generalmente dicho camino está compuesto por varios tramos que unen distintos nodos entre sí. Cada nodo envía sus mensajes al siguiente en la ruta, y así sucesivamente, hasta que el mensaje llegue al destino. Dicha ruta se debe planificar en base a un objetivo planteado, que puede ser maximizar el tiempo de vida de la red, seguridad de que todos los mensajes lleguen al nodo de destino o sumidero, mínima sobrecarga de la red, entre otros. Un buen camino es aquel que proporciona costo mínimo de acuerdo al objetivo planteado, no siempre el camino más corto es el más conveniente (Garbarino, 2011), (Wang, Tao , & Qian , 2014).

Las diferentes formas en las que operan los protocolos de enrutamiento los hacen adecuados para un tipo de aplicaciones. Hoy en día, existen muchas propuestas relacionadas con los algoritmos de enrutamiento en redes de sensores inalámbricas. (Querol, 2011).

Fernández Barcell (2010, pág. 8) indica que generalmente los nodos no tienen previo conocimiento de la topología de la red, por lo que deben realizar acciones de descubrimiento de rutas. La idea básica es que cuando un nodo se une a la red, anuncia su presencia y escucha los anuncios *broadcast* de sus vecinos. (Giandoménico, D'Agostino, Corti, Martínez, & Belmonte, 2010).

El nodo se informa del estado de sus vecinos y de la manera de encaminar paquetes a través de ellos. A su vez, puede anunciar al resto de nodos que pueden ser accedidos desde él, transcurrido un tiempo, cada nodo sabrá qué nodos tiene alrededor y una o más formas de alcanzarlos (Flores, 2012) y (Castro, 2014).

Los protocolos de enrutamiento para redes de sensores tienen diferentes clasificaciones de acuerdo a su esquema de direccionamiento. Existen protocolos centrados en los datos, generalmente usados en redes con una alta densidad de nodos. En este tipo de protocolos, lo que se direcciona no es un nodo en sí, sino el dato que se desea encaminar, a través de un identificador o meta-dato. De esta manera se reduce el tráfico redundante generado por nodos que cubren una misma área geográfica. (Barcell, 2010, pág. 13).

2.5 Particularidades de la clasificación de protocolos de enrutamiento

A pesar de las diferentes clasificaciones de protocolos de enrutamiento, todos los protocolos propuestos para trabajar en una WSN deben cumplir con las siguientes condiciones:

- Lograr un balance adecuado entre los beneficios del *overhead* y la disminución causada en la eficiencia de la red por la transmisión de datos de control.
- Mantener una tabla de enrutamiento moderadamente pequeña.
- Elegir la mejor ruta para un destino dado (ya sea la más rápida, confiable, de mejor capacidad, o de menor costo).
- Mantener la tabla regularmente para actualizarse ante la caída de nodos, cambios de posición o aparición de nuevos vecinos.
- Requerir una pequeña cantidad de mensajes y tiempo para converger (Mazón, 2011).

Según Kaur & Singh (2013, pág. 18) Los protocolos de enrutamiento pueden ser reactivos (*On-Demand*, bajo demanda) o proactivos (*Table-Driven*, manejado por tablas). En cambio, Querol (2011) al respecto señala que se puede clasificarlos en tres categorías en función de cómo la fuente es capaz de encontrar la ruta hasta su destino, proactivos, reactivos e híbridos.

Los proactivos son algoritmos en los que los nodos mantienen una serie de tablas con información de rutas al resto de nodos de la red (o al menos a una parte de ésta). Los nodos han de encargarse de mantener actualizados estos datos, por si la topología de la red cambia (Querol, 2011).

Un ejemplo de este tipo es el algoritmo DSDV (*Destination-Sequenced Distance Vector*), que tiene 2 tipos de paquetes para mantener actualizada la información de rutas ‘*full dump*’ e “incremental”, el primero conteniendo todos los datos y el segundo únicamente las diferencias respecto al último ‘*full dump*’. Sin embargo, tiene el problema de introducir una gran cantidad de *overhead* debido a las actualizaciones periódicas necesarias. (Abolhasan, Wysocki, , & Dutkiewicz, 2004) y (Mazón, 2011, pág. 4).

Los reactivos son algoritmos diseñados para reducir el problema del *overhead* de los proactivos, conocidos como bajo demanda, haciendo que los nodos únicamente guarden información de las rutas activas, y por tanto las rutas sean determinadas únicamente cuando hay datos a enviar, proceso que se hace normalmente por peticiones en *flood* hasta dar con el destino. Un ejemplo de este tipo de protocolos es el algoritmo AODV (*Ad hoc On-demand Distance Vector*), en donde los nodos envían paquetes ‘hello’ para colaborar en la búsqueda de rutas y en donde los paquetes llevan como cabecera los datos del nodo de destino (Abolhasan, Wysocki, & Dutkiewicz, 2004) y (Mazón, 2011, pág. 5).

En la tabla 2.2 se realiza un resumen de estos dos tipos de protocolos de enrutamiento.

Tabla 2. 2 Ventajas y desventajas de protocolos de enrutamiento proactivos y reactivos

Protocolos de ruteo	Proactivos	Reactivos
Descripción	<ul style="list-style-type: none"> - Operado por tablas (<i>Table-Driven</i>). - Buscan rutas periódicamente, suponiendo que serán útiles. 	<ul style="list-style-type: none"> - Bajo demanda (<i>On-demand</i>). - Buscan una ruta sólo cuando se necesita.
Ventajas	<ul style="list-style-type: none"> - Una ruta puede ser utilizada de forma inmediata, sin ningún retardo. 	<ul style="list-style-type: none"> - Solicita un ancho de banda reducido para mantener las tablas de ruteo. - Hace un uso eficiente de la energía.
Desventajas	<ul style="list-style-type: none"> - Introduce mayor tráfico de control. - Utiliza una cantidad elevada de ancho de banda. - Produce congestión en la red. - Reacción lenta en reestructuración y fallas. 	<ul style="list-style-type: none"> - Tiempo de latencia alto en la búsqueda de rutas. - Una inundación excesiva puede llevar al congestionamiento de la red

Fuente: (Datta & Soundaralaskshmi, 2010) y (Kaur & Singh, 2013) . Traducido por el autor

Otra clasificación de tipos de protocolos de enrutamiento viene dada por modelos.

Modelo de Difusión Directa (modelo de un salto)

Este es el modelo más simple y simboliza la comunicación directa, es decir todos los nodos en la red transmiten a la estación base, obviamente al operar de dicha forma consume mucha energía y no es viable porque todos los nodos tienen un rango de transmisión limitado, sus transmisiones no siempre enlazan la estación base. Por aquello la comunicación por difusión directa no es una óptima solución.

Modelo Multi-salto (*multi-hops*)

En este tipo de protocolo, un nodo transmite a la estación base reenviando sus datos a uno de sus vecinos, el cual está más próximo a la estación base, a la vez que éste

enviará a otro nodo más próximo hasta que llegue a la base. Entonces la información es transmitida hacia el receptor de destino salto a salto desde un nodo a otro hasta que llega a destino. Fernández Barcell (2010, pág. 8) al respecto indica que un gran número de protocolos utilizan este modelo, entre ellos. Este tipo de modelo puede ser una solución viable.

En la Tabla 2.3 se muestra la comparación entre estos dos modelos para protocolo de ruteo.

Tabla 2. 3 Comparación de los modelos de enrutamiento salto a salto y fijado en origen.

	Salto a salto	En el origen
Descripción	SIMPLICIDAD: Los nodos han de tener un conocimiento parcial de la red (saber qué rutas son las mejores)	Los sistemas finales han de tener un conocimiento completo de la red
Complejidad	En los sistemas intermedios ya que son los que tienen que encaminar	Recae toda en los sistemas finales
Problemas de bucle	Sí pueden ocurrir: no se tiene una visión completa de la red (INCONSISTENCIA)	No hay bucles: el sistema final fija la ruta (ROBUSTEZ).

Fuente: (Sánchez, 2010)

Si se comparan ambas técnicas, en el ruteo en la técnica de salto a salto (*hop by hop routing*) cada router resuelve solo el sucesivo salto y la información del enrutado se almacena en los *routers*. En el enrutamiento de origen (*Source Routing*) la ruta se instituye antes de realizar el envío y los datos de enrutado se guarda en el paquete.

Los bucles (situación que se da cuando los paquetes pasan más de una vez por un nodo) ocurren porque los criterios de los nodos no son coherentes, generalmente debido a que los criterios de enrutamiento no han convergido después de un cambio en la ruta de un paquete; cuando por cualquier causa un paquete sufre un cambio de enrutamiento, la red tarda en adaptarse a ese cambio pues la noticia del mismo tiene que llegar a todos los nodos. (Sánchez, 2010).

En la tabla 2.4 se muestra un listado de protocolos basados en los tipos proactivo y reactivo según los dos modelos descritos.

Tabla 2. 4 Clasificación de protocolos de enrutamiento

Protocolos	Enrutamiento salto a salto	Enrutamiento en origen
Proactivos	DSDV, OLSR, CGSR, WRP, TBRPF	

Reactivos	AODV, LMR, TORA	DSR, LQSR
-----------	-----------------	-----------

Elaborado por: el autor

2.5.1 Consumo de energía

La energía es utilizada por los nodos en tres funciones principales: medición, procesamiento y comunicación. Se ha comprobado que el proceso de comunicación consume la mayor parte de la energía disponible, así que se intenta minimizar la tarea de transmisión realizando todo el procesamiento local posible (Morales & Gómez, 2007) (Kaur & Singh, 2013).

Una de las limitaciones fundamentales de las redes de sensores es la capacidad de batería. En las redes móviles y ad hoc los protocolos se diseñan con el objetivo de minimizar la cantidad de saltos que atraviesa un paquete. A diferencia de este tipo de redes, en las WSNs es crucial minimizar el consumo de energía empleado para la tarea de enrutamiento, o resolver el problema del enrutamiento de menor energía a partir de diferentes enfoques de diseño, que se manifiestan en las métricas de energía utilizadas (Barcell, 2010).

La técnica de energización para el enrutamiento no aparece como un modelo en sí misma, sino que la eficiencia en términos de energía se busca utilizando técnicas adicionales. El empleo de métricas de energía en el enrutamiento no siempre aparece en el diseño de protocolos que buscan eficiencia en el uso de la energía (Giandoménico, D'Agostino, Corti, Martínez, & Belmonte, 2010).

Se presentan varias métricas de diseño que permiten construir rutas eficientes en términos de energía, y que aparecen muy frecuentemente en los protocolos para redes de sensores (Chang & Tassiulas, 2005):

- **Minimizar la energía total de la ruta**

Con el enfoque de energía total mínima o MTE (*Minimum Total Energy*), se busca minimizar la energía de transmisión y recepción total consumida por un paquete para alcanzar el destino. Sucede que, si todo el tráfico se encamina a través de las rutas más económicas, los nodos en esa ruta se agotan rápidamente, particionando la red, por lo que no se recomienda utilizar esta métrica en forma aislada. (Garbarino, 2011).

- **Maximizar el tiempo hasta que la red se particiona.**

Se determinan los nodos que al ser extraídos causan particiones, y se trata de balancear la carga sobre los mismos.

- **Uniformizar el consumo de energía**

Este objetivo se puede alcanzar mediante el uso de rutas compuestas por los nodos de mayor energía residual, evitando los nodos de menor capacidad restante. Se sacrifica el uso de rutas más cortas por el beneficio de posponer al máximo posible la muerte del primer nodo. (Garbarino, 2011) y (Gómez, Posada, & Vallejo, 2014).

- **Minimizar el costo por paquete**

Se seleccionan las rutas de menor costo total, que equivale a la suma de los costos de cada enlace que se utiliza. Si se define adecuadamente el costo del enlace, puede utilizarse la métrica con diferentes objetivos. Por ejemplo, si el costo depende de la energía residual, se puede lograr aumentar el tiempo hasta el particionamiento de la red (Woo & Singh, 2008, pág. 187)

2.6 Protocolos de enrutamiento tradicionales

Debido a la heterogeneidad de los escenarios de aplicación para las redes de sensores inalámbricos, existe una gran cantidad de protocolos de enrutamiento diseñados para cubrir las necesidades de cada una de ellas. En los siguientes apartados se describen algunos de los protocolos de enrutamiento más populares en redes de sensores, atendiendo a las diferentes clasificaciones.

2.6.1 Protocolo AODV

AODV es el protocolo usado por defecto en redes Zigbee de topología mesh, y pertenece al conjunto de protocolos de enrutamiento para redes planas. Fue diseñado para adaptarse rápidamente a cambios en la topología de la red, con bajas demandas de procesamiento y comunicación.

Cada destino dentro de la tabla de enrutamiento tiene asociado un temporizador y un número de secuencia. La función del temporizador es evitar el uso de enlaces de los que no se ha recibido reportes de estado desde hace mucho tiempo, mientras que el número de secuencia asociado permite distinguir entre información nueva y

antigua. De esta manera se puede evitar la formación de bucles y la transmisión de rutas obsoletas (Roselló, 2009, pág. 60).

El descubrimiento de rutas está basado en ciclos de petición y respuesta, y la métrica utilizada es el número de saltos desde la fuente hasta el destino. Cuando un nodo desea enviar datos a un destino envía mediante *broadcast* un mensaje de petición de rutas RREQ (*Route Request*). Este mensaje se propaga a través de la red, y cualquier nodo que conozca una ruta hacia el destino, o el mismo destino, contestarán a la petición con la ruta solicitada mediante un mensaje RREP (*Route Reply*) (Fernández, Ordieres, González, & Alba, 2009).

En la tabla 2.5, se muestra el formato del paquete de solicitud de ruta.

Tabla 2. 5 Representación del formato del paquete solicitud de ruta

Dirección de Origen	ID de Solicitud	Dirección de Destino	Número de secuencia de Origen	Número de secuencia de Destino	Cuenta de Saltos
---------------------	-----------------	----------------------	-------------------------------	--------------------------------	------------------

Elaborado por: el autor

Se aprecia en el formato de paquete las direcciones de red de origen y destino, también contiene un ID de solicitud, que es un contador local que se mantiene por separado en cada nodo y se incrementa cada vez que se difunde un paquete de solicitud de ruta. En conjunto, los campos dirección de origen e ID de solicitud identifican de manera única al paquete de solicitud de ruta, a fin de que los nodos descarten cualquier duplicado que pudieran recibir (Roselló, 2009).

Además del contador ID de solicitud, cada nodo mantiene también un segundo contador de secuencia que se incrementa cada vez que se envía un paquete de solicitud de ruta (o una respuesta al paquete de solicitud de ruta de algún otro nodo). El cuarto campo es un contador de secuencia del emisor del mensaje de descubrimiento, y el quinto campo representa el valor más reciente del número de secuencia que el origen ha recibido del destino (0 si nunca se han comunicado). El último campo es un contador de saltos, se inicializa en 0 y se incrementa con cada salto que realiza el paquete (Sánchez, 2010).

Cuando un paquete de solicitud de ruta llega a un nodo vecino se procesa de la siguiente manera:

1. El par (Dirección de origen, ID de solicitud) se busca en una tabla de historia local para ver si esta solicitud ya se había recibido y procesado. Si es un duplicado, se descarta y el procesamiento se detiene. En caso contrario, el par se introduce en la tabla de historia a fin de que se puedan rechazar futuros duplicados, y el procesamiento continúa.
2. El receptor busca el destino en su tabla de enrutamiento. Si se conoce una ruta reciente al destino, se regresa un paquete de respuesta de ruta RREP al origen, que le indica cómo llegar al destino. Una ruta se considera reciente si el número de secuencia de destino almacenado en la tabla de enrutamiento es mayor que o igual al número de secuencia de destino del paquete de solicitud de ruta. Si es menor, la ruta guardada es más antigua que la que el origen tenía para el destino, por lo que se ejecuta el paso 3.
3. Puesto que el receptor no conoce una ruta reciente al destino, incrementa el campo “cuenta de saltos” y vuelve a difundir el paquete de solicitud de ruta. También extrae los datos del paquete y los almacena como una entrada nueva en su tabla de rutas invertidas. Esta información se utiliza para construir la ruta invertida a fin de que la respuesta pueda regresar posteriormente al origen. También se inicia un temporizador para la nueva entrada de ruta invertida. Si el temporizador expira, la entrada se borra. (Robles, Marrone, & Díaz, 2010).

Cuando el paquete de solicitud difundido llega al nodo destino, éste construye un paquete de respuesta de ruta. El formato del paquete RREP se muestra en la tabla 2.6.

Tabla 2. 6 Formato del paquete de respuesta de ruta

Dirección de Origen	Dirección de Destino	Número de secuencia de Origen	Cuenta de Saltos	Tiempo de vida
---------------------	----------------------	-------------------------------	------------------	----------------

Elaborada por: el autor

Los campos dirección de origen y dirección de destino se invierten respecto a la solicitud entrante, y el número de secuencia de origen se copia de su contador en memoria. El campo “cuenta de saltos” se inicializa en 0, mientras que el campo “tiempo de vida” determina el tiempo en milisegundos para el cual los nodos que reciben el mensaje RREP consideran la ruta válida.

El paquete se encamina a través de la ruta inversa, siendo inspeccionado por cada nodo intermedio, y se introduce en la tabla de enrutamiento local como una ruta al destino si se cumple una o más de las siguientes condiciones:

1. No se conoce una ruta al destino (generador del paquete RREP)
2. El número de secuencia de origen en el paquete de respuesta de ruta es mayor que el valor en la tabla de enrutamiento.
3. Los números de secuencia son iguales, pero la nueva ruta es más corta.

Debido a que es posible mover o apagar los nodos, la topología puede cambiar de manera espontánea. El algoritmo debe ser capaz de manejar cualquiera de estas circunstancias. Para ello, cada nodo difunde de manera periódica un mensaje de saludo (Hello), que espera que cada uno de sus vecinos responda. Si no se recibe ninguna respuesta, el difusor sabe que el vecino se ha movido del alcance y ya no está conectado a él. De manera similar, si el difusor trata de enviar un paquete a un vecino que no responde, se da cuenta de que el vecino ya no está disponible. Esta información se utiliza para eliminar rutas que ya no funcionan. (Garbarino, 2011).

Para cada destino posible, cada nodo N mantiene un registro de los vecinos que le han proporcionado un paquete para el destino en cuestión durante los últimos ΔT segundos. Éstos se llaman vecinos activos de N para ese destino. Para gestionar toda esta información, el nodo N mantiene una tabla de enrutamiento codificada por destino que contiene el próximo salto a utilizar, la cuenta de saltos, el número de secuencia de destino más reciente, y la lista de vecinos activos para cada destino.

Cuando cualquiera de los vecinos de N se vuelve inalcanzable, N verifica su tabla de enrutamiento para ver cuáles destinos tienen rutas en las que se incluya al vecino ahora perdido. Para cada una de estas rutas, se les informa a los vecinos activos que su ruta a través de N ahora es inválida, y que se debe eliminar de sus tablas de enrutamiento. Seguidamente, los vecinos activos indican este hecho a sus vecinos activos, y así sucesivamente, de manera recursiva, hasta que las rutas que dependen del nodo perdido se eliminan de todas las tablas de enrutamiento (Mazón, 2011).

En la figura 2.6 se puede ver una representación gráfica del mecanismo de creación de rutas anteriormente descrito.

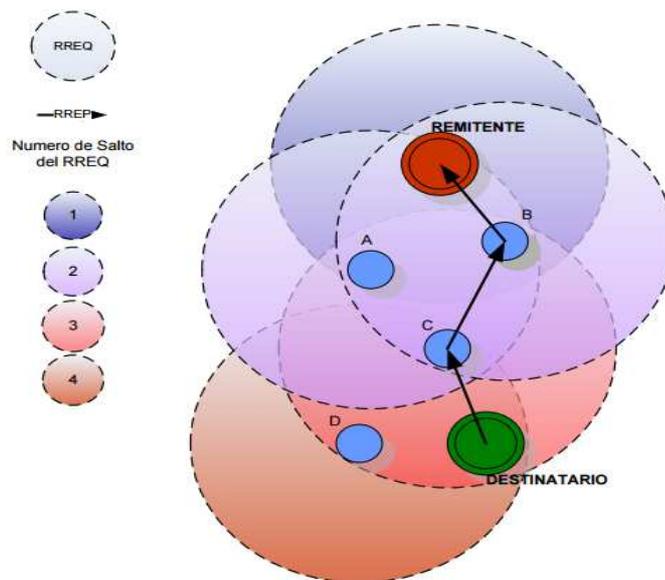


Figura 2. 6 Creación de rutas con AODV.
Fuente: (Roselló, 2009)

AODV ha servido de inspiración para el desarrollo de otros protocolos específicos para redes de sensores inalámbricos. Dentro de estos protocolos se pueden mencionar a REL (*Routing by Energy and Link Quality*), que provee un esquema de enrutamiento consciente de la energía, y nst-AODV (*not so tiny-AODV*), una versión simplificada de AODV (Robles, Marrone, & Díaz, 2010).

Trabaja bien en condiciones donde la movilidad es alta, incorpora mecanismos para evitar la formación de bucles y detecta rápidamente cambios en la topología de la red adaptando las rutas a estos cambios. Sin embargo, tiene como desventaja que ningún nodo tiene conocimiento total de la topología de la red, cada uno sólo tiene datos de los nodos con los que necesita comunicarse.

2.6.2.1 Protocolo DYMO (*Dynamic MANET On-demand*)

DYMO es un sucesor simplificado del protocolo AODV, está especialmente diseñado para redes ad-hoc con movilidad variable y con diferentes patrones de tráfico. Entre sus ventajas está el tener un menor consumo de ancho de banda y de procesamiento computacional, ya que no envía paquetes a menos de que sea necesario. En redes grandes tiene un mejor comportamiento en escenarios donde cada nodo sólo se comunica con una limitada porción de los demás (Peral, Moncayo, & Morillo, 2007) y (Agrawal, Jain, , & Sharma, 2011).

Como todos los protocolos reactivos, DYMO se compone de dos operaciones: descubrimiento y mantenimiento de rutas. Las rutas se descubren bajo demanda cuando un nodo necesita enviar un paquete a un destino que no está en su tabla de enrutamiento. Un mensaje RREQ es enviado a todos los nodos, los cuales al recibir la petición verifican si el destino se encuentra en su tabla de enrutamiento. Si el nodo destino se encuentra en la tabla, estos proceden a verificar si el enlace aún se encuentra activo. Una vez el nodo destino recibe el RREQ, lanza un mensaje RREP, que es enviado de forma unicast, salto a salto hacia el nodo origen con la información del número de saltos intermedios y su correspondiente ruta (Peral, Moncayo, & Morillo, 2007) y (Agrawal, Tripathi, & Tiwari, 2012)

A diferencia del protocolo AODV, el cual sólo genera entradas a la tabla de enrutamiento para el nodo de destino y el siguiente, en el protocolo DYMO, además de la información acerca de una ruta de destino solicitado, un nodo también recibe los datos acerca de todos los nodos intermedios de un camino recién descubierto (Dietrich, Sommer, & Dressler, 2007). Como se muestra en la figura 2.7 (a), en AODV, cuando el nodo A inicia un proceso de descubrimiento de ruta para el nodo D, sólo tiene conocimiento sobre las rutas al nodo B, su vecino del siguiente salto, y el nodo destino D, mientras que en DYMO (figura 2.7 b), el nodo A adicionalmente tiene conocimiento sobre la ruta al nodo C y B.

Esta característica en DYMO se conoce como la acumulación de ruta.

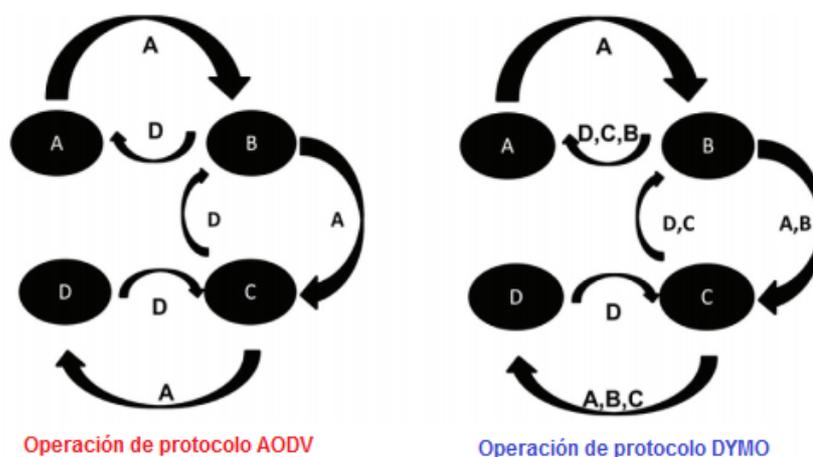


Figura 2. 7 (a) Desempeño Protocolo AODV (b) Desempeño Protocolo DYMO

Fuente: (Gómez, Posada, & Vallejo, 2014)

2.6.2 Protocolo SPIN (*Sensor Protocol for Information Via Negotiation*)

SPIN es una familia de protocolos centrados en los datos, diseñado sobre la base de dos ideas fundamentales. Primeramente, si bien intercambiar la información censada puede ser una actividad costosa, intercambiar datos sobre los datos censados no necesariamente lo es. (Sarkar & Murugan, 2016).

En segundo lugar, los nodos deben adaptarse a cambios en su disponibilidad de recursos para extender su vida útil y la de la red. Para poder superar las deficiencias del *flooding* clásico (superposición o inundación de recursos), SPIN incorpora dos técnicas innovadoras en su momento (Heinzelman, Kulic, & Balakrishnan, 2009)

1. Negociación: Para evitar la implosión y la superposición, los nodos negocian entre sí la transmisión de los datos, utilizando meta-datos para describir la información disponible.

2. Adaptación de recursos: Cada nodo dispone de un administrador de recursos que mantiene registro del consumo de recursos. Cuando la energía es baja, el nodo se abstiene de participar en ciertas actividades, por ejemplo, el envío de información de terceros. (Garbarino, 2011).

SPIN asume una topología de red plana, donde cualquier nodo sensor puede hacer función de sumidero. Asimismo, cualquier nodo puede ser fuente de datos, es decir, genera datos a partir del censado de un parámetro del medio, y la información debe ser diseminada a través de la red (Querol, 2011).

Los nodos se comunican mediante tres tipos de mensajes:

- a) **ADV:** aviso o anuncio de nuevos datos. Cuando un nodo dispone de nueva información, lo informa mediante un mensaje ADV que contiene metadatos.
- b) **REQ:** petición de datos. Cuando un nodo recibe un mensaje ADV, verifica si ya ha recibido los datos que describe el metadato. En caso negativo, responde con un mensaje REQ pidiendo la nueva información, que será enviada en un mensaje DATA.
- c) **DATA:** mensaje de datos. Contiene la información propiamente dicha, encabezada por el metadato.

Las rondas de negociación se muestran en las figuras 2.8 y 2.9, obtenidas del trabajo de Heinzelman, Kulic & Balakrishnan (2009), la negociación (figura 2.8a), comienza en 1), donde el nodo a anuncia que dispone de nuevos datos. En 2) el nodo b solicita la transmisión de la información nueva.

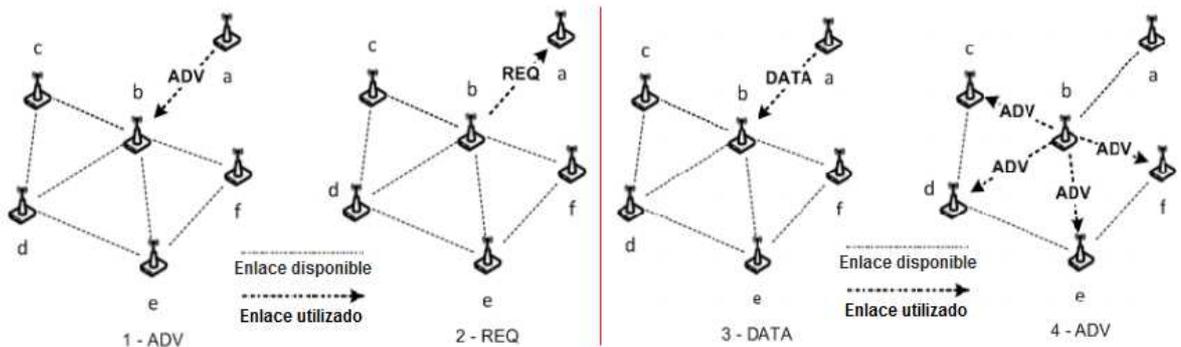


Figura 2. 8 (a) Negociación SPIN pasos 1 y 2; (b) Negociación SPIN pasos 3 y 4
Fuente: (Garbarino, 2011)

Luego, en la parte (b) de la figura 2.8, en 3) el nodo a transmite la información propiamente dicha al nodo b, encabezada por el meta-dato. En 4) el nodo b anuncia a sus vecinos que dispone de nuevos datos. Por último, en la figura 2.9, en 5) los nodos c y e solicitan la transmisión de los nuevos datos de b. Finalmente en 6) El nodo b transmite los datos a los nodos que lo han solicitado. (Garbarino, 2011)

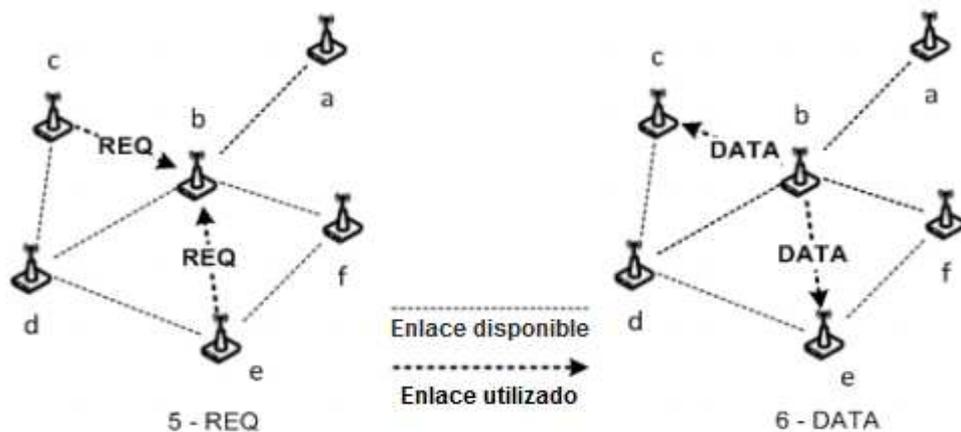


Figura 2. 9 Negociación SPIN pasos 5 y 6
Fuente: (Garbarino, 2011)

Es importante aclarar que en este protocolo no todos los nodos necesitan responder a los mensajes que reciben, por ejemplo, un nodo que recibe un mensaje ADV de

un vecino no deberá responder con REQ si ya dispone de la información descrita en el metadato.

A continuación, se enlista algunos de los protocolos de la familia SPIN:

- **SPIN-PP (SPIN *Point-to-Point*):** Es el más básico de los protocolos de la familia SPIN. Es un diseño optimizado para una red teórica que utiliza un medio de transmisión punto a punto, asumiendo que es posible que dos nodos A y B se comuniquen exclusivamente uno con el otro, sin interferir la comunicación con otros nodos. SPIN-PP trabaja en tres etapas, cada una de las cuales corresponde a uno de los mensajes descritos anteriormente.
- **SPIN-EC:** Agrega lógica de conservación de energía bastante simple a SPIN-PP. Cuando un nodo observa que su energía residual se aproxima a un umbral inferior de carga de batería, reduce su participación en el protocolo. Si el nodo recibe un dato nuevo, sólo iniciará la ronda de negociación si puede completar todas las etapas (anuncio de datos y respuesta de petición). De la misma manera, si recibe un anuncio, sólo enviará una petición si tiene energía suficiente para recibir el dato.
- **SPIN-BC:** es una versión de SPIN que agrega optimizaciones para cuando se utiliza un canal de transmisión compartido. Cuando un nodo envía un mensaje, éste es recibido por todos los nodos dentro del rango de alcance de la radio, sin importar el destino original de mensaje. (Querol, 2011).

2.6.3 Protocolo LEACH (*Low Energy Adaptive Clustering Hierarchy*)

LEACH es un protocolo que plantea la utilización de clusters (porciones o conjuntos) de nodos consintiendo distribuir el consumo de energía de forma más uniforme entre los sensores de la red. Garbarino (20011) indica que se especifica por concebir el rotamiento al azar, el rol de cabecera del *cluster*. Interiormente en un *cluster*, la cabecera regulariza localmente la transmisión y fusión de datos para comprimir la información y luego transmitirla directamente a la estación base, minimizando el ancho de banda requerido. La topología resultante se esquematiza en la figura.2.10.

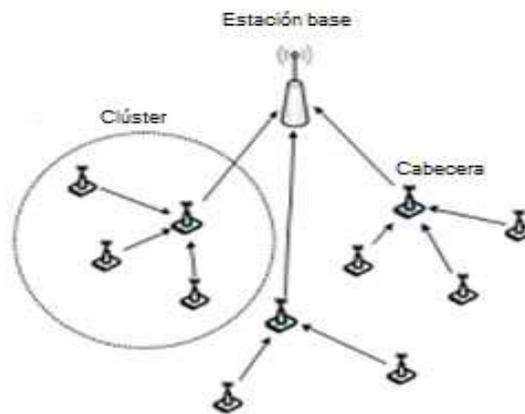


Figura 2. 10 Esquema de red con protocolo LEACH

Fuente: (Garbarino, 2011)

El modelo de red para el cual se desarrolla LEACH comprende cientos o miles de nodos sensores económicos y eficientes en términos de energía. Los nodos son homogéneos y se encuentran restringidos en cuanto a la duración de la batería. La calidad de la información se mejora con la utilización de un gran número de nodos. (Balakrishnan, Heinzelman, & Chandrakasan , 2000)

LEACH opera en rondas divididas en las siguientes cuatro fases (Balakrishnan, Heinzelman, & Chandrakasan , 2000):

1. Anuncio

En la fase de inicialización, cada clúster decide si se vuelve cabecera dependiendo del porcentaje de cabeceras que debe tener la red y el número de veces que el nodo ya ha tenido el rol. Cada nodo que se elige a sí mismo como cabecera hace *broadcast* del anuncio al resto de los nodos, utilizando CSMA (*Carrier Sense Multiple Access*). Los nodos que no son cabecera mantienen su radio encendida, y eligen su cabecera sobre la base de la fuerza de la señal recibida de los anuncios que escuchan.

2. Armado del cluster

Los nodos eligen la cabecera que requiere la menor energía de transmisión, e informan a la cabecera seleccionada que se harán miembros del cluster usando el protocolo CSMA.

3. Planificación del canal

Una vez que la cabecera recibe todos los mensajes de los nodos miembros, crea un calendario TDMA (*Time Division Multiple Access*) donde asigna un período de tiempo de transmisión a cada nodo, y lo informa por difusión a sus miembros.

4. Transmisión de datos

Cada nodo envía la información durante la franja de tiempo de transmisión que le fue asignada. Mientras el nodo no está transmitiendo, puede apagar su radio. Cuando el nodo cabecera recibe los datos de todos los nodos del clúster, puede combinarlos para comprimir la señal que se enviará a la estación base en una transmisión de alta energía. (Garbarino, 2011)

Luego de un determinado tiempo de transmisión, comienza una nueva ronda. Para minimizar el costo indirecto asociado al mantenimiento de la jerarquía, la fase de transmisión es mucho más larga que la de armado de clústeres.

LEACH es un algoritmo distribuido y no requiere coordinación de la estación base, ni conocimiento global de la topología de red. Es eficiente al distribuir el consumo de energía de manera uniforme a lo largo de la red, lo que incrementa la vida útil de la misma. Dado que las cabeceras pueden comprimir o fusionar los datos recibidos de los nodos locales, contribuyen a reducir el tráfico global. (Querol, 2011).

Como desventaja de este protocolo, la comunicación de cabecera a estación base es de salto-único, lo cual en realidad es una limitación, ya que en áreas amplias puede que el alcance de la radio no sea suficiente (Garbarino, 2011). A pesar de las múltiples propuestas, no existe un protocolo de enrutamiento adecuado para todos los escenarios de aplicación.

La continua necesidad de encontrar nuevas soluciones, y de tomar decisiones óptimas en base a diferentes métricas conlleva a la introducción de técnicas de Inteligencia Computacional al enrutamiento en redes de sensores inalámbricos.

2.7 Protocolos de enrutamiento basados en Inteligencia Computacional

Hay muchos problemas que requieren algoritmos cuyo esfuerzo computacional es tan grande que a pesar de la capacidad de cálculo de los ordenadores actuales se tardaría años en encontrar una solución óptima. Tales problemas se presentan en

biología, robótica, aeronáutica, etc. Son problemas de secuenciación, reconocimiento de patrones, visión artificial, percepción, etc. (Muñoz, 2010, pág. 33).

La Inteligencia Computacional (IC) es parte del paradigma de Inteligencia Artificial, investigadores diseñan algoritmos de enrutamiento para ambientes complejos y dinámicos como las redes de sensores. Estos algoritmos exhiben un comportamiento adaptativo y suelen ser robustos ante cambios de topología y fallos en la comunicación. Kulkarni, Forster & Kumar (2011) indican que las técnicas más utilizadas son aquellas basadas en aprendizaje por refuerzos, inteligencia de enjambres, algoritmos evolutivos y lógica difusa.

Se pueden encontrar otras técnicas dentro del campo del aprendizaje de máquina, pero no son adecuados para su aplicación en redes de sensores debido a que sus propiedades no se adecúan a los requerimientos de estos sistemas. (Kulkarni, Forster, & Kumar, 2011)

Las IC utiliza una combinación de cinco principales técnicas complementarias. La lógica difusa que permite a la computadora entender el lenguaje natural, redes neuronales artificiales que permiten al sistema aprender datos de la experiencia operando como las biológicas, la computación evolutiva, que se basa en el proceso de la selección natural, la teoría del aprendizaje y métodos probabilísticos que ayudan a hacer frente a la incertidumbre y la imprecisión (Kulkarni, Forster, & Kumar, 2011).

2.7.1 Algoritmos basados en Redes Neuronales

Una red neuronal es el ensayo de efectuar una simulación computacional del comportamiento de partes del cerebro humano mediante la réplica en pequeña escala de los patrones que éste desempeña para la formulación de resultados a partir de los sucesos percibidos. Son sistemas paralelos para el procesamiento de la información, inspirados en el modo en el que las redes de neuronas biológicas del cerebro procesan los datos. (Machado, y otros, 2013)

Una característica fundamental de las redes neuronales es que tienen altos requerimientos computacionales, por lo que no se han desarrollado muchos algoritmos de enrutamiento basados en este paradigma.

Barbancho, León & Molina (2009) estudiaron un algoritmo de enrutamiento manejado por QoS (Quality of Service) denominado SIR (*Sensor Intelligence Routing*), donde en cada nodo usa una red neuronal para manejar el tráfico de datos.

El rendimiento de SIR es comparado con el de los protocolos *Directed Diffusion* y *Energy Aware Routing*. En ambos casos, SIR mostró un rendimiento superior en cuanto a latencia promedio y consumo de energía, sin embargo, la propuesta es costosa si se tiene en cuenta que cada nodo debe hacer un ping a sus vecinos para determinar los parámetros del enlace. (Barbancho, León, & Molina, 2009) Además, la implementación de un mapa auto-organizado en nodos sensores conlleva a un gasto computacional significativo que puede impactar en la limitada capacidad de memoria, procesamiento y energía de las motas (Kulkarni, Forster, A, & Kumar, G., 2011).

2.7.2 Algoritmos basados en Lógica Difusa

La Lógica Difusa es una lógica multi evaluada que permite que valores intermedios sean definidos dentro de valores umbrales convencionales. En este aspecto, se asemeja al razonamiento humano, que es capaz de incluir una medida de imprecisión o incertidumbre, que es marcada por el uso de variables lingüísticas tales como muy, mucho, frecuentemente, poco, muy poco, etc. Los sistemas difusos permiten el uso de conjuntos difusos para sacar conclusiones y tomar decisiones para optimizar algún parámetro.

Una elección acertada de las cabezas de cluster puede reducir el consumo de energía y extender el tiempo de vida de la red. Un algoritmo de lógica difusa basado en la energía, concentración y centralización es propuesto en Gupta, Riordan & Sampalli (2005) para la elección de la cabeza del cluster. El estudio usa un modelo de red en el cual todos los nodos sensores transmiten a la estación base información sobre su ubicación y energía disponible.

La estación base entonces toma en cuenta, junto con estos parámetros, el número de nodos en la vecindad y la distancia entre los mismos, para determinar cuáles son

los nodos adecuados para ejercer la función de cabecera. Para ello, clasifica las variables “energía del nodo” y “concentración del nodo” en tres niveles: bajo, medio y alto. De modo similar lo hace con la variable de centralización del nodo en cerca, adecuado y lejano. El resultado difuso que representa la probabilidad de que un nodo sea elegido como cabeza de cluster es dividido en siete niveles que sirven como indicador de clasificación.

En el artículo se observa un incremento sustancial en la vida de la red en comparación con una red que usa el protocolo LEACH. Para un escenario de 20 nodos ubicados en un campo de 100 m x 100 m, el número de rondas de datos antes de la caída del primer nodo es en promedio 1.8 veces mayor que en LEACH. Sin embargo, esta alternativa supone un gasto de recursos en recolectar la información necesaria en la estación base antes de determinar las cabezas del cluster.

2.7.3 Algoritmos Evolutivos (AE)

Los algoritmos evolutivos modelan la evolución natural, que es el proceso de adaptación en vista de mejorar las capacidades de supervivencia a través de procesos como la selección natural, supervivencia del más fuerte, reproducción, mutación, competición y simbiosis. Los AE usan una población de candidatos llamados cromosomas que se reproducen, tomando sus hijos diferentes características. Los cromosomas que mejor se adaptan son seleccionados para formar una nueva generación, los otros son eliminados. El proceso se repite generación tras generación, hasta obtener una solución lo suficientemente aceptable (Maldonado, 2013).

Islam & Hussain (2006) estudiaron la propuesta de un protocolo de enrutamiento multi-salto llamado GA-Routing (*Genetic Algorithm Routing*). Está basado en un algoritmo genético y su objetivo es maximizar la longevidad de la red en términos del tiempo transcurrido antes de la caída del primer nodo. El algoritmo propuesto trabaja generando árboles de agregación que abarcan todos los nodos de la red.

Aunque el mejor árbol de agregación es el camino más eficiente hacia la estación base, el uso continuo del mismo puede conllevar al fallo de un grupo de nodos antes que otros. El objetivo entonces es encontrar un árbol de agregación y el número de veces a usar antes de crear otro árbol. Dadas las localizaciones de los nodos sensores

y de la estación base, el algoritmo genera una secuencia de caminos para el enrutamiento, y es capaz de reducir el número de transmisiones gracias a una técnica de agregación de datos. De esta manera, se asegura un menor consumo energético en las tareas de comunicación.

Los resultados de la simulación muestran que *GA-Routing* provee un mayor tiempo de vida de la red que un algoritmo simple que siempre selecciona el mejor árbol. Sin embargo, es un protocolo centralizado, y no se consideran los costos de diseminación de los caminos. A esto se debe agregar que las simulaciones se realizan sobre una aplicación en Java, y no sobre un simulador de redes, donde se obtienen resultados más reales.

Otro algoritmo de enrutamiento basado en un algoritmo genético es *Two-TierGA (Two-Tier Genetic Algorithm)*. Está diseñado para maximizar el tiempo de vida de una red de dos niveles. En este tipo de redes un grupo de sensores actúan como cabezas de cluster, y si alguno de estos nodos falla debido a falta de energía en la batería, el cluster completo deja de operar. Esto influye en la carga que deben manejar los nodos restantes, que agotan su batería rápidamente (Wazed, Bari, & Jackel, 2007)

Sierra (2009) indica que un cromosoma es representado como una cadena de números de nodos para un esquema de enrutamiento en particular, y la longitud del mismo es igual al número de nodos de relevo. El nodo en el cromosoma que disipa el máximo de energía es marcado como un nodo crítico, y los mensajes dirigidos a él son redirigidos hacia algún otro nodo elegido al azar. Mediante este método se observa en el artículo una extensión de un 200% del tiempo de vida de la red en comparación con otras redes que emplean modelos de mínima energía de transmisión y mínima cantidad de saltos (Sierra, 2009).

Sin embargo, las simulaciones se realizan sobre una aplicación en Java, por lo que no se puede considerar la influencia de la subcapa MAC en el consumo de energía de los nodos.

2.7.4 Algoritmos basados en Inteligencia de Enjambres

La inteligencia de enjambres se originó del estudio del comportamiento colectivo de sociedades de especies biológicas tales como bandadas de pájaros, cardúmenes de peces y colonias de hormigas. En esencia, es la propiedad de un sistema donde los comportamientos colectivos de agentes interactuando localmente con su medio ambiente causan la emergencia de patrones globales.

Un algoritmo de enrutamiento basado en inteligencia de enjambres es ACO-QoS (*Ants Colony Optimization based QoS Routing algorithm*), que utiliza un enfoque del algoritmo de optimización de colonias de hormigas, modificado para resolver problemas de enrutamiento. Toma en cuenta el retardo, las limitantes de energía, y el ancho de banda de la comunicación inalámbrica. ACO-QoS busca las mejores rutas que satisfagan los requerimientos mínimos de QoS y el equilibrio entre el cuidado de sus restricciones y la complejidad computacional (Syed, Piyush, & Tahsin, 2016).

Cuando un nodo origen ha capturado información del medio, ésta debe ser enviada a la estación base. Primero, el nodo revisa su tabla de enrutamiento para encontrar la ruta apropiada. La fase de prueba de una nueva ruta se realiza sólo si no existen rutas disponibles desde el nodo actual hasta la estación base, mientras que el nodo requiere mantener la información guardada en memoria hasta que se pueda transmitir. 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 (Dominguez, 2011, pág. 20).

Un algoritmo basado en la meta-heurística de optimización basada en la colonia de hormigas para el enrutamiento de datos en redes de sensores inalámbricos configurados en una topología de malla desarrollado por Hernández (2011) mediante el uso de una regla que comprende la actualización de feromonas, el algoritmo de colonia de hormigas encuentra rutas de bajo consumo de energía, aumentando así la vida útil de la red.

Ambos algoritmos obtienen resultados superiores en cuanto al tiempo de vida de la red en comparación con algoritmos que utilizan como único parámetro la distancia

en saltos hacia el sumidero, pero no se realizan comparaciones con protocolos de enrutamiento convencionales. (Hernández, 2011)

2.7.5 Algoritmos basados en Aprendizaje por Refuerzos

Muchas investigaciones se han realizado en el campo de los algoritmos de enrutamiento conscientes de la energía y de los basados en aprendizaje por refuerzos. A pesar de que varios de ellos no son específicos para redes de sensores inalámbricos, existen múltiples protocolos que pueden ser beneficiosos en este campo (Kulkarni, Forster, A, & Kumar, G., 2011)

Uno de los principales algoritmos de enrutamiento basados en aprendizaje por refuerzos es Q-Routing, está basado en el esquema Q-Learning, una de las ramas del aprendizaje por refuerzos. Es creado con el objetivo de aprender una política de enrutamiento capaz de balancear la tarea de minimizar el número de saltos que toma un paquete con la posibilidad de congestión en las rutas más utilizadas (Kaur & Singh, 2013).

Para ello, experimenta diferentes políticas de enrutamiento y guarda estadísticas para determinar qué decisiones son las más adecuadas para minimizar el tiempo total de envío. Mediante este esquema, cada nodo es capaz de decidir a cuál vecino encaminar paquetes para minimizar el tiempo promedio de envío, basado solamente en información local obtenida mediante realimentación.

El tiempo total que toma un paquete para alcanzar su destino está determinado por la demora total de transmisión en enlaces intermediarios y el tiempo total de espera en cola en los nodos del trayecto. La información de enrutamiento es almacenada en cada nodo en una tabla de valores Q como el tiempo de trayecto estimado desde cada nodo vecino hasta el nodo de destino. (Heinzelman, Kulic, & Balakrishnan, 2009) y (Garbarino, 2011).

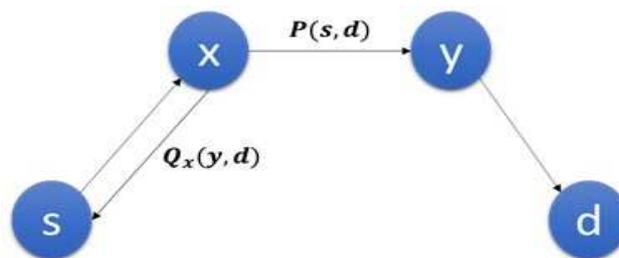


Figura 2. 11 Red con proceso de envío de un paquete en Q-Routing.

Fuente: (Garbarino, 2011)

Cuando un paquete es enviado, el nodo receptor envía a la fuente un estimado del tiempo restante de trayecto para el paquete en cuestión. Este estimado es interpretado por el nodo fuente como una señal de refuerzo para actualizar el valor Q correspondiente. Después de un período de aprendizaje inicial, los valores Q convergen a los estimados reales, y el estado de red almacenado representará de manera más precisa el estado real de la red, resultando en un mejor rendimiento.

Cada nodo x mantiene una tabla de valores $Q_x(y, d)$, donde $y \in N(x)$ es el conjunto de vecinos de x , y $d \in V$, el conjunto de nodos en la red. Cada valor Q representa el tiempo estimado para que un paquete alcance el destino d desde el nodo y , excluyendo el tiempo de espera en la cola de y . Si los valores Q representan con exactitud el estado actual de la red, la mejor opción es enviar el paquete al vecino con el menor tiempo de entrega estimado para el destino d , esto es lo que se denomina una política local ávida. (Heinzelman, Kulic, & Balakrishnan, 2009).

La figura 2.11, muestra el esquema de encaminamiento utilizado por Q-Routing. Cuando un nodo x recibe un paquete $P(s, d)$ con fuente s y destino d , selecciona el vecino y del vector de valores Q $Q_x(*, d)$, para el cual $Q_x(y, d)$ es el valor mínimo. Posteriormente, el paquete es enviado a y , que inmediatamente responde a x con su mejor estimado τ , que representa el tiempo de trayecto estimado para que un paquete curse desde y hasta el destino d . El cálculo de τ se describe por la ecuación 1.1, donde q_y es el tiempo de espera del paquete en la cola de y .

$$\tau = \min_{z \in \text{vecinos de } y} Q_y(z, d) + q_y \quad (2.1)$$

El nodo x tendrá en cuenta la variable τ durante sus actualizaciones. Cuando un nodo actualiza su tabla local de estimados, también toma en cuenta el tiempo de transmisión del paquete entre los nodos x y y (indicador de la calidad del enlace), representado por s . Posteriormente, se compara el nuevo estimado con el antiguo, y se aplica un factor de aprendizaje η_f a la diferencia (ecuación 2.2). Esto resulta en un valor delta que será usado por x para actualizar el estimado local para el

vecino y de acuerdo a la regla de actualización representada por la ecuación 2.3. Este tipo de exploración donde el nodo que envía el paquete recibe un estimado de su vecino se denomina exploración hacia adelante, y es utilizado por varios de los algoritmos de enrutamiento inspirados en *Q-Routing*. (Heinzelman, Kulic, & Balakrishnan, 2009).

$$\Delta Q_x(y, d) = n_f(s + \tau - Q_x(y, d)) \quad (2.2)$$

$$Q_x(y, d) = Q_x(y, d) + \Delta Q_x(y, d) \quad (2.3)$$

Q-Routing ha servido de inspiración para el desarrollo de otros protocolos de enrutamiento basados en aprendizaje por refuerzo. Tales son los casos de *DRQ-Routing*, *CQ-Routing*, *CDRQ-Routing* y *FROMS*.

2.7.6 FROMS (*Feedback Routing to Multiple Sinks*)

Este protocolo de enrutamiento multicast denominado *FROMS* explota las potencialidades del aprendizaje por refuerzos. Dicho protocolo es adecuado para cualquier aplicación de envío periódico de datos hacia uno o varios sumideros en un ambiente multi-salto. Sus principales ventajas son:

- Capacidad de encontrar rutas multicast óptimas.
- Incorpora diferentes métricas de costo, lo que lo hace adaptable a diferentes metas de optimización.
- Rápida recuperación ante fallas. (Förster & Murphy, 2007)

La principal meta de *FROMS*, según sus creadores, es proveer al desarrollador de una WSN de una única solución de enrutamiento, capaz de configurarse para diferentes escenarios de aplicación. Este es el protocolo de enrutamiento inteligente elegido para su estudio en el presente trabajo, ya que provee un esquema de encaminamiento consciente de la energía.

Cada modelo de IC tiene criterios de convergencia y requerimientos computacionales particulares. Asimismo, no todos pueden ofrecer soluciones de forma distribuida, que es lo ideal en el enrutamiento para redes de sensores inalámbricos. Otro factor que los diferencia es el tráfico generado para el descubrimiento y mantenimiento de rutas. A continuación, se ofrece una breve

comparación de los paradigmas de IC de acuerdo a los requerimientos para su uso en una WSN. (Noorani, 2012).

2.7.7 Comparación entre las técnicas de IC

Las redes neuronales y algoritmos evolutivos tienen altas demandas de procesamiento y son generalmente soluciones centralizadas. En el caso de las redes neuronales, el aprendizaje se puede hacer en línea en cada uno de los nodos, pero es un proceso lento y con altos requerimientos de memoria. La lógica difusa es bastante adecuada para implementar algoritmos de optimización de rutas, sin embargo, no genera soluciones óptimas y nuevas reglas difusas deben aprenderse ante cambios de topología (Kulkarni, Forster, A, & Kumar, G., 2011)

La inteligencia de enjambres es un paradigma muy popular para computar esquemas de enrutamiento para MANET (*Mobile Ad-hoc Network*). Sin embargo, en las WSN se requiere de un alto *overhead* de comunicación para que las “hormigas artificiales” transiten separadamente para el manejo de rutas. Además, generalmente se necesita que las “hormigas” vuelvan a la fuente de datos, lo que a largo plazo se traduce en un gasto significativo de energía.

Tabla 2. 7 Requerimientos de los diferentes paradigmas de IC

Paradigma	Requerimientos computacionales	Requerimientos de memoria	Flexibilidad
Redes Neuronales	Medio	Medio	Baja
Lógica Difusa	Medio	Medio	Alta
Algoritmos Evolutivos	Medio	Alto	Baja
Inteligencia de Enjambres	Bajo	Medio	Alta
Aprendizaje por refuerzos	Bajo	Medio	Alta

Fuente: (Kulkarni, Forster, A, & Kumar, G., 2011)

El aprendizaje por refuerzos es la mejor opción para lidiar con problemas distribuidos y dinámicos como el enrutamiento y la agrupación de nodos en las WSN. Tiene exactamente las propiedades necesitadas y ha sido aplicado a ambos problemas. El aprendizaje por refuerzos produce decisiones de enrutamiento

óptimas, es robusto y flexible ante fallos de nodos y enlaces, además de ser un algoritmo totalmente distribuido. Sus requerimientos de comunicación son muy bajos y asegura el envío de datos incluso ante cambios en la topología. *Q-Learning* es el método más popular de aprendizaje por refuerzos aplicado a redes de sensores inalámbricos por su flexibilidad, y porque presenta las menores demandas de comunicación y procesamiento (Kulkarni, Forster, A, & Kumar, G., 2011).

2.8 Elección del Simulador

Los simuladores ayudan a los desarrolladores a llevar a cabo experimentos con nuevos algoritmos, protocolos y aplicaciones antes de ser implantados en bancos de pruebas con nodos sensores reales. Algunos de los simuladores son adaptaciones de los simuladores de redes de propósito general con la inclusión de módulos para la simulación de redes de sensores. También existen simuladores para plataformas específicas.

2.8.1 Simuladores de propósito general

Estos simuladores se caracterizan por haber sido diseñados para la simulación de redes tanto cableadas como inalámbricas, para los que se han creado módulos que permiten llevar a cabo la simulación de redes de sensores. Entre los más populares podemos encontrar:

NS-3

Network Simulator 3 (NS-3), es un simulador de eventos discretos desarrollado en C++. Es uno de los simuladores de redes no específicos más populares y soporta una gran cantidad de protocolos para diversos tipos de redes. Este simulador se basa en el paradigma de la reusabilidad y proporciona una buena colección de protocolos de enrutamiento. Incluye un módulo de aplicación llamado *Network Animator* (nam), que proporciona los resultados de forma visual. El objetivo del proyecto NS-3 es el desarrollo de un entorno de simulación preferido, abierto a la investigación de redes: debe estar alineado con las necesidades de simulación de redes de investigación moderna y debería fomentar la contribución de la comunidad, la revisión por pares, y la validación del software. (Network Simulator NS-3, 2015).

OMNeT++

Objective Modular Network Testbed (OMNeT++), es un simulador de eventos discretos implementado en C++. Proporciona una potente GUI (*Graphical User Interface*) para llevar a cabo el seguimiento y el depurado de las aplicaciones. Está basado en una jerarquía anidada de módulos que proporciona modelos de hardware, incluyendo CPU (Central Processing Unit), batería y radio. En esencia, es un software que provee la infraestructura necesaria para implementar simulaciones de sistemas que pueden ser modelados por eventos discretos. (OMNeT++ , 2016)

Existen una serie de *frameworks* y simuladores específicos para llevar a cabo la simulación de redes de sensores, entre los que se encuentran *MiXiM*, *Mobility Framework* y *Castalia*. Cuentan con una amplia comunidad de seguidores que comparte en línea ideas y código fuente. (OMNeT++ , 2016)

2.8.2 Simuladores para plataformas específicas

El creciente interés por las redes de sensores, y la necesidad de realizar pruebas preliminares, previas a la implementación en bancos de pruebas reales, han llevado al desarrollo de simuladores para plataformas específicas. A continuación, se explican 3 tipos de simuladores utilizados en investigaciones afines.

TOSSIM

Es un simulador de eventos discretos que está incluido en TinyOS. Simula la ejecución de código nesC en nodos Mica, permitiendo la emulación de nodos reales mediante el mapeado de las interrupciones hardware en eventos discretos. Los componentes de emulación de hardware son compilados junto con los componentes reales de TinyOS, utilizando el compilador de nesC. De esta forma, se obtiene un programa ejecutable real de TinyOS que es ejecutado sobre una capa física simulada. (Ortiz Torres, 2011), (TinyOS, 2013).

ATEMU

Es un simulador y emulador específico para nodos Mica. Durante su funcionamiento, el modelo de radio es simulado mientras que la operación de los nodos es emulada instrucción por instrucción, permitiendo llevar a cabo pruebas de aplicaciones, protocolos, algoritmos y sistemas operativos, además de la capacidad

para simular redes heterogéneas compuestas por diferentes modelos de nodo (Ortiz Torres, 2011).

La elección del simulador está relacionada con los requerimientos de la investigación que se desea desarrollar. La precisión de los resultados varía de acuerdo al simulador elegido, ya que implementan diferentes modelos de canal de radio, modulación, consumo de energía, etc.

Para el trabajo de titulación se emplea el simulador NS-3 y en base al análisis de algoritmos (códigos en C++) basados en inteligencia computacional, el protocolo seleccionado es FROMS, en el siguiente capítulo se detalla la ejecución de dicho protocolo a través de simulaciones.

CAPÍTULO III: DESEMPEÑO Y PRUEBAS DEL PROTOCOLO FROMS

En el capítulo anterior se caracterizó varios algoritmos basados en Inteligencia Computacional. En el presente capítulo se determina el desempeño de FROMS basado en ensayos de científicos e investigadores, al final de este capítulo se presenta de forma estadística y grafica la evaluación con el abordaje del protocolo de enrutamiento convencional REL, el cual se debe comparar con el protocolo de enrutamiento multicast FROMS, que puede probar la hipótesis del menor consumo energético por el seleccionamiento de dicho algoritmo para el enrutado de datos en redes tipo Zigbee.

3.1 Desempeño del algoritmo FROMS

Para explicar el funcionamiento de FROMS se toma como referencia la red de la figura 3.1, donde existe una fuente de datos S y dos sumideros: P y Q. Un posible camino desde la fuente hacia los sumideros está formado por la unión de los caminos individuales desde la fuente hacia cada uno de los sumideros (representados por líneas puntuadas), sin embargo, un camino más corto puede existir, el cual tiene la forma de un árbol, como aquel formado por los nodos B, F y H.

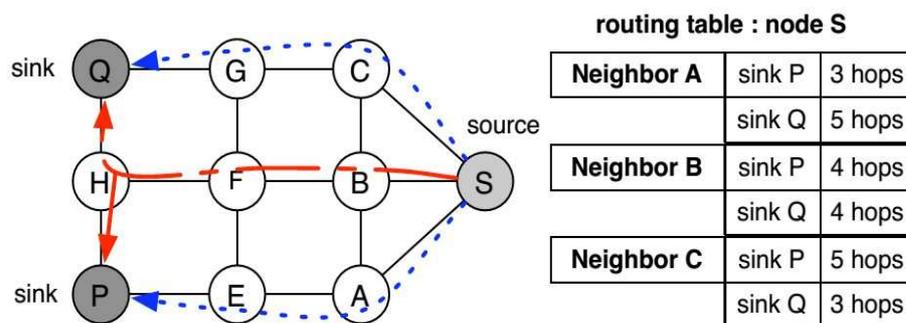


Figura 3. 1 Topología de ejemplo con dos sumideros (*sink*) y una fuente

Fuente: (Förster & Murphy , 2007)

El reto es identificar globalmente este camino sin conocimiento de toda la topología, y usando solamente intercambio de datos locales. Entonces, la tarea se traduce en actualizar la información local relacionada con los próximos saltos para alcanzar los sumideros desde cada nodo, de manera tal que el costo del árbol resultante sea óptimo.

Durante una fase inicial de anuncios del sumidero, todos los nodos reúnen información inicial de enrutamiento y registran los sumideros conocidos de la red. En el ejemplo de la figura 3.1, el nodo S reúne datos de cantidad de saltos para alcanzar cada sumidero individualmente, ser encaminado, éste selecciona uno o más nodos como próximo salto hacia el sumidero.

Sin embargo, en lugar de simplemente seleccionar el más indicado en ese momento (en el ejemplo: nodo C para el sumidero Q y nodo A para el sumidero P), también explora rutas sub-óptimas en asunción de que algunas de ellas puedan tener menores costos que los asociados en su tabla de enrutamiento. La razón detrás de este planteamiento está en que nodos vecinos podrían compartir próximos saltos. Por ejemplo, el nodo fuente S estima que desde el nodo A necesita 7 saltos para alcanzar ambos sumideros: 3 saltos para alcanzar P, 5 saltos para el sumidero Q, y el primer salto, que es compartido, por lo que se descuenta.

Por otra parte, el nodo S no conoce si A será capaz de compartir el próximo salto o necesitará dividir el camino a través de dos vecinos diferentes. En el ejemplo, el nodo A es capaz de compartir el próximo salto, y calcula que puede alcanzar los sumideros a través del nodo E (ver la tabla de enrutamiento de la figura 3.1) en $(2+4) - 1 = 5$ saltos. Por tanto, el nodo S será capaz de alcanzar ambos sumideros en un salto hasta el nodo A, más 5 saltos desde A hasta todos los sumideros, para un total de 6 saltos, que es 1 salto menos que la información inicial en el nodo fuente. De ahí que A necesita informar a S sobre su propio estimado del costo de enrutamiento hacia ambos sumideros.

Para ello, asumiendo que todos sus vecinos pueden escuchar sus transmisiones en el canal inalámbrico, el nodo A hace uso de la difusión cuando va a enviar el paquete de datos hacia el próximo salto, y agrega a cuentas su propio estimado de costo. Similarmente, el nodo E añade a cuentas sobre los datos su estimado de costo e informa al nodo A, y así sucesivamente.

3.1.1 Enrutamiento multicast con Q-Learning

Cada nodo sensor es un agente independiente, y las acciones son opciones de enrutamiento usando diferentes vecinos como próximo salto hacia un subconjunto

de sumideros $D_p \subseteq D$, listado en el paquete de datos.

En el modelo de FROMS, una acción es una decisión de enrutamiento posible para el paquete de datos. Sin embargo, la información de enrutamiento puede incluir varios nodos vecinos como próximos saltos. Consecuentemente, se necesita realizar modificaciones en el algoritmo Q-Learning original, y definir una posible acción a como un conjunto de sub acciones $\{a_1 \dots a_k\}$. Cada sub-acción $a_i = (n_i, D_i)$ incluye un único vecino n_i y un conjunto de destinos $D_i \subseteq D_p$, indicando que el vecino n_i es el próximo salto deseado para encaminar paquetes hacia los destinos D_i . Una acción completa es un conjunto de sub-acciones tales que $\{D_1 \dots D_k\}$ particiona D_p (esto es, cada sumidero $d \in D_p$ es cubierto por exactamente una sub-acción a_i).

Continuando con el ejemplo de la figura 3.1, considérese un paquete destinado a $D_p = \{P, Q\}$. Una posible acción completa de la fuente S es la sencilla sub-acción $(B, \{P, Q\})$, indicando al vecino B como próximo salto para todos los destinos. Alternativamente, el nodo S podría elegir dos sub-acciones: $(A, \{P\})$ y $(C, \{Q\})$, indicando que dos vecinos diferentes deben tomar la responsabilidad de encaminar el paquete a diferentes subconjuntos de vecinos. La distinción entre acciones completas y sub-acciones es importante ya que solamente las sub-acciones reciben recompensa.

Debido a que FROMS está basado en *Q-Routing*, tiene la necesidad de explorar sus vecinos para poder encontrar nuevas rutas con mejores estimados. El algoritmo se apoya en la técnica *ϵ -greedy*, donde cada nodo realiza una acción al azar con una probabilidad ϵ , y selecciona la mejor ruta disponible con probabilidad $1 - \epsilon$. Cuando un nodo selecciona la mejor acción disponible, se dice que explota la información almacenada, mientras que acciones aleatorias resultan en exploración.

3.1.2 Valores Q

En FROMS el procedimiento de inicialización de los valores Q es diferente al de *Q-Learning*, que inicialmente inicializa los valores Q aleatoriamente. En el caso de FROMS, inicialmente representan el costo real de las rutas, por ejemplo, si la función de costo es el número de saltos, el valor Q de una ruta es también el número

de saltos. Para inicializar estos valores, se usa un procedimiento que calcula un estimado del costo basado en la información individual sobre el vecino esperado y los sumideros. Esta inicialización no aleatoria agiliza significativamente el proceso de aprendizaje y evita oscilaciones de los valores Q.

El primer paso es calcular el valor de las sub-acciones, y luego el de una acción completa. El valor inicial para una sub-acción $a_i = (n_i, D_i)$ está determinado por la ecuación 3.1, donde $salto_s_d^{n_i}$ es el número de saltos para alcanzar el destino $d \in D_i$ usando el vecino n_i , y $|D_i|$ es el número de sumideros en D_i .

La primera parte de la ecuación calcula el número total de saltos para alcanzar individualmente los sumideros, y la segunda parte sustrae cierta cantidad de saltos del total, basada en la asunción de que la comunicación por difusión se realiza tanto para transmitir hacia n_i como por n_i para alcanzar al próximo salto. El valor Q de una acción completa a con sub-acciones $\{a_1, \dots, a_k\}$ se calcula mediante la ecuación 3.2, donde k es el número de sub-acciones. Este valor Q es el número de saltos para hacer llegar por difusión un paquete desde el agente hacia todos los sumideros.

$$Q(a_i) = \left(\sum_{d \in D_i} salto_s_d^{n_i} \right) - 2(|D_i| - 1) \quad (3.1)$$

$$Q(a) = \left(\sum_{a_i \in a, i=1 \dots k} Q(a_i) \right) - (k - 1) \quad (3.2)$$

3.1.3 Actualización de valores Q

Para aprender los valores reales de las acciones, el agente debe recibir las recompensas del ambiente. La regla de actualización de valores Q es similar a la usada por el algoritmo Q-Routing (ecuación 3.3), donde $R(a_i)$ es el valor de la recompensa.

$$Q_{nuevo}(a_i) = Q_{viejo}(a_i) + \gamma(R(a_i) - Q_{viejo}(a_i)) \quad (3.3)$$

La recompensa es el valor numérico que permite al nodo en el camino de los datos, informar al nodo anterior de su costo actual para la acción solicitada. Por lo tanto, cuando un nodo calcula la recompensa, selecciona su mejor valor Q en la tabla de

enrutamiento para el conjunto de destinos, y añade el costo de la acción misma. Este procedimiento se describe por la ecuación 3.4, donde c_{a_i} es el costo de la acción y tiene valor 1. Esta propagación de valores Q en el camino contrario a los datos, eventualmente permite que todos los nodos aprendan los costos reales de las acciones.

$$R(a_i) = c_{a_i} + \min_a Q(a) \quad (3.4)$$

3.1.4 Enrutamiento consciente de la energía con FROMS

Toda técnica de encaminamiento propuesta a trabajar en una WSN debe ser consciente en cuanto a los gastos energéticos, esto es, disminuir los gastos de comunicación y tomar medidas que favorezcan el tiempo de vida de la red. Con la métrica de costo combinada de saltos y energía residual de las baterías, se pretende balancear los gastos energéticos en una red de sensores inalámbricos.

En este caso, los valores Q son funciones de ambas métricas, como indica la ecuación 3.5, donde E_{saltos} es el estimado de costo en saltos de la ruta y $E_{batería}$ es el costo estimado de batería de la ruta, que se define como la mínima energía residual de todos los nodos a través de la misma (ecuación 3.6).

$$Q_{comb}(ruta) = f(E_{saltos}, E_{batería}) \quad (3.5)$$

$$E_{batería}(ruta) = \min_{n_i \in ruta} batería \quad (3.6)$$

$$f(E_{saltos}, E_{batería}) = hcm(E_{batería}) * E_{saltos} \quad (3.7)$$

La función f que combina ambos estimados en un solo valor Q, se define por la ecuación 3.7, donde hcm (*hop count multiplier*) es el multiplicador del número de saltos, y es generado por una función que pesa el estimado de número de saltos en base a la energía residual de la batería del vecino. Mientras menor sea este valor, mayor será el multiplicador. Esto se hace con el objetivo de desalentar futuros usos de rutas que involucran nodos con un bajo nivel de energía residual, y extender así el tiempo de vida de la red.

La figura 3.2 muestra cuatro funciones de costo diferentes (también conocidas

como funciones de ponderación). Si el nivel de batería es irrelevante, entonces hcm ($E_{batería}$) es constante y $f(E_{saltos}, E_{batería})$ se reduce a una función basada en saltos solamente, donde $hcm = 1$.

Por el otro lado, si se desea encontrar un balance entre la distancia en saltos y el consumo de energía, se utiliza una función de costo dinámica. Se consideran tres funciones de este tipo: dos lineales (ecuaciones 3.8 y 3.9) y una exponencial (ecuación 3.10). Cada una de ellas genera un rango de multiplicadores diferentes, por lo que el criterio de elección depende de los requerimientos de la aplicación. Los valores de $energía_disponible$ se encuentran en el rango de 0 y 1, con $energía_máxima = 1$.

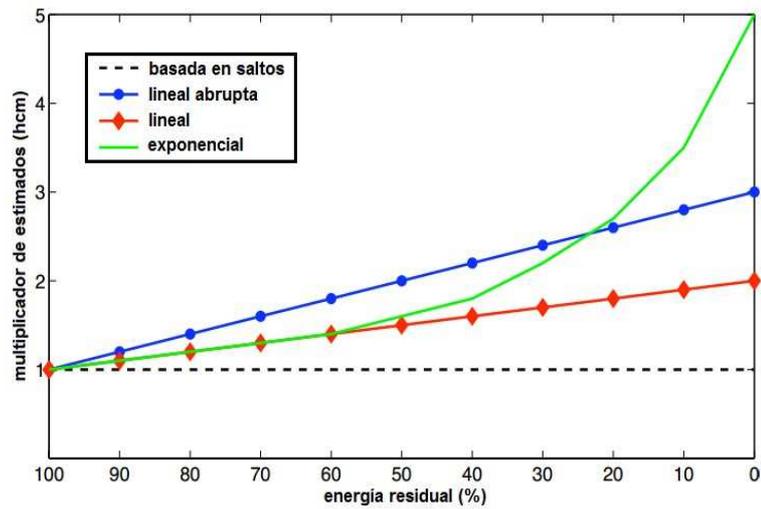


Figura 3. 2 Multiplicadores de estimados para diferentes funciones de costo

Fuente: (Förster & Murphy , 2007)

$$W_1(energía_disponible) = (2 * energía_máxima) - energía_disponible \quad (3.8)$$

$$W_2(energía_disponible) = (3 * energía_máxima) - energía_disponible \quad (3.9)$$

$$W_3(energía_disponible) = 5^{(energía_máxima - energía_disponible)} \quad (3.10)$$

3.1.5 Manejo de fallas

Cada nodo mantiene una estructura de datos llamada SinkControl, cuyo objetivo es detectar enlaces caídos. Dicha estructura almacena información sobre cada

sumidero conocido por el nodo. La tabla 3.1 representa un ejemplo de *Sink Control* para el nodo E de la red de ejemplo de la figura 3.1.

En la realimentación se incluye una estampilla de tiempo sobre la última vez que el nodo ha recibido información sobre el sumidero. Si esta estampilla es demasiado vieja (excede un determinado umbral), se invalidan las rutas hacia el sumidero. Esto sucede cuando el sumidero ha caído, o cuando fallan los enlaces en un punto determinado de la red, y no se pueden encaminar los datos hacia su destino.

Para adaptarse a fallas en los enlaces, cada nodo almacena la última vez que escuchó a cualquier vecino transmitir un paquete. Adicionalmente, guarda la última vez que encaminó un paquete hacia ese vecino. En caso de que la diferencia entre ambas estampillas exceda un umbral determinado, el vecino es eliminado de la tabla. Si esto sucede por error, la próxima vez que el nodo escuche a este vecino recuperará la ruta. A diferencia de algunos protocolos que usan manejo de vecinos, FROMS no se auxilia de balizas o difusiones periódicas. En su lugar, se auxilia de la escucha de los paquetes de datos transmitidos por los vecinos para chequear la disponibilidad de los mismos.

Tabla 3. 1 Estructura *Sink Control* para el nodo E

(Sink) Sumidero	Última marca de tiempo	Vecino directo	Marca de tiempo
Sumidero 1	-2 seg	Verdadero	-2 seg.
Sumidero 2	-14 seg	Falso	-

Nota: Control de sumidero: nodo 7.

Fuente: (Förster & Murphy , 2007)

Para el estudio de FROMS es necesario hacer una comparación con un protocolo de enrutamiento convencional. Se elige al protocolo REL, ya que también incorpora realimentación sobre la energía residual de los nodos y utiliza reglas que permiten balancear el consumo de energía.

3.2 REL (Routing by Energy and Link Quality)

Es un protocolo de enrutamiento para redes planas cuyo objetivo es incrementar la confiabilidad y eficiencia en el consumo de energía en una red de sensores inalámbricos.

REL selecciona rutas en base a un mecanismo de estimación de la calidad del enlace de extremo a extremo, energía residual y cantidad de saltos. Además, propone un mecanismo manejado por eventos para la distribución de la carga y evitar el agotamiento prematuro de la energía de los nodos. (Machado, y otros, 2013)

3.2.1 Estimación de la calidad del enlace

Para analizar un enlace, REL se apoya del LQI (Link Quality Indicator), que es una métrica proveída por la capa física del estándar IEEE 802.15.4. LQI varía desde 0 hasta 255, y es calculada en base a la RSSI (Received Signal Strength Indication), SNR (Signal-NoiseRatio) o una combinación de ambas métricas. Sin embargo, el valor de LQI solamente refleja la calidad del enlace para los nodos vecinos, no de extremo a extremo. Para determinar la calidad del enlace hacia el destino, REL propone el uso de WeakLinks, que no es más que un contador de saltos para los peores enlaces a través de un camino determinado. En términos específicos, si el LQI para un determinado enlace es menor que LQI_{th} (LQI threshold), el enlace se considera débil y el contador WeakLinks es incrementado.

WeakLinks es incorporado en los mensajes RREQ y RREP, y posteriormente actualizado en cada salto durante el proceso de descubrimiento de rutas. Cada vez que un nodo recibe un mensaje RREQ o RREP, actualiza su valor de LQI. Seguidamente, debe determinar si dicho valor de LQI es menor que LQI_{th} , y actualizar *WeakLinks* si es necesario.

Asuma que $LQI_{th} = 170$ en la red de ejemplo de la figura 3.3, donde la fuente es el nodo S y el destino el nodo D. Un proceso de selección de rutas que elige el camino de menor distancia en saltos seleccionará al nodo C como próximo salto. Sin embargo, este nodo posee enlaces de baja calidad, y es más susceptible a causar pérdidas de paquetes que los otros enlaces.

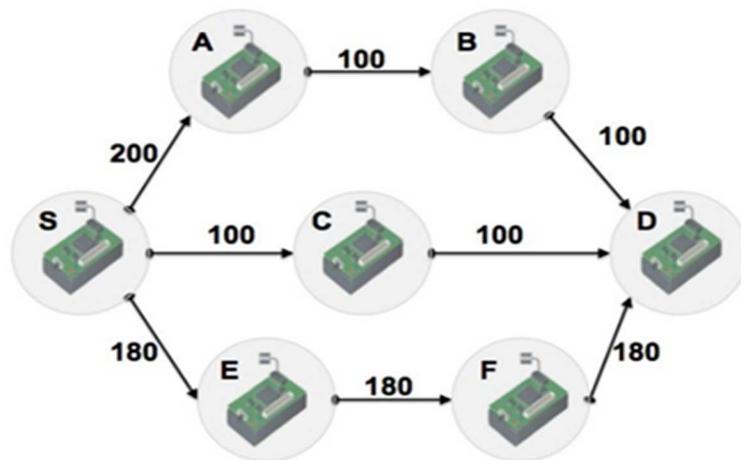


Figura 3. 3 Estimación de la calidad del enlace extremo a extremo

Fuente: (Machado, y otros, 2013)

Por otro lado, un mecanismo de selección de rutas que use solamente la calidad del enlace como métrica, seleccionará al nodo A como el próximo salto, aunque existen caminos poco confiables en el camino restante. Finalmente, cuando se utiliza WeakLinks como métrica, puede determinarse que los nodos A y B poseen 2 enlaces débiles cada uno, y que el nodo E no los posee. Por lo tanto, cuando los enlaces son analizados y seleccionados desde la perspectiva de la calidad extremo a extremo, el camino más confiable es a través del nodo E. Esta ruta es capaz de proporcionar mayor confiabilidad para la entrega de datos.

3.2.2 Selección de rutas y balanceo de cargas

REL explota un esquema reactivo para encontrar rutas bajo demanda con el objetivo de reducir los gastos de señalización. El proceso de descubrimiento de rutas incluye la difusión de mensajes RREQ y RREP. Estos mensajes buscan rutas disponibles y asisten al proceso de selección de rutas mediante la recolección de información sobre la energía residual y la calidad del enlace. Cada mensaje RREP representa una ruta disponible hacia el destino, y de acuerdo a la configuración de REL, es posible almacenar n rutas posibles hacia cada nodo de destino. REL integra/administra los valores de tres métricas claves para encontrar las mejores rutas:

- Calidad de los enlaces inalámbricos en base a WeakLinks.

- Energía residual de la batería.
- Distancia en saltos, para evitar caminos largos e ineficientes.

El proceso de selección de rutas depende de dos umbrales para establecer comparaciones entre las posibles rutas. El primero de ellos es el umbral de conteo de saltos $HCdiff_{\max_allow}$ (*Hop Count Maximum Difference*), que determina cuál es la máxima diferencia en saltos para determinada ruta. El segundo umbral es E_{th} (*Energy Threshold*), un umbral de energía residual que es utilizado en dos fases: en el proceso de selección de rutas y en el mecanismo de balanceo de cargas.

En el caso del mecanismo de balanceo de cargas, E_{th} responde al seguimiento de los niveles de energía observados en cada nodo individualmente. Cuando la red inicia su ejecución (fase de *bootstrap*), cada nodo debe almacenar su porcentaje de energía residual y, cada t unidades de tiempo, debe comparar el nivel de energía actual E_t con el que había sido guardado previamente (E_{t-1}). Si la diferencia entre ambos valores es mayor que E_{th} , se activa un evento en respuesta.

La diferencia entre valores de energía responde al nombre de Ind_{RADV} (*index of Route Advisor*). Si $Ind_{RADV} > E_{th}$, se envía un mensaje de notificación RADV (*Route Advisor*) hacia los nodos vecinos. Este mensaje contiene información sobre el nuevo valor de energía residual del nodo en cuestión, y advierte a los nodos vecinos que deben evaluar nuevamente el uso de dicho nodo en sus rutas.

La figura 3.4 muestra el algoritmo de selección de rutas de REL, donde existen tres reglas básicas, las cuales filtran las rutas de acuerdo a las sentencias condicionales de las líneas 3, 9 y 15, que evalúan la ruta candidata R_b y la categorizan de acuerdo a su nivel de energía, comparándola con la ruta activa R_a .

La línea 15 muestra el uso de E_{th} como un parámetro de tolerancia para la diferencia que es aceptable si R_b tiene menor energía. Luego de analizar el nivel de energía, el algoritmo calcula el número de saltos y evalúa la calidad de los enlaces. Este paso ocurre en las líneas 5, 10 y 16, donde el umbral utilizado es $HCdiff_{\max_allow}$. Si ambos casos son ciertos, entonces se conmuta hacia la ruta R_b (líneas 6 y 12).

```

1: Considere  $R_a$  = ruta activa
2: Considere  $R_b$  = ruta alternativa
3: if  $R_a.energía = R_b.energía$  then
4:   if  $R_a.saltos > R_b.saltos + HCdiff_{max\_allow}$  then
5:     if  $R_a.weakLinks \geq R_b.weakLinks$  then
6:       SeleccionarRuta( $R_b$ )
7:     end if
8:   end if
9: else if  $R_a.energía < R_b.energía$  then
10:  if  $R_a.saltos + HCdiff_{max\_allow} \geq R_b.saltos$  then
11:    if  $R_a.weakLinks \geq R_b.weakLinks$  then
12:      SeleccionarRuta( $R_b$ )
13:    end if
14:  end if
15: else if  $R_a.energía > R_b.energía$  and  $R_a.energía \leq R_b.energía + E_{th}$  then
16:  if  $R_a.weakLinks \geq R_b.weakLinks$  and  $R_a.saltos > R_b.saltos + HCdiff_{max\_allow}$  then
17:    SeleccionarRuta( $R_b$ )
18:  end if
19: end if

```

Figura 3. 4 Algoritmo de selección de rutas de REL

Fuente: (Machado, y otros, 2013)

La regla descrita en la línea 15 es un caso especial en el proceso de selección, ya que analiza los casos en los que R_b tiene mejor energía que la ruta activa. En este caso, R_b debe reemplazar a R_a si la diferencia con respecto al umbral E_{th} y R_b es considerablemente pequeña (línea 16).

El análisis del desempeño de REL en comparación con AODV fue realizado en el simulador NS-3. Los resultados de las simulaciones muestran que REL incrementa el tiempo de vida de la red (en función del tiempo de caída del primer nodo) en hasta un 26.6% en comparación con AODV (Machado, y otros, 2013)

Para la comparación de FROMS contra REL en cuanto al tiempo de vida de la red, se implementan ambos en un simulador. Como se comentó en el capítulo anterior, la elección del simulador está determinada por las necesidades específicas de la investigación. Consecuentemente, para la evaluación de la capacidad de un protocolo de enrutamiento de incrementar la vida de la red, es necesario que el simulador considere los gastos energéticos de comunicación.

Para la implementación de los protocolos se utiliza NS-3, que posee un modelo de consumo de energía realístico basado en el estado y potencia de transmisión de transceptores reales. Además, está disponible el código fuente escrito por los desarrolladores de REL.

3.3 NS-3

Es un simulador para redes de sensores inalámbricos, redes de área personal, y en general, redes de dispositivos de baja potencia. Puede ser utilizado por investigadores y desarrolladores que desean probar sus algoritmos distribuidos y/o protocolos en canales inalámbricos y modelos de radio realísticos. También puede ser usado para evaluar plataformas en aplicaciones específicas, ya que es altamente paramétrico y puede simular un amplio rango de plataformas. (Boulis, 2011)

Sus principales características son:

- Modelo avanzado del canal basado en datos obtenidos empíricamente.
- El modelo define un mapa de las pérdidas del trayecto, en vez de conexiones simples entre los nodos.
- Soporta movilidad de los nodos.
- Modelo avanzado de radio basado en sistemas reales para comunicaciones de baja potencia.
- Múltiples niveles de potencia de transmisión.
- Estados con diferentes consumos de potencia y retardos en la conmutación entre estados.
- Ruido del dispositivo sensor, rango de error y consumo de potencia.
- Protocolos de enrutamiento y MAC disponibles.
- Diseñado para la adaptación y expansión.

3.3.1 Estructura básica

Tiene una estructura jerárquica basada en módulos. Cada nodo es un módulo compuesto, cuya estructura se muestra en la figura 3.5. Los nodos no se conectan directamente entre ellos, sino a través del módulo del canal inalámbrico. Por el otro extremo se enlazan al proceso físico, del cual obtienen información sobre alguna magnitud física.

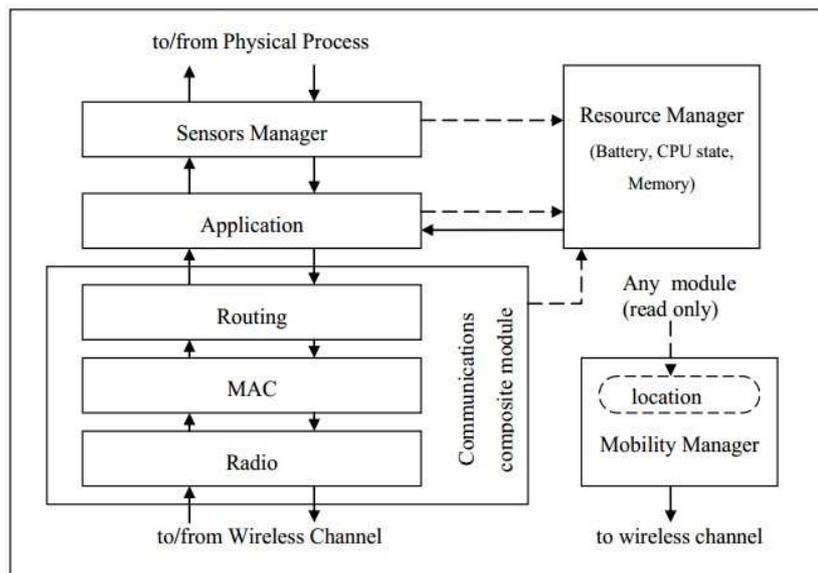


Figura 3. 5 Módulo compuesto para los nodos en NS-3

Fuente: (Boulis, 2011)

El software ofrece, en forma de clases abstractas, las herramientas necesarias para construir nuevos protocolos o aplicaciones. Todos los módulos existentes son configurables a través de varios parámetros.

3.3.3 Programación de módulos

NS-3 provee la infraestructura necesaria para crear nuevos algoritmos, pero ciertas reglas deben seguirse para su integración con el programa de simulación. Los nuevos módulos deben acoplarse a la jerarquía ya existente e implementar las interfaces necesarias. Asimismo, las clases que brindan funcionalidad a los nuevos módulos deben heredar de clases virtuales base que implementan los comportamientos básicos esperados de cualquier módulo de su tipo. Por ejemplo, la clase base de enrutamiento *VirtualRouting* provee funciones para encapsulado/desencapsulado de paquetes, mapeo de direcciones de red a direcciones MAC y manejo de duplicados. De esta manera, se asegura la continua innovación del simulador a través del código propuesto por terceras fuentes, y una adaptación más sencilla del código ante las nuevas versiones del simulador.

Por defecto el simulador no incluye módulos de protocolos de enrutamiento utilizados en redes de sensores.

3.4 Programación del algoritmo FROMS

FROMS es un algoritmo de enrutamiento inteligente y *multicast* para redes de sensores inalámbricos. Sin embargo, en ninguno de los simuladores de código libre encontrados existe la implementación de un protocolo de enrutamiento *multicast* para redes de sensores, por lo que en las simulaciones solamente se utiliza un sumidero o nodo destino. Esto trae como consecuencia algunas consideraciones previas a la implementación del módulo.

Dado que existe un único sumidero en la red, no es necesario dividir los procedimientos de toma de decisiones en acciones y sub-acciones. Entonces, partiendo de la ecuación 3.1 y sustituyendo $|D_i| = 1$ se obtiene que:

$$Q(a_i) = \left(\sum_{d \in D_i} saltos_d^{n_i} \right) \quad (3.11)$$

Esto significa que, si un vecino indica que puede acceder al sumidero en n saltos, entonces el valor Q inicial para ese vecino será precisamente n . Consecuentemente, las tablas de enrutamiento se reducen, ya que solamente necesitan almacenar información relativa a cada vecino.

La implementación de FROMS consiste en la definición de un nuevo módulo de enrutamiento que modele el comportamiento del protocolo. Para ello, se debe definir la tabla de enrutamiento, los formatos de paquetes, y el algoritmo de enrutamiento. La figura 3.6 muestra la relación entre el módulo principal *FromsRouting* y sus clases auxiliares.

3.4.1 Implementación del módulo de enrutamiento

El módulo de enrutamiento que modela el algoritmo FROMS se denomina *FromsRouting*. El primer paso para la codificación es definir la estructura del módulo simple en lenguaje de descripción NED (*Network Description Language*). Los parámetros del módulo son utilizados para inicializar ciertos campos críticos en el código C++. Adicionalmente se definen compuertas para el intercambio de mensajes con los módulos de aplicación y MAC. La estructura general de *FromsRouting.ned* se presenta en la figura 3.7.



Figura 3. 6 Relación entre las clases programadas

Fuente: Elaboración propia

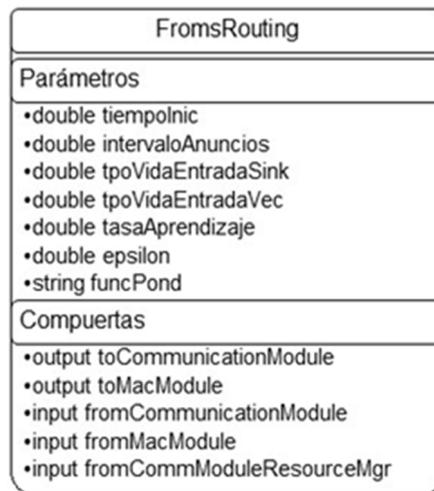


Figura 3. 7 Estructura del módulo *Froms Routing* en NS-3.

Fuente: (Datta & Soundaralaskshmi, 2010)

El parámetro tiempoInic representa el tiempo de inicialización del módulo de red, esto es, el instante de tiempo en el que el sumidero anuncia su presencia a través de un *SINK_ANNOUNCE*, que permitirá a los nodos recolectar información de enrutamiento inicial. Adicionalmente, con el parámetro *intervaloAnuncios*, se indica el intervalo de tiempo entre transmisiones de mensajes *SINK_ANNOUNCE* desde el sumidero. La transmisión periódica de estos mensajes es obligatoria para el manejo de fallas, ya que FROMS no implementa ningún mecanismo de manejo de vecinos mediante mensajes de control.

Los parámetros *tpoVidaEntradaSink* y *tpoVidaEntradaVec* establecen un tiempo de vida máximo para las entradas del sumidero y de los vecinos en la tabla de

enrutamiento, respectivamente. Una vez expirado ese tiempo, se invalidan las entradas correspondientes en la tabla.

El tiempo de vida de la entrada de un vecino se incrementa en *tpoVidaEntradaVec* milisegundos cada vez que se recibe información actualizada del vecino. De manera similar, cada vez que se recibe un mensaje *SINK_ANNOUNCE*, se incrementa en *tpoVidaEntradaSink* segundos el tiempo de vida de las rutas hacia el sumidero, así como la entrada del vecino que entregó el mensaje al nodo. Más adelante, en la definición del comportamiento del módulo en lenguaje C++ se explica con más detalles cómo mediante el uso de temporizadores se maneja la validez de las entradas.

Los parámetros; tasa Aprendizaje y *epsilon* corresponden a los parámetros de aprendizaje. La elección adecuada de estos valores es crítica para obtener el mejor desempeño por parte del algoritmo inteligente. Por último, *funcPond* corresponde a la función ponderadora utilizada por FROMS para degradar los estimados recibidos por los vecinos de acuerdo a la energía residual.

El archivo *FromsRouting.ned* describe el módulo de enrutamiento, pero no implementa ninguna funcionalidad. Los parámetros se definen con el objetivo de ser accesibles desde el archivo de configuración de la simulación (*omnetpp.ini*). Por su parte, las compuertas definen el aspecto del módulo, es decir, las interfaces de entrada/salida para el intercambio de mensajes con otros módulos. Es necesario crear un código en C++ (archivos *.h* y *.cc*) responsable de dar vida al módulo de enrutamiento. Este código será el encargado de procesar los mensajes recibidos de otros módulos e implementar el comportamiento del algoritmo FROMS.

3.4.2 Clase Tabla Enrutamiento

La tabla de enrutamiento del módulo es pequeña, ya que solamente necesita acumular información relacionada con los vecinos del nodo. Es en esencia una lista de objetos de tipo Entrada, una estructura contenida en el archivo de cabecera, y cuya composición se presenta en la figura 3.8.

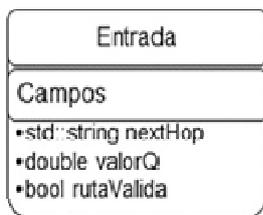


Figura 3. 8 Estructura Entrada

Fuente: (Datta & Soundaralaskshmi, 2010)

Las entradas están compuestas por tres elementos. El primero de ellos es la dirección del próximo salto, que es alguno de los vecinos del nodo, el segundo elemento es el *valorQ* asociado al nodo vecino, y el tercero es un indicador de la validez de la ruta. Cada vez que un nodo recibe un paquete, usa los datos de realimentación para calcular el nuevo estimado a partir del estimado actual. Posteriormente, el estimado calculado se introduce en la tabla de enrutamiento, y automáticamente se valida la entrada.

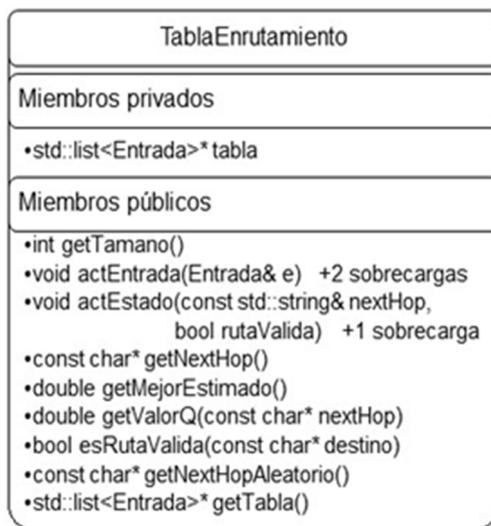


Figura 3. 9 Estructura de la clase Tabla Enrutamiento

Fuente: (Datta & Soundaralaskshmi, 2010)

La clase *TablaEnrutamiento* provee los mecanismos necesarios para manipular la tabla de enrutamiento utilizada por el algoritmo. El cuerpo de la clase provee los métodos que permiten manipular los elementos de la tabla de enrutamiento, su composición se puede observar en la figura 3.9. Los métodos son públicos para que sean visibles por el módulo principal, que debe acceder a la tabla y modificar sus campos cuando sea necesario.

3.4.3 Formatos de paquetes

FROMS maneja dos tipos de mensajes: los mensajes de anuncios y los mensajes de datos, cada uno de ellos con un formato diferente. Para representar ambos tipos de mensajes se crean dos archivos de definición de mensajes (.msg): *FromsRoutingSink AnnouncePkt* y *FromsRoutingDataPkt*. Las figuras 3.10 y 3.12 muestran la estructura de cada uno de los paquetes. Los campos de dirección de origen, dirección de destino, y número de secuencia se omiten ya que son heredados de la clase base *RoutingPacket*.

Paquetes del tipo *FromsRoutingSinkAnnouncePkt* se envían periódicamente desde el sumidero para anunciar su presencia. La cadena *dirSink* identifica la dirección del sumidero, que permite a los nodos de la red identificar el destino de los reportes periódicos. El campo *cantSaltos* contiene la cantidad de nodos por los que transita el paquete. Este valor se incrementa cada vez que un nodo encamina el paquete. De esta manera, el próximo salto es capaz de conocer la distancia en saltos hasta el sumidero desde su ubicación y determinar el mejor vecino para encaminar el próximo paquete de datos.

Cada nodo, antes de reenviar el *SINK_ANNUNCE*, actualiza el campo *energResidual*. La secuencia de valores agrupados conocida en programación como tupla (*cantSaltos*, *energResidual*) permite al próximo salto degradar el estimado recibido de acuerdo a la energía residual del nodo, dando prioridad a rutas que involucren a nodos de mayor capacidad energética. El campo *sinkTimeStamp* contiene la estampilla de tiempo, cuya función se explica más adelante.

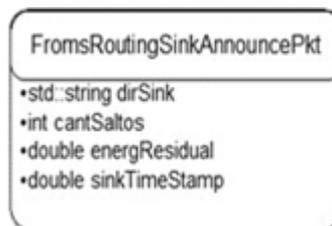


Figura 3. 10 Formato del paquete de anuncios del sumidero

Fuente: (Datta & Soundaralaskshmi, 2010)

Los paquetes de tipo *FromsRoutingDataPkt* son, como su nombre indica, los paquetes de datos del algoritmo FROMS. La cadena *lastHop* especifica la dirección del último nodo por el que transitó el paquete. Este campo, en conjunto con *valorQ*

y *energResidual*, permiten a cada nodo actualizar los estimados para cada uno de los vecinos de acuerdo a la regla de actualización del algoritmo.

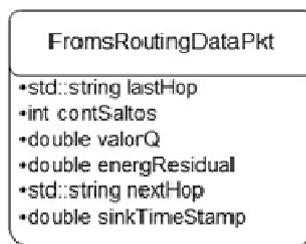


Figura 3. 11 Formato del paquete de datos.

Fuente: (Datta & Soundaralaskshmi, 2010)

Cada vez que un nodo encamina un paquete de datos, reemplaza el valor de *lastHop* por el de su dirección de red. Acto seguido, busca en su tabla de enrutamiento la mejor entrada válida hacia el destino, actualiza los campos de realimentación del paquete y estampilla de tiempo, incrementa el contador de saltos, y lo envía al módulo MAC para su difusión. El campo *sinkTimeStamp* almacena una estampilla de tiempo sobre la última vez que el nodo recibió información del sumidero o destino.

Los archivos .msg que definen los formatos de paquetes no son utilizados por el programa de simulación. En cambio, son traducidos en tiempo de compilación a clases de C++ que implementarán las funciones de acceso a miembros necesarias.

Las clases *TablaEnrutamiento* y las que definen los formatos de paquete son clases auxiliares ya que realmente no implementan ningún comportamiento específico del algoritmo de enrutamiento. Sin embargo, su implementación es necesaria para que el código que define el módulo sea lo más legible posible.

3.4.4 Implementación del comportamiento del módulo

NS-3 tiene una clase en C++ que implementa su comportamiento. Por convención, tiene el mismo nombre que el módulo que describe, en este caso el nombre de la clase es *Froms Routing*. La figura 3.12 muestra su estructura.

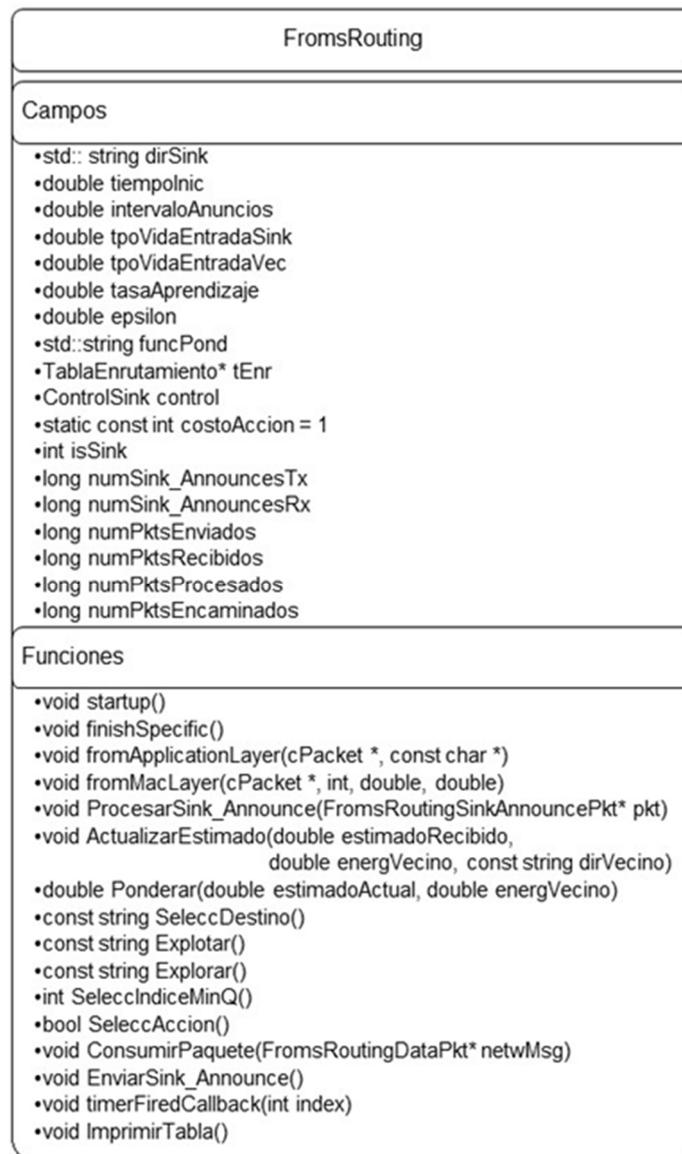


Figura 3. 12 Estructura de la clase *Froms Routing*

Fuente: (Datta & Soundaralaskshmi, 2010)

La cadena *dirSink* almacena la dirección del sumidero. Este campo es necesario debido a que las tablas de enrutamiento reducidas están codificadas por próximo salto, y como consecuencia, no contienen información sobre el destino final. Cada vez que el nodo reciba un anuncio del sumidero, actualizará este valor para confirmar la presencia del mismo.

Para determinar el próximo salto para un paquete, el algoritmo puede usar como métrica la distancia en saltos hasta el sumidero, o bien una métrica combinada de saltos y energía residual. Si se usa la métrica simple de saltos, el nodo no necesita

apoyarse de ninguna entidad externa para evaluar los estimados recibidos, puesto que la información necesaria para llenar las tablas de enrutamiento se puede encontrar en los paquetes recibidos. En cambio, si la métrica utilizada es la combinación de saltos y energía residual, los datos sobre la energía de la batería del nodo debe ser accesible.

El modelo del consumo de energía en un nodo sensor puede ser tan preciso y complejo como se quiera, considerado todos los factores de operación (censado, procesamiento, comunicación, etc.). En general, se considera que cada nodo tiene inicialmente un nivel finito de energía E_T en Joules, determinado por el tipo de batería utilizado, y que este nivel se va consumiendo en el censado (E_S), en el procesamiento (E_P), y en la comunicación, considerando tanto la transmisión (E_{TX}) como en la recepción de datos (E_{RX}).

La literatura científica en WSN señala que el nivel de energía será igual tanto en E_{TX} y E_{RX} y considera que no existe una fuente de carga de energía en los nodos, cuando el nivel de energía remanente (E_R) en el mismo es igual a cero, luego de q_t transmisiones y q_r recepciones de paquetes de datos, el nodo se queda sin energía, pasando a estar fuera de servicio.

$$E_R = E_T - (q_t * E_{TX}) + (q_r * E_{RX}) \quad (\text{ecuación 3.12})$$

El módulo encargado de gestionar los recursos (estados de la CPU, RAM, batería) se llama *resourceManager* (ver figura 3.5). Los campos *initialEnergy* y *remainingEnergy* representan la energía inicial y residual de la batería en Joules, respectivamente. Ambos campos son privados, por lo que no son accesibles desde otros módulos. Para acceder a estos datos se necesita modificar la clase *ResourceManager* e implementar al final dos funciones de acceso para ambos campos, como se muestra en la figura 3.13.

```

class ResourceManager: public NS3{
private:
    double baselineNodePower;
    double currentNodePower;
    simtime_t timeOfLastCalculation;
    double periodicEnergyCalculationInterval;
    double initialEnergy;
    :
    /*--- Custom class parameters ---*/
    double remainingEnergy;
    :
public:
    :
    double getInitialEnergy() {return initialEnergy;}
    double getRemainingEnergy() {return remainingEnergy;}
};

```

Figura 3. 13 Fragmento de código de *ResourceManager.h*.

Fuente: (Datta & Soundaralaskshmi, 2010)

Cuando un nodo adjunta información de realimentación a los datos, el indicador de energía que se propaga no es la energía en *Joules* de la batería, sino el porcentaje de energía residual. Esto se hace para adaptarse a los requerimientos de la función ponderadora, que trabaja con indicadores de energía entre 0 y 1. Por tanto, es necesario llamar a ambas funciones de acceso y obtener la división entre la energía residual y la energía inicial. Este valor es luego insertado en el campo *energResidual* del paquete de datos o de anuncios.

FROMS no usa ningún mecanismo de descubrimiento y manejo de vecinos mediante el intercambio de mensajes de control. Cuando un nodo no recibe información de un vecino o del sumidero durante un determinado tiempo, se invalida la entrada correspondiente dentro de la tabla de enrutamiento. Este comportamiento involucra el uso de temporizadores dentro del módulo de enrutamiento, cada uno de ellos asociado a una entrada dentro la tabla de enrutamiento.

Los mensajes son punteros a la clase base de mensajes *cMessage*, y pueden programarse para recibirse en el módulo en un tiempo de simulación deseado mediante una llamada al método *scheduleAt* (*simtime_t t, cMessage *msg*). Si se desea asociar un temporizador a cada entrada de la tabla, basta con crear un arreglo

de punteros a *cMessage* y programar cada uno de ellos para recibirse en el módulo en el tiempo de simulación deseado.

La creación de un temporizador se define mediante una llamada al método *setTimer* (int *index*, simtime_t *time*). El parámetro *index* sirve para identificar unívocamente a cada temporizador creado. El segundo parámetro especifica el instante de tiempo en el que se dispara el evento y se ejecutan las acciones correspondientes.

El principal problema es que este mecanismo de creación de temporizadores dificulta la interrelación entre las entradas en la tabla de enrutamiento y su temporizador asociado. Si a la entrada de un vecino *i* dentro de la tabla de enrutamiento se le asigna un temporizador de índice *i*, entonces existiría un conflicto con los índices de otros temporizadores usados por el módulo. Por ejemplo, el temporizador de envío de anuncios del sumidero tiene índice 1, y entraría en conflicto con el temporizador del vecino con dirección 1 en la tabla de enrutamiento.

Para enfrentar este problema, se define un *offset* igual a 10 para los índices de los temporizadores. El valor 10 es lo suficientemente grande para la definición de nuevos temporizadores en caso de que en el futuro se vaya a reutilizar el código en la definición de un nuevo módulo. Cada vez que un nodo recibe información sobre un vecino *i*, cancelará el temporizador anterior en caso de que exista, y creará uno nuevo con índice $10 + i$. El tiempo de activación es igual al tiempo de simulación actual más el tiempo de vida de las entradas de los vecinos.

Los protocolos de capa de enlace de las WSN no implementan confirmaciones de recepción para las tramas de difusión. Para enfrentar este problema, FROMS implementa un mecanismo de ratificación implícito a nivel de capa de red. Cada vez que un nodo envía un paquete, reserva una copia en *buffer* y espera a escuchar la transmisión del vecino. Si durante un tiempo determinado el nodo no percibe que el vecino ha encaminado el paquete, procede a retransmitir la copia en el *buffer*. Los campos de dirección de origen, número de secuencia y último salto se utilizan para identificar unívocamente a cada paquete transmitido. Este comportamiento también es considerado y se definen temporizadores para esta tarea.

Hasta este momento, se han explicado las consideraciones tomadas y mecanismos necesarios para la implementación de FROMS. El resto de los métodos que se incluyen en la clase son propios del simulador, y sus funciones pueden encontrarse en el manual de usuario disponible en su portal web oficial.

3.5 Modificaciones en el simulador

NS-3 recopila información sobre la energía consumida y la energía residual de las baterías de cada nodo al final de las simulaciones. Sin embargo, cuando se tiene una considerable cantidad de nodos, el análisis de los niveles de energía de cada uno por separado se hace muy tedioso. Otro señalamiento es que no existen datos disponibles sobre el instante de tiempo en que algún nodo ha agotado su energía, ni cuáles son los nodos que llegan a este estado. Para enfrentar estas carencias se deben introducir nuevas funciones dentro del módulo de administración de recursos.

El primer aporte al módulo es la implementación de un histograma de energía residual de las baterías. La información brindada por un histograma es mucho más compacta, y permite determinar con mayor facilidad la distribución de la energía residual entre los nodos de la red. Para la implementación de esta herramienta se agregan dos nuevos parámetros al archivo de descripción del módulo de administración de recursos. El primer parámetro permite ajustar el límite superior en el histograma, mientras que el segundo parámetro determina el número de celdas.

Mediante un apropiado ajuste de ambos parámetros, se puede observar con la precisión deseada la distribución de la energía en la red. Los valores especificados son leídos por el código C++, y la creación del histograma se hace efectiva mediante una llamada al método *declareHistogram* (const char*, double, double, int) con los parámetros deseados.

Para la recopilación de la información sobre el tiempo de caída de los nodos, así como la identificación de los mismos, se introducen nuevas variables dentro del código C++ del módulo de administración de recursos. Cuando un nodo falla porque ha agotado su energía, se almacena su dirección de red y el tiempo de simulación actual. Estos valores posteriormente son recopilados por NS-3 y se presentan al final de la simulación.

Además, se introduce un método de estimación del tiempo de vida de la red, que es indispensable para estimar el tiempo de simulación transcurrido al momento de caída del primer nodo, y así determinar un buen tiempo de simulación para los experimentos. El método original estima el tiempo de caída del primer nodo en días, pero esta escala es demasiado grande para ejecuciones de minutos de duración en tiempo de simulación. Para disminuir el error y obtener un resultado más aproximado se disminuye a minutos la escala dentro del cuerpo del método.

Dado que las estadísticas se desean generar cuando se produce la caída del primer nodo, resulta conveniente detener la simulación cuando ocurre este evento. Para implementar este comportamiento se introduce un parámetro en el módulo de administración de recursos que activa o desactiva esta opción, y modificaciones menores en el código C++.

Se ha descrito la implementación de FROMS en el simulador como lo ha realizado Datta & Soundaralaskshmi, se comprobó que NS-3 provee las herramientas necesarias para realizar simulaciones de redes de sensores inalámbricos. Su estructura modular facilita la integración de nuevos módulos con los ya existentes, y permite personalizar el simulador sin comprometer su integridad.

Asimismo, se comprobó que REL es un buen candidato para la comparación con FROMS, ya que ambos usan un esquema de enrutamiento consciente de la energía. Además, está disponible el código fuente para NS-3 compartido por los propios desarrolladores del protocolo. En base a esto se puede replicar u obtener resultados similares a los datos (gráficas y estadística) obtenidos por los investigadores Datta & Soundaralaskshmi.

3.6 Simulaciones de protocolos de enrutamiento

A continuación, se evalúa el desempeño del algoritmo de enrutamiento ‘inteligente’ FROMS mediante el simulador NS-3. Se debe comprobar, si un algoritmo basado en Inteligencia Computacional, brinda un comportamiento superior con respecto a uno convencional. Se realiza la comparación con el algoritmo convencional REL bajo la premisa de que el aprendizaje, en los algoritmos inteligentes, permite tomar decisiones que prolonguen el tiempo de vida de la red.

3.6.1 Configuración de los parámetros del simulador

Antes de la ejecución de una simulación es necesario realizar dos pasos. El primero de ellos es el modelado del dispositivo que se va a usar. Se indican cuáles son los módulos que describen el comportamiento del dispositivo, por ejemplo: tipo de aplicación, módulos de comunicación y modelo del transceptor. El segundo paso es especificar los parámetros de simulación. Consiste en describir la topología de la red y especificar el tiempo de simulación y parámetros de cada uno de los módulos utilizados.

3.6.2 Topología de la red

Todas las simulaciones se realizan sobre una red de topología en malla con un aspecto de grilla regular. Este tipo de topologías es muy común en simulaciones para la evaluación de protocolos para redes de sensores inalámbricos (Sarkar & Murugan, 2016). El tamaño de la red se estableció en 35 nodos sensores y una estación base, como se muestra en la figura 3.14.

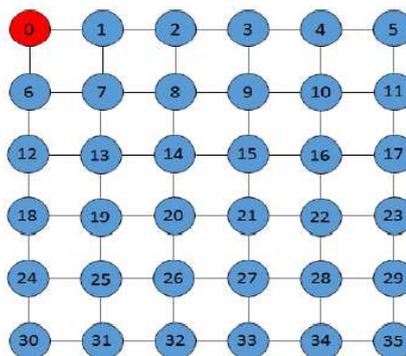


Figura 3. 14. Topología de red de prueba.

Fuente: el autor

El área de despliegue podría ser un terreno de 100 m x 100 m, por lo que los nodos tendrán separación de 20 m de distancia. La estación base se ha ubicado en la esquina superior izquierda, ya que en esta posición se maximiza la cantidad de saltos desde los nodos más alejados. Con esta configuración se puede comparar el rendimiento de los umbrales de REL contra el mecanismo de aprendizaje y funciones de costo de FROMS.

Al respecto de las métricas más importante en WSNs, utilizada para evaluar y comparar la performance de los diferentes protocolos propuestos en la literatura, es el tiempo de vida de la red (network lifetime). Esta métrica puede ser definida de varias maneras. La más comúnmente usada es la que define a la misma como: i) la duración en tiempo hasta que el primer nodo de la red se queda sin energía. Sin embargo, pueden definirse variantes de esta, como, por ejemplo, ii) la duración en tiempo hasta que un porcentaje de nodos se quede sin energía, entre otras. (Gáston, 2015).

Desde otro punto de vista, se puede definir también como iii) la duración en tiempo durante el cual un porcentaje específico de nodos están funcionales y conectados, cumpliendo la red con requerimientos de calidad de servicio. Un buen protocolo de comunicación para WSNs es aquel que logra su objetivo maximizando el tiempo de vida de la red, es decir, que permita mantener funcional a la red la mayor cantidad de tiempo posible, sin que se produzca agotamiento energético prematuro.

3.6.2.1 Modelo del dispositivo

El simulador NS-3 posee una estructura modular, donde cada nodo sensor se construye mediante la interconexión de módulos para las diferentes capas. Cuando se define un nodo, se debe especificar sus coordenadas y características (protocolos utilizados, transceptor, módulo de aplicación, etc.). La tabla 3.2 recoge los módulos utilizados en la definición de cada uno de los nodos.

Tabla 3. 2. Módulos utilizados en la definición de los nodos

Nombre del módulo	Descripción
ThroughputTest	Aplicación de reportes periódicos.
FromsRouting/RelRouting	Protocolo de enrutamiento FROMS/REL
TMAC	Protocolo de subcapa MAC específico para redes de sensores.
CC2420	Modelo de transceptor de Texas Instruments.

Fuente: el autor

El módulo ThroughputTest se corresponde con una aplicación que genera reportes cada un determinado intervalo de tiempo. Incorpora estadísticas sobre la tasa de recepción, tasa de pérdidas, latencia y cantidad de paquetes recibidos en el nodo

designado como sumidero. Dado que tanto REL como FROMS son diseñados para su uso en escenarios de tráfico periódico, este es el módulo de aplicación adecuado para las simulaciones.

El protocolo de subcapa MAC utilizado es T-MAC (Timeout-MAC). Este protocolo permite a los nodos sensores activar el transceptor en instantes de tiempo sincronizados, y apagarlos cuando no se detecta actividad en el canal durante un determinado periodo de tiempo. Usa un ciclo útil adaptativo, lo que se traduce en que los nodos consumen más o menos energía en dependencia de la cantidad de tiempo que esté activo su transceptor y la cantidad de paquetes que procesan (Wang, Tao , & Qian , 2014).

La interfaz de red es la CC2420 de Texas Instruments es compatible con el estándar 802.15.4 y utiliza la banda de 2.4 GHz para comunicaciones de baja potencia y tasa de datos de hasta 250 kbps. La potencia de transmisión es programable y varía desde 0 hasta -25 dBm, como se muestra en la tabla 3.3. La hoja de datos no especifica el alcance de radio para cada uno de los niveles de potencia, por lo que el valor adecuado se debe determinar mediante una simulación. Una potencia de transmisión de -5 dBm es la adecuada para la topología de red de prueba. Los niveles mayores provocan la conectividad con nodos más lejanos, mientras que con niveles menores se pierde toda conectividad entre los nodos.

Tabla 3. 3. Niveles de potencia de salida del transceptor CC2420

Registro TXCTRL	Potencia de salida [dBm]	Consumo [mA]
0xA0FF	0	17.4
0xA0FB	-1	16.5
0xA0F7	-3	15.2
0xA0F3	-5	13.9
0xA0EF	-7	12.5
0xA0EB	-10	11.2
0xA0E7	-15	9.9
0xA0E3	-25	8.5

Fuente: el autor

3.6.2.2 Parámetros generales de simulación

Para la evaluación de los protocolos se realizan varias simulaciones sobre la topología de red propuesta. Los parámetros generales de simulación se recogen en la tabla 3.4. Estos son los parámetros que se mantienen invariables en todas las simulaciones, y se establecen en el archivo de configuración. Los parámetros para cada uno de los protocolos de enrutamiento varían en cada escenario, por lo que no se listan en la tabla. También se listan los módulos de cada capa (ver tabla 3.1) con los parámetros anteriormente explicados en el sub capítulo Modelo del dispositivo.

Tabla 3. 4. *Parámetros generales de simulación*

Parámetro	Valor
Dimensiones del terreno	100m x 100m
Número de nodos	36
Tiempo de simulación	80 min
Energía inicial de los nodos	100 J
Intervalo entre reportes	10 s
Envío del primer reporte	10 s
Envío del primer anuncio	5 s

Fuente: el autor

El tiempo de simulación es de 80 minutos, un valor lo suficientemente grande para que se produzca la caída del primer nodo y se construya el histograma de energía antes de que se venza el tiempo de simulación. Este valor se determinó mediante el método de estimación del tiempo de vida de la red de NS-3, modificado para las simulaciones en este trabajo.

La energía inicial de los nodos es de 100 Joules. El motivo de elegir este valor es disminuir considerablemente el tiempo de simulación necesario para visualizar los resultados. Dado que tanto REL como FROMS toman decisiones de encaminamiento en base al porcentaje de energía residual de los nodos, no es necesario especificar un valor mucho mayor para la energía de las baterías, lo cual aumentaría considerablemente el tiempo de simulación necesario para que algún

nodo agote sus reservas energéticas. De hecho, 100 Joules es el valor especificado en el código de REL proveído por sus desarrolladores para las simulaciones.

El primer anuncio se envía a los 5 segundos de iniciada la simulación, para dar margen a que el protocolo T-MAC sincronice los nodos antes del intercambio de datos. El primer reporte de aplicación se produce a los 10 segundos, cuando ya los nodos han inicializado sus tablas de enrutamiento con la información contenida en los mensajes de anuncios.

3.7 Escenarios de simulación

La tabla 3.5 muestra los cuatro escenarios de simulación considerados. En el primer escenario se varía la función de costo para determinar el impacto de cada una sobre el tiempo de vida de la red. De acuerdo a los resultados obtenidos, se varía la velocidad de aprendizaje para determinar cuál ofrece el mejor rendimiento en términos de eficiencia energética y tasa de recepción.

Tabla 3. 5. *Escenarios de simulación.*

Escenario	Protocolo	Descripción
1	FROMS	Variación de la función de costo
2	FROMS	Variación de la velocidad de aprendizaje
3	FROMS	Variación de la tasa de exploración
4	FROMS/REL	Comparación de los protocolos

Fuente: el autor

La mejor velocidad de aprendizaje luego se incluye como parámetro en el próximo escenario, donde se varía la tasa de exploración del algoritmo Q-Learning utilizado por FROMS. Estos tres pasos permiten determinar los mejores parámetros de aprendizaje para el algoritmo inteligente. Finalmente, en el cuarto escenario se realiza una comparación del rendimiento de REL y FROMS con sus parámetros óptimos, para identificar las ventajas del uso de técnicas inteligentes en el enrutamiento en redes de sensores.

Tabla 3. 6. *Métricas de evaluación.*

Métrica	Resultado deseado
Tiempo de caída del primer nodo	Alto
Distribución de la energía	Grupo compacto
Desviación estándar de la energía residual de las baterías	Baja
Tasa de recepción	Alta

Fuente: el autor

Para la comparación de los protocolos se utilizan las métricas resumidas en la tabla 3.6. Para evaluar la capacidad de extender la vida de la red se consideran tres métricas: distribución de la energía residual en el histograma de energía, tiempo de caída del primer nodo, y la desviación estándar de la energía residual de las baterías.

Estas métricas plantean que el comportamiento deseado es que, en cualquier punto dentro del tiempo de vida de la red, el histograma de distribución de la energía debe exhibir un grupo compacto, con todos los nodos teniendo más o menos la misma energía residual. (WANG, 2015). En las simulaciones se construye el histograma en un solo punto en el tiempo, aquel donde se produce la caída del primer nodo.

La desviación estándar de la energía residual de los nodos es una medida de la dispersión de la energía residual de los nodos. Una desviación estándar baja significa que se logra un buen balance de los gastos energéticos entre los nodos de la red. En las comparaciones también se incluye la tasa de recepción como una medida de la confiabilidad de los protocolos.

3.7.1 Comparación de las funciones de costo

Para observar de manera aislada los resultados obtenidos por cada una de las funciones de costo, se utiliza una política de enrutamiento ávida. En otras palabras, no se realizan acciones de exploración ($\epsilon = 0$), por lo que en todo momento cada nodo utiliza la ruta de mejor estimado en su tabla de enrutamiento.

Se fija además una tasa de aprendizaje $\gamma = 1$, así cada nodo sustituye inmediatamente los estimados actuales por los recibidos. Todo esto permite ver de

manera más clara al final de la simulación la influencia de las funciones de costo sobre los resultados.

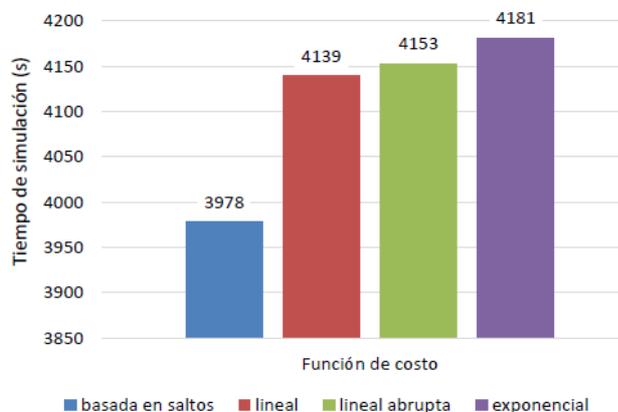


Figura 3. 15. Tiempo de vida de la red para cada una de las funciones de costo

Fuente: el autor

La figura 3.15 muestra el tiempo de caída del primer nodo para cada función de costo. Las funciones de costo que usan una métrica combinada de saltos y energía residual logran aumentar el tiempo de vida de la red entre 161 y 210 segundos con respecto a la función de costo basada en saltos. En este sentido, el mejor resultado se obtiene con el uso de la función de costo exponencial, que logra un incremento de un 5.1% en el tiempo de vida de la red.

La función basada en saltos selecciona rutas que minimicen la cantidad de saltos requerida para alcanzar el sumidero. Esto se traduce en una disminución de la energía necesaria para alcanzar el sumidero. Sin embargo, los nodos en las rutas más cortas agotan su batería en menor tiempo.

Por el otro lado, las funciones dinámicas de costo, que usan una métrica combinada de saltos y energía, favorecen las rutas más cortas con mayor energía residual en los nodos involucrados, dispersando así el gasto de energía entre los nodos de la red. Este comportamiento se puede observar en la desviación estándar de la energía residual de los nodos. Mientras menor es el valor de la desviación estándar, mejor es el balance de energía obtenido por la función de costo.

La tabla 3.7 recoge el decremento de la desviación estándar de la energía residual para cada función de costo dinámica con respecto a la función de costo basada en saltos. El menor valor se logra con la función de costo exponencial, y esta es la

razón por la que incrementa el tiempo de vida de la red en mayor medida, en comparación con las otras funciones.

Tabla 3. 7. Desviación estándar de la energía residual para cada función de costo.

Función de costo	Decremento de la desviación estándar
Lineal	2.04 %
Lineal abrupta	5.09 %
Exponencial	5.74 %

Fuente: el autor

El estudio realizado por Machado, et al (2013) sobre el uso de funciones de costo en FROMS en un ambiente menos realístico como MATLAB, muestra resultados diferentes. El tiempo de caída del primer nodo usando la función de costo exponencial se extiende en un 80%. En dichas simulaciones, los nodos degradan la energía de su batería en una cantidad fija cada vez que transmiten un paquete, y no se considera el consumo de energía por la escucha pasiva del canal.

En un ambiente más realístico y corriendo el algoritmo de enrutamiento sobre un protocolo MAC real, los resultados pueden diferir, pues en la simulación no se toma en cuenta ambiente extremos, obstáculos artificiales o la curvatura de la tierra cuando se trata de cubrir decenas de kilómetros con una red de sensores inalámbricos. Y por sobre todo en una red real, los nodos consumen energía de acuerdo al estado y tiempo de actividad del módulo de radio, y no precisamente sobre la cantidad de paquetes transmitidos. Por lo tanto, debido a que cada nodo escucha las transmisiones de todos sus vecinos, se consume energía aun cuando no se interviene en el encaminamiento del paquete hacia su destino.

Incluso cuando se conmuta hacia rutas alternativas con mayores niveles de energía, la función de costo dinámica selecciona rutas con estimados similares a la anterior.

3.7.2 Selección de la velocidad de aprendizaje

La velocidad de aprendizaje es una constante que determina el tratamiento que se le da a los estimados recibidos. Velocidades de aprendizaje altas dan mayor prioridad a los estimados recibidos sobre los almacenados. Por el otro lado, bajas

velocidades de aprendizaje priorizan los estimados acumulados sobre los recibidos. La selección del valor adecuado depende de las fluctuaciones de los valores Q y de los requerimientos de la aplicación.

Para encontrar dicho valor, se prosigue con el uso de una política ávida de selección de rutas, nuevamente con el objetivo de tener información más clara sobre el parámetro que se desea encontrar. En este caso se utiliza la función de costo exponencial, que es la que más prolonga el tiempo de vida de la red.

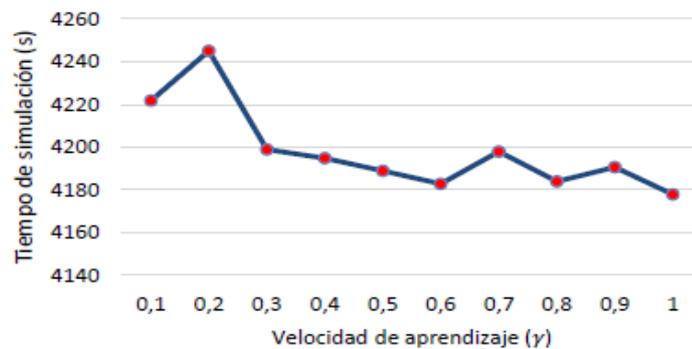


Figura 3. 16. Tiempo de vida de la red para diferentes velocidades de aprendizaje

Fuente: el autor

La figura 3.16 muestra el tiempo de vida de la red en función del tiempo de caída del primer nodo para cada velocidad de aprendizaje γ . Como se puede apreciar, el mejor resultado se obtiene para una velocidad de aprendizaje $\gamma=0.2$. Esto se debe a que los estimados recibidos no varían mucho en el tiempo, ya que la energía de los nodos no decae bruscamente. Por tanto, con bajas velocidades de aprendizaje se logra una mejor adaptación ante las fluctuaciones de los valores Q .

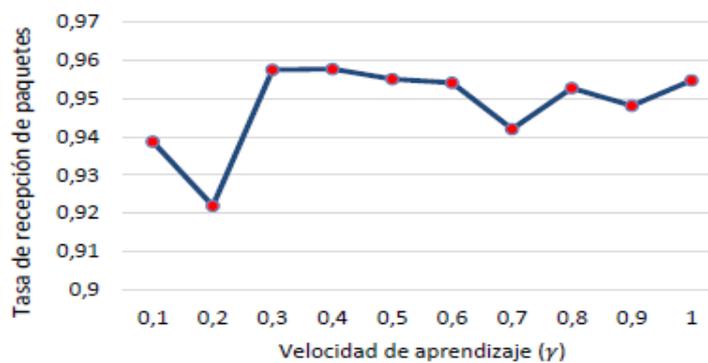


Figura 3. 17. Tasa de recepción de paquetes para diferentes velocidades de aprendizaje

Fuente: el autor

La figura 3.17 recoge la tasa de recepción de paquetes en el sumidero para cada velocidad de aprendizaje. En todos los casos se obtienen valores por encima del 92%, que es un buen valor si se considera que la mayoría de las aplicaciones de redes de sensores requieren tasas de recepción de al menos 80 %

En el anexo I se puede observar la influencia de la velocidad de aprendizaje sobre la latencia. Los requerimientos de latencia y tasa de recepción de paquetes varían de acuerdo a la aplicación, pero el tiempo de vida de la red es un parámetro crítico en toda WSN. Por lo tanto, para los escenarios siguientes se utiliza una tasa de aprendizaje de 0.2, ya que es la que más extiende el tiempo de caída del primer nodo.

3.7.3 Selección de la tasa de exploración

La tasa de exploración (ϵ) define la probabilidad de realizar acciones exploratorias en búsqueda de rutas óptimas. Tasas de exploración bajas provocan que el algoritmo explote con más frecuencia su mejor ruta en la tabla de enrutamiento. Lo opuesto sucede para altas tasas de exploración. Para determinar el valor adecuado, se fija la velocidad de aprendizaje en 0.2, y se varía la tasa de exploración desde 0 hasta 0.3, en pasos de 0.1. En todas las repeticiones de la simulación se usa la función de costo exponencial.

La tasa de exploración tiene un fuerte impacto en la tasa de recepción de paquetes, como se muestra en la figura 3,18. Para $\epsilon > 0$, la tasa de recepción cae bruscamente, coincidiendo en este aspecto con los resultados obtenidos por los creadores de FROMS en sus simulaciones en MATLAB. Esto se debe a un doble comportamiento exploratorio: la exploración implícita de la función de costo y la técnica de exploración ϵ -greedy. La técnica de ϵ -greedy es una variación de la exploración greedy. El agente identifica la mejor acción de acuerdo a la función ϵ -greedy (s,a). La constante ϵ es un valor entre 0 y 1 que determina la probabilidad con la que se elegirá una acción que no corresponda a la que provee el mayor refuerzo. (Polimeni & Richi, 2004)

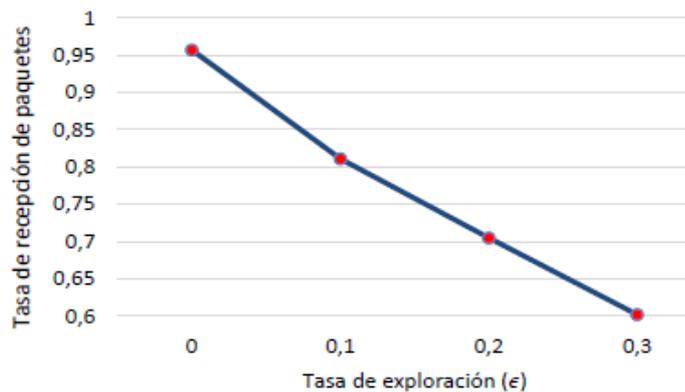


Figura 3. 18. Tasa de recepción de paquetes para diferentes tasas de exploración.

Fuente: el autor

Este comportamiento se puede apreciar con más detalle en la figura 3.19, obtenida mediante el script de gráficos de NS-3. A medida que aumenta la tasa de exploración, y los nodos utilizan rutas sub-óptimas con más frecuencia, se produce un incremento en las pérdidas de paquetes procedentes de los nodos más alejados.

Esta tendencia es consecuencia de que las acciones de exploración no están limitadas a un conjunto de vecinos, y todos tienen la misma probabilidad de ser elegidos como próximo salto en el camino del paquete. Por lo tanto, es posible que se encaminen paquetes hacia nodos más alejados del sumidero. El número de nodos intermedios tomando acciones sub-óptimas aleatorias aumenta con la distancia, lo que provoca que los paquetes deambulen por la red, tomando rutas más largas y potencialmente malas, e incrementando así las pérdidas.

Antes de introducir el valor Q del vecino en la regla de actualización, la función de costo degrada el estimado recibido de acuerdo al nivel de energía residual de la batería. Por lo tanto, en todo momento se tiene un estimado casi exacto del costo de encaminar un paquete hacia cualquiera de los nodos en la vecindad, por lo cual un mecanismo de exploración explícito se hace redundante.

Visto de otra manera, cuando se usan funciones dinámicas como la función de costo exponencial, cuyos valores varían en el tiempo, incluso con una tasa de exploración nula se pueden aprender las mejores rutas. Esto es posible porque los estimados recibidos también varían, y con ellos los valores Q de cada ruta almacenada. En otras palabras, incluso cuando la tasa de exploración se fija en 0, FROMS está

implícitamente explorando la red, por lo que otro mecanismo de exploración externo se hace indeseable.

De acuerdo a los resultados obtenidos en este escenario, los mejores parámetros para el aprendizaje de FROMS en la topología de red de prueba son $\gamma=0.2$ y $\epsilon=0$, y la función de costo que mejores resultados obtiene en cuanto al tiempo de vida de la red es la función de costo exponencial.

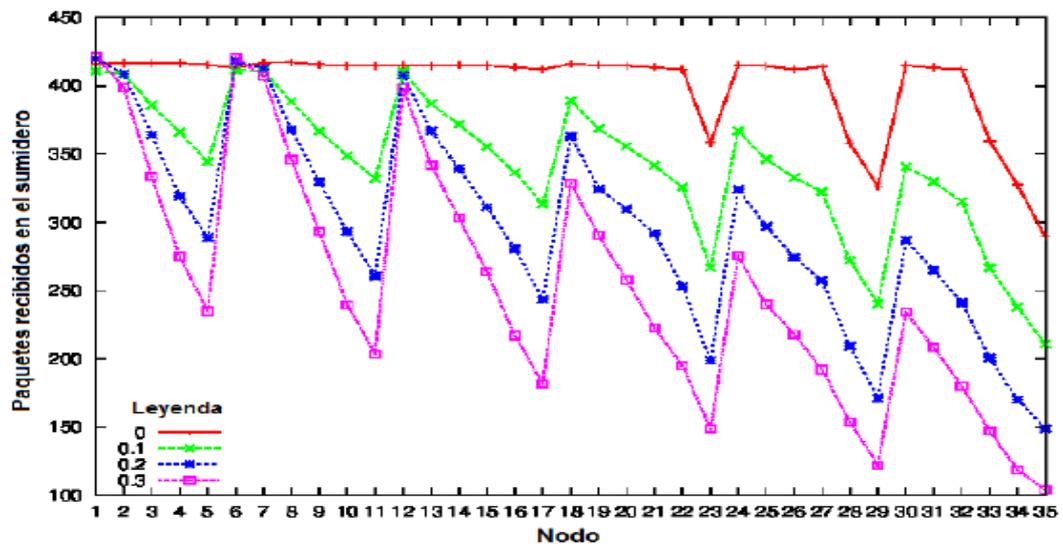


Figura 3. 19. Paquetes recibidos en el sumidero por cada nodo en la red

Fuente: el autor

Comparación de FROMS y REL

Para identificar las potencialidades de FROMS se realiza una comparación con el protocolo REL. Para ello se corren dos simulaciones por separado, cada una utilizando uno de los protocolos. Las métricas consideradas para la comparación son las de la tabla 3.8.

La tabla 3.8 muestra los resultados obtenidos por cada protocolo en cuanto al tiempo de vida de la red, tasa de recepción de paquetes y desviación estándar de la energía residual. La tasa de recepción de paquetes en el sumidero alcanzada por FROMS es ligeramente menor que la de REL, esto se debe a que las transmisiones se realizan por difusión, por lo que aumenta el número de colisiones. Para disminuir las pérdidas de paquetes, FROMS incrementa un mecanismo de confirmación de recepción pasivo, ya que las confirmaciones de recepción para tramas de difusión

no están definidas en T-MAC, pero esto no es suficiente para alcanzar los resultados alcanzados por REL.

REL, por su parte, utiliza envíos unicast para sus paquetes de datos. Esto le permite beneficiarse del handshake Request-To-Send (RTS)/Clear-To-Send (CTS), reduciendo así posibles colisiones. También se apoya de mensajes de confirmación de recepción a nivel de enlace de datos.

Tabla 3. 8. Resultados obtenidos por cada protocolo.

Protocolo	Tasa de recepción	Tiempo de caída del primer nodo	Desviación estándar de la energía residual
FROMS	95.73 %	4206 segundos	5.47 J
REL	97.54 %	3585 segundos	2.23 J

Fuente: el autor

FROMS logra incrementar el tiempo de vida de la red en un 17 % con respecto a REL, en parte debido a un menor gasto de energía por la transmisión de mensajes de control para el mantenimiento de rutas. Cuando se usa el protocolo REL, cada vez que algún nodo excede el umbral de energía residual E_{th} , informa a sus vecinos mediante un mensaje de notificación RADV (*Route Advisor*) sobre su estado de energía, de manera que éstos evalúen su inclusión en la ruta hacia el sumidero. Este esquema permite el cambio hacia rutas más largas, evitando los nodos con bajos niveles de energía residual, pero el envío de los mensajes RADV también acarrea un costo energético.

Por el otro lado, FROMS se apoya de la realimentación adjunta a los datos para informarse sobre los estimados y estado energético de sus vecinos. Debido a que las transmisiones se realizan por difusión a nivel de enlace de datos, cada nodo puede escuchar las transmisiones de sus vecinos, por lo que no se requieren gastos adicionales para la transmisión de información de control.

Otra causa de la eficiencia de FROMS es el algoritmo inteligente subyacente. La velocidad de aprendizaje es capaz de suavizar los estimados recibidos y usar por un tiempo mayor la ruta actual antes de conmutar hacia otra. En otras palabras, no reemplaza directamente el estimado actual de la ruta por la realimentación del

vecino, como sucede en los protocolos de enrutamiento convencionales. Este comportamiento permite acumular experiencia, y a largo plazo, aumentar el rendimiento.

La figura 3.20 muestra el histograma de energía en el instante de tiempo en que ha caído el primer nodo.

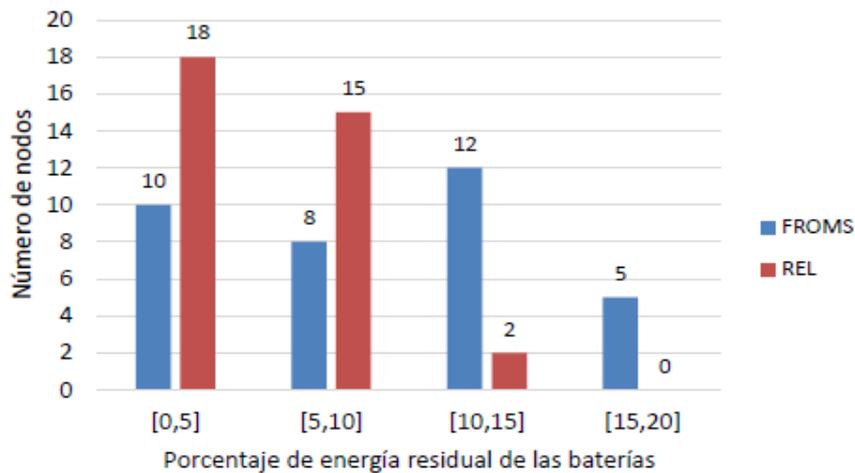


Figura 3. 20 Histograma de la energía residual en baterías

Fuente: el autor

El número de nodos con niveles de energía menores que el 5% casi se duplica cuando se usa el protocolo REL, lo mismo sucede en el intervalo del 5-10 %. Esto es en parte debido a la necesidad de transmitir los RADV (*Route Advisor*) de manera separada, que conlleva a un aumento de los gastos energéticos.

La otra razón es que REL depende de que se cumplan varias condiciones antes de conmutar hacia otra ruta alternativa. Incluso cuando la energía residual del próximo salto en la ruta es menor que la de algún otro vecino en una ruta alternativa, el algoritmo no conmuta si se excede el umbral de saltos $HCdiffmax_allow$, o si la ruta alternativa posee mayor cantidad de enlaces débiles. Aunque esto representa una desventaja con respecto a FROMS, forma parte de la confiabilidad característica de REL

3.8 Discusión de Resultados

El uso de algoritmos basados en IC otorga capacidad de aprender a los nodos de una red Zigbee, es posible que cada nodo ‘cambie’ su comportamiento en base a

nueva información censada, es, decir a medida que se ‘entrena’ un sistema mejora y termina siendo capaz de hacer ‘operaciones’ para las que no se le había programado explícitamente.

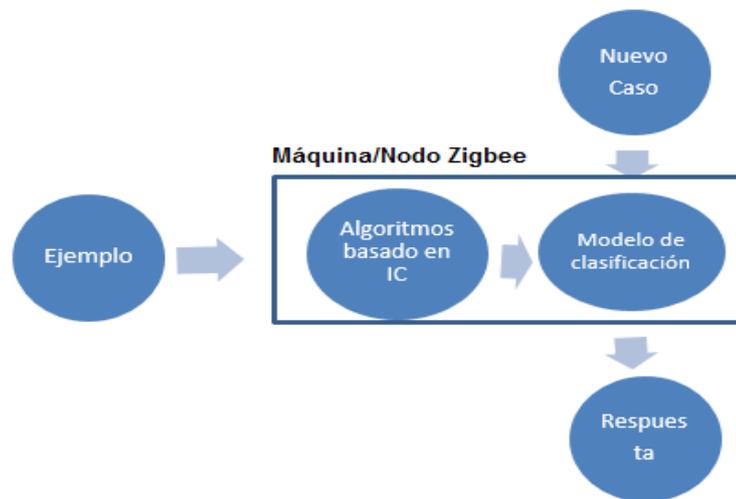


Figura 3. 21 Esquema de aprendizaje de nodo en red Zigbee

Fuente: el autor

Básicamente, la máquina aprende ‘sola’ porque su algoritmo de aprendizaje automático se basa en ejemplos de actuación, ejemplos que clasifica y optimiza para futuras respuestas. Es decir, que la máquina adquiere su propia experiencia y aprende de ella para mejorar. El aprendizaje puede ser supervisado, en mayor o menor medida, por un humano, o puede no ser supervisado.

El protocolo REL logra una mejor distribución de los gastos energéticos entre los nodos de la red, casi logrando un grupo compacto entre un 5 % y 10 % de la energía de los nodos. De hecho, la desviación estándar de la energía residual de las baterías obtenida por REL es menor que la alcanzada por el protocolo FROMS. Sin embargo, se agota la energía de las baterías más rápidamente.

Con esto se muestra la importancia de disminuir al máximo posible los costos de comunicación en una red de sensores inalámbricos, y la capacidad de FROMS de aprender sobre el estado de la red sin gastos adicionales.

CONCLUSIONES

En cumplimiento del primer objetivo, las plataformas basadas en Zigbee, consiste normalmente en pequeños dispositivos espacialmente distribuidos para monitorear de forma cooperativa condiciones físicas o ambientales tales como temperatura, sonido, vibración, etc. Integran las llamadas redes de sensores inalámbricos WSN, los datos de sensores remotos de diferentes tipos son recolectados por la unidad central para procesamiento y posterior análisis.

En cumplimiento del segundo objetivo específico; caracterizar el protocolo de enrutamiento adecuado para operar en forma autónoma y maximizando su tiempo de vida, a través de la literatura científica, se ensayan y se tienen resultados favorables con el uso de la inteligencia computacional que introduce mecanismos adaptativos a los algoritmos de enrutamiento. Estos mecanismos están fundamentados en reglas de aprendizaje capaces de tomar decisiones de enrutamiento óptimas en base a una o varias métricas, aprendizaje que permite tomar las mejores decisiones a largo plazo.

El cumplimiento del tercer objetivo específico, se determinó que el protocolo de enrutamiento de clase reactiva AODV donde los nodos mantienen una tabla de enrutamiento para los destinos conocidos, ha sido el protocolo usado por defecto en redes Zigbee de topología mesh pero la variante de enrutamiento realimentado a múltiples sumidero (Feedback ROuting to Multiple Sinks, FROMS) puede minimizar el consumo energético en la red Zigbee. Este tipo de protocolo explota las potencialidades del aprendizaje por refuerzos necesita de programación o códigos en C++, es adecuado para cualquier aplicación de envío periódico de datos hacia uno o varios destinos o sumideros en un ambiente multi-salto.

Las estimaciones estadísticas de dos tipos de protocolos uno convencional REL con 35 nodos, muestran que FROMS logra extender el tiempo de vida de la red en un 17% con respecto a REL y otro basado en inteligencia computacional indican que este último es un algoritmo de enrutamiento eficiente.

Para evaluar la capacidad de extender la vida de la red se consideran tres métricas: distribución de la energía residual en el histograma de energía, tiempo de caída del primer nodo, y la desviación estándar de la energía residual de las baterías

En cumplimiento del cuarto objetivo específico; concluye que la topología de comunicación con protocolos de ruteo basados en inteligencia computacional, FROMS puede permitir un desempeño o comportamiento inteligente a una red Zigbee o WSN, ya que reduce los gastos en el descubrimiento y mantenimiento de rutas, y mediante el uso de funciones dinámicas de costo y una regla de actualización puede seleccionar en todo momento las rutas más cortas hacia el sumidero o nodo destino, a la vez que incrementa el tiempo de vida de la red.

RECOMENDACIONES

Se recomienda:

Analizar y evaluar la eficiencia del modelado de optimización energético para redes con más de 40 nodos con el uso de métodos de optimización numérica más complejos.

Comparar protocolos de enrutamiento reactivo basado en algoritmos inteligentes utilizando otro simulador diferente de NS-3.

Realizar comparaciones con redes de sensores implementadas y con escenarios en área rural y urbana.

Utilizar la topología bajo FROMS para diversas áreas en donde dispositivos sensoriales (nodos) de monitoreo para la prevención y detección de desastres ambientales.

Desarrollar algoritmos basados en inteligencia artificial, basados en métodos de optimización y búsqueda de soluciones para redes de sensores inalámbricos.

BIBLIOGRAFÍA

- Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2004). *A review of routing protocols for mobile ad hoc*. Obtenido de Research Online. Repositorio University of Wollongong: <http://ro.uow.edu.au/cgi/viewcontent.cgi?article=2167&context=eispapers>
- Agrawal, R., Tripathi, R., & Tiwari, S. (2012). *Performance comparison of AODV and DYMO MANET protocols under wormhole attack environment International Journal of Computer Applications*, vol. 44, n.º 9, pp. 9-16. Obtenido de <http://research.ijcaonline.org/volume44/number9/pxc3878482.pdf>
- Agrawal, S., Jain, S., & Sharma, S. (2011). *Mobility based performance analysis of AODV and DYMO under varying degree of node misbehavior*”, *International Journal of Computer Applications*, vol. 30, n.º 7, pp. 36-41. Obtenido de <https://pdfs.semanticscholar.org/fa8b/3ef06d7e88f6c9b3f82c2dd56281a84c2ed0.pdf>
- Barcell, F. (2010). *Wireless Sensor Network*. Obtenido de Sitio web: Manuel Fernández Barcell: <http://www.mfbarcell.es/conferencias/wsn.pdf>
- Buratti, C., Martaló, M., Verdone, R., & Ferrari, G. (2011). *Sensor Networks with IEEE 802.15.4 Systems: Distributed Processing, MAC, and Connectivity*. p-p. 12. Berlín. Alemania: Springer.
- Castro, J. (2014). *Uso del protocolo CoAP para la implementación de una aplicación domótica con redes de sensores inalámbricas*. Obtenido de Universidad Politécnica de Cartagena. Escuela Técnica Superior de Ingeniería de Telecomunicación: <http://repositorio.upct.es/bitstream/handle/10317/4163/pfc5908.pdf?sequence=1>
- Chang, J., & Tassiulas, L. (2005). *Maximum Lifetime Routing for Wireless Sensor Networks*,» *IEEE/ACM Transactions On Networking*, Vol. 12, No.4.

- Datta, A., & Soundaralaskshmi, S. (2010). *A Survey of State-of-the-Art Routing Protocols for Mobile Ad Hoc Networks*. In Laurence T. Yang & Agustinus Borgy Waluyo (Ed.), *Mobile Intelligence*. John Wiley & Sons.
- Dietrich, I., Sommer, C., & Dressler, F. (2007). “*Simulating DYMO in OMNeT++*”, *Departament of Computer Science, University of Erlangen, Technical*. Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.8764&rep=rep1&type=pdf>
- Dominguez, C. (2011). *Algoritmos bioinspirados para el encaminamiento de datos en redes inalámbricas de sensores*. Obtenido de http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/9230/Dominguez_Tesis.pdf?sequence=1
- Fernández, R., Ordieres, J., González, A., & Alba, F. (2009). *Redes Inalámbricas de Sensores: Teoría y aplicación práctica*. ISBN: 978-84-692-30007-7 Logroño. España: Publicaciones de Universidad de la Rioja. Obtenido de Repositorio Digital; Universidad de la Rioja.
- Flores, E. (2012). *Redes de sensores Inalámbricos Aplicados a la Medicina*. Obtenido de Repositorio Digital de Universidad de Cantabria: <https://repositorio.unican.es/xmlui/bitstream/handle/10902/1288/349251.pdf?sequence=1>
- Garbarino, J. (2011). *Protocolos para redes inalámbricas de sensores*. Obtenido de Repositorio digital de la Universidad de Buenos Aires: <http://materias.fi.uba.ar/7500/Garbarino.pdf>
- Gáston, G. (2015). Obtenido de Tesis doctoral de la Universidad Nacional de Córdoba
- Giandoménico, E., D’Agostino, E., Corti, R., Martínez, R., & Belmonte, J. (2010). *Métricas de evaluación de un algoritmo de encaminamiento para redes de sensores*. Obtenido de Secretaría de Ciencia y Técnica. Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario: http://www.fceia.unr.edu.ar/secyt/rt/2010/rtid10_01.pdf

- Gómez, P., Posada, G., & Vallejo, m. (2014). *Evaluación del desempeño del protocolo de enrutamiento AODV para diferentes escenarios de redes de sensores inalámbricos*. Obtenido de Revista: Ingeniería y Desarrollo. V. 32 N. (1): <http://www.redalyc.org/comocitar.ou?id=85230428006>
- González, D. (2011). *Estudio comparativo del estandar Ultrawide Band (UWB) frente a Bluetooth Zigbee*. Obtenido de <http://cybertesis.uach.cl/tesis/uach/2011/bmfcig643e/doc/bmfcig643e.pdf>
- Heinzelman, W., Kulic, J., & Balakrishnan, H. (2009). *Adaptive Protocols for Information Dissemination in Wireless Sensor Networks*,» de *Proceedings of the ACM MobiCom'09, Seattle*.
- Hernández, R. (2011). *Algoritmo de colonia de hormigas para el enrutamiento en redes de sensores inalámbricos*. Repositorio Universidad Centroccidental, Barquisimeto.
- Kanjo, E., & Trossen, D. (2015). *Sensor Systems and Software: 5th International Conference, S-CUBE 2014*. DOI: 10.1007/978-3-319-17136-4 Londres. Reino Unido: Springer.
- Kaur, A., & Singh, H. (2013). *A Study of Secure Routing protocols*. *International Journal of Application or Innovation in Engineering & Management, Vol. 2(3)*. Obtenido de International Journal of Application or Innovation in Engineering & Management (IJAIEM): <http://www.ijaiem.org/Volume2Issue2/IJAIEM-2013-02-27-042.pdf>
- Kulkarni, A., Forster, A, & Kumar, G. (23 de Septiembre de 2011). "*Computacional Intelligence in Wireless Sensor Networks*". Obtenido de IEEE Communications Surveys and Tutorials, vol. 13.
- Kulkarni, R., Forster, A., & Kurnar, G. (2011). *Computacional Intelligence in Wireless Sensor Networks*», *IEEE Communications Surveys and Tutorials, vol. 13*. Obtenido de <http://dl.acm.org/citation.cfm?id=2211525>
- Machado, K., Rosário, D., Cerqueira, E., Loureiro, A., Neto, A., & Souza, J. d. (2013). *A Routing Protocol Based on Energy and Link Quality for Internet of Things Applications*. *Multidisciplinary Digital Publishing Institute*

(MDPI). *Base de Datos Digital NCBI*. DOI: 10.3390/sl30201942, 1942-1964.

Maldonado, J. (2013). *Sistema Inteligente de monitoreo para el consumo de energía de redes inalámbricas Ad hoc*. Obtenido de <http://www.bdigital.unal.edu.co/39515/1/300221.2014.pdf>

Mazón, D. (2011). *Implementación de algoritmos de enrutamiento para redes inalámbricas de sensores móviles*. Obtenido de Repositorio digital ETSINF: <https://riunet.upv.es/bitstream/handle/10251/11710/Memoria.pdf?sequence=1>

Morales, J., & Gómez, A. (2007). *La Red Inteligente: Ahorro energético y Telecomunicaciones*. Conferencias sobre eficiencia energética en infraestructura de redes inteligentes: L&M Data Communications. Obtenido de Ethernet Forum.

Morocho, & Hipo. (2012). *Implementación de una red Zigbee para aplicaciones industriales, caso práctico Laboratorio de automatización industrial EIS*. Obtenido de Repositorio Digital; Universidad ESPOCH: <http://dspace.espoch.edu.ec/bitstream/123456789/1498/1/18T00460.pdf>

Muñoz, J. (2010). *Inteligencia Computacional inspirada en la vida*. Repositorio institucional de la Universidad de Málaga (RIUMA) <http://riuma.uma.es>. Málaga. España: SPICUM. Servicio de Publicaciones de la Universidad de Málaga.

Murias, G. (2009). *Comunicación digital Inalámbrica de baja velocidad en la banda ISM de 2.4 GHz*. Obtenido de <http://www3.fi.mdp.edu.ar/electronica/catedras/comunicacionesdigitales/files/Informes%20Trabajos/2009/Gustavo%20Murias%20-%20Comunicaci%C3%B3n%20digital%20Inal%C3%A1mbrica%20de%20baja%20velocidad%20en%20la%20banda%20ISM%20de%202.4%20GHz.pdf>

Noorani, N. (2012). *Wireless Multi-Sensor Monitoring System Utilizing IEEE 802.15.4 Communication Standards for Water Leakage Detection*.

- Obtenido de https://www.ieee.org/education_careers/education/standards/wireless_multi_sensor_monitoring_system_utilizing_ieee_csun.pdf
- Ortiz Torres, A. (2011). *Técnicas de enrutamiento inteligentes para redes de sensores inalámbricas*. Obtenido de Universidad de Castilla - La Mancha: <https://ruidera.uclm.es/xmlui/bitstream/handle/10578/2263/TESIS%20Ortiz%20Torres.pdf?sequence=1>
- Park, J., Stojmenovic, I., Young, H., & Yi, G. (2015). *Computer Science and its Applications: Ubiquitous Information Technologies*. DOI: 10.10007/978-3-662-45402-2 Berlín. Alemania: Springer.
- Polimeni, M., & Richi, M. (2004). *Aprendizaje por Refuerzo: Algoritmo Q – Learning*. Obtenido de <http://www-2.dc.uba.ar/materias/robotica/TPFinal2004/RachiPolimeni.pdf>
- Querol, J. (2011). *Estudio y evaluación de prestaciones de redes inalámbricas de sensores*. Obtenido de Repositorio Digital Universidad de Valencia: https://riunet.upv.es/bitstream/handle/10251/10354/Proyecto_Querol.pdf
- Revolvy. (2014). *Zigbee*. Obtenido de https://www.revolvy.com/main/index.php?s=Zigbee&item_type=topic
- Robles, M., Marrone, L., & Díaz, J. (2010). *Comparación de protocolos de ruteo en redes Ad Hoc*. Obtenido de XVI Congreso Argentino de Ciencias de la Computación: http://sedici.unlp.edu.ar/bitstream/handle/10915/19343/Documento_completo.pdf?sequence=1
- Sánchez. (2010). *Clasificación de los métodos de Enrutamiento*. Obtenido de http://www.profesores.frc.utn.edu.ar/sistemas/ingsanchez/Redes/Archivos/enrutam.asp#_Toc522523405
- Sarkar, A., & Murugan, S. (2016). *Routing protocols for wireless sensor networks: What the literature says?* Obtenido de Elsevier Volumen 55, Número 4: <http://www.sciencedirect.com/science/article/pii/S1110016816302125>

- Sierra, M. (2009). *Mejora de Algoritmos de búsqueda Heurística mediante Poda por Dominancia. aplicación a problemas de Scheduling*. Obtenido de Tesis Doctoral. Universidad de Oviedo: http://digibuo.uniovi.es/dspace/bitstream/10651/15043/1/TD_M%20Rita%20Sierra%20Sanchez.pdf
- Syed, H., Piyush, C., & Tahsin, U. (2016). *Performance Evaluation of AODV Protocol for Energy Consumption and QoS in IEEE 802.15.4 Based Wireless Sensor Network Using QualNet Simulator*. Obtenido de http://file.scirp.org/Html/2-9501451_69763.htm
- Tennina, S. (2013). *IEEE 802.15.4 and ZigBee as Enabling Technologies for Low-Power Wireless for Low Power Wireless System with Quality of service constraints*. Londres: Springer.
- TinyOS. (2013). *TOSSIM*. Obtenido de <http://tinycos.stanford.edu/tinycos-wiki/index.php/TOSSIM>
- Wang, C., T. J., & Q. Z. (2014). *ZigBee® Network Protocols and Applications*. Boca Ratón. USA, p-p 233.: CRC Press.
- WANG, X. (2015). *Artificial Intelligence-Based Techniques for Emerging Heterogeneous Network: State of the Arts, Opportunities,*. Obtenido de <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7185326>
- Woo, M., & Singh, S. (2008). *Power-Aware Routing in Mobile Ad Hoc Networks, de Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*. ACM Press.

GLOSARIO DE TÉRMINOS

ACO-QoS Ants Colony Optimization based QoS Routing algorithm

AODV Ad hoc On Demand Distance Vector

CDRQ-Routing Confidence-based Dual Reinforcement Q-Routing

CQ-Routing Confidence-based Q-Routing

CSMA Carrier Sense Multiple Access

CTS Clear-To-Send

DRQ-Routing Dual Reinforcement Q-Routing

DSR Dynamic Source Routing

FROMS Feedback ROuting to Multiple Sinks

GA-Routing Genetic Algorithm Routing

GUI Graphical User Interface

IC Inteligencia Computacional

LEACH Low Energy Adaptive Clustering Hierarchy

LQI Link Quality Indicator

MAC Medium Access Control

MTE Minimum Total Energy

NED Network Description Language

NS-3 Network Simulator 3

nst-AODV not so tiny Ad hoc On Demand Distance Vector

OMNeT++ Objective Modular Networks Testbed

QoS Quality of Service

RADV Route Advisor

REL Routing by Energy and Link Quality

RREP Route Reply

RREQ Route Request

RSSI Received Signal Strength Indication

RTS Request-To-Send

SIR Sensor Intelligence Routing

SNR Signal-Noise-Ratio

SOM Self-Organized Map

SPIN Sensor Protocol for Information via Negotiation

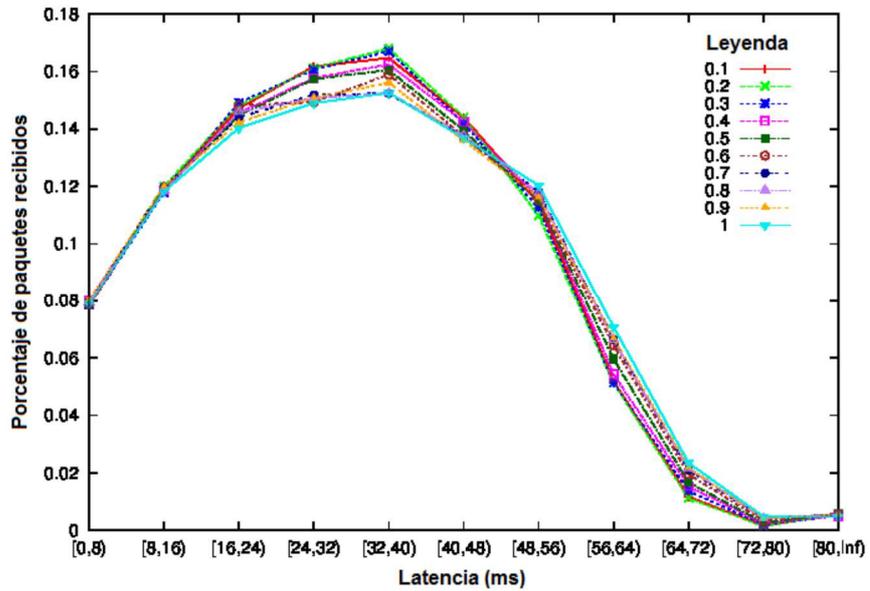
TDMA Time Division Multiple Access

T-MAC Timeout-MAC

WSN Wireless Sensor Network

ANEXO 1

Se muestra el Histograma de latencia para velocidades de aprendizaje entre 0.1 y 1. Los requerimientos de latencia y tasa de recepción vienen dadas por la aplicación para la cual se diseña la WSN.





DECLARACIÓN Y AUTORIZACIÓN

Yo, **Esparza Abad, Julio Armando**, con C.C: # 1200646683 autor del trabajo de titulación: **Análisis y evaluación de una plataforma de monitoreo y control utilizando algoritmos de enrutamiento para obtener el menor costo energético en redes Zigbee**, previo a la obtención del título de **Magister en Telecomunicaciones** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 24 de Agosto del 2017

f. _____

Nombre: **Esparza Abad, Julio Armando**

C.C: **1200646683**



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Análisis y evaluación de una plataforma de monitoreo y control utilizando algoritmos de enrutamiento para obtener el menor costo energético en redes Zigbee		
AUTOR(ES)	Esparza Abad, Julio Armando		
REVISOR(ES)/TUTOR	MSc. Orlando Philco Asqui; MSc. Luis Córdova Rivadeneira / MSc. Manuel Romero Paz		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Sistema de Posgrado		
PROGRAMA:	Maestría en Telecomunicaciones		
TÍTULO OBTENIDO:	Magister en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	24 de Agosto de 2017	No. DE PÁGINAS:	91
ÁREAS TEMÁTICAS:	Redes inalámbricas, Redes de área personal.		
PALABRAS CLAVES/ KEYWORDS:	AODV, REL, FROMS, Zigbee, WSN		
RESUMEN/ABSTRACT:	<p>En el presente trabajo se caracteriza y evalúa el consumo de energía por comunicación en una red IEEE 802.15.4 conocida como Zigbee, dicha red opera en cortas distancias. Como problemática surge determinar o seleccionar el protocolo de enrutamiento eficiente para el cumplimiento del monitoreo y transmisión de datos entre diversos nodos que conforma la red Zigbee. La metodología es descriptiva-analítica, se emplea la metodología empírica por el uso de un programa simulador para valorar o cuantificar el protocolo de enrutamiento eficiente. Se comparan dos tipos de protocolos de enrutamiento, uno basado en Routing by Energy and Link Quality, REL y otro basado en Feedback ROuting to Multiple Sinks, FROMS, las evaluaciones indican que este último es más eficiente en un 17% en su tarea de censado, procesamiento y análisis de datos. Se concluye que los protocolos de ruteo basados en inteligencia computacional, puede permitir un desempeño o comportamiento inteligente a una red Zigbee o WSN.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-0999614885	E-mail: armando_esparza@hotmail.com	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::	Nombre: Romero Paz Manuel de Jesús		
	Teléfono: +593-994606932		
	E-mail: manuel.romero@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			



**Presidencia
de la República
del Ecuador**



**Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes**



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	