

TRABAJO ESPECIAL DE GRADO

PROPUESTA DE UN ESQUEMA DE APAGADO Y ENCENDIDO DE NODOS PARA MEJORAR EL DESEMPEÑO ENERGETICO DEL PROTOCOLO DE ENRUTAMIENTO CTP (COLLECTION TREE PROTOCOL)

Tutor Académico: Prof. Carolina Regoli

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Andrés E., Mujica R.
para optar al título de
Ingeniero Electricista

Caracas, 20

CONSTANCIA DE APROBACIÓN

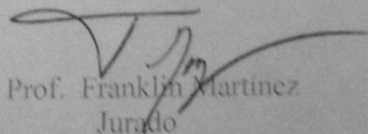
CONSTANCIA DE APROBACIÓN

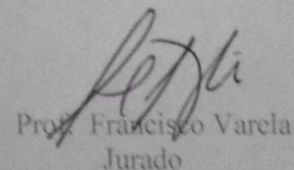
Caracas, 23 de noviembre de 2017

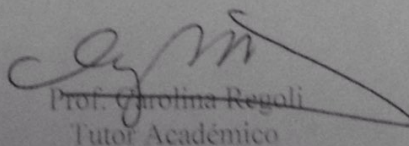
Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Andrés E. Mujica R., titulado:

“PROPUESTA DE UN ESQUEMA DE APAGADO Y ENCENDIDO DE NODOS PARA MEJORAR EL DESEMPEÑO ENERGÉTICO DEL PROTOCOLO DE ENRUTAMIENTO CTP (COLLECTION TREE PROTOCOL)”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.


Prof. Franklin Martinez
Jurado


Prof. Francisco Varela
Jurado


Prof. Carolina Regoli
Tutor Académico

*PROPUESTA DE UN ESQUEMA DE APAGADO Y ENCENDIDO DE NODOS
PARA MEJORAR EL DESEMPEÑO ENERGÉTICO DEL PROTOCOLO DE
ENRUTAMIENTO CTP (COLLECTION TREE PROTOCOL)*

Tutor Académico: Carolina Regoli. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Comunicaciones. Institución: Universidad Central de Venezuela. 2016. 63 h. + anexos.

Palabras Claves: Redes inalámbricas de sensores, CTP, enrutamiento, Castalia

Resumen. Las redes inalámbricas de sensores se han vuelto importantes en la sociedad actual, su integración en las aplicaciones cotidianas ha permitido facilitar la investigación, la medicina, entre otros. Sin embargo, las restricciones de procesamiento de información y energéticas son una gran limitante a la hora de su implementación, por lo que el desarrollo de nuevos protocolos de enrutamiento que optimicen el uso de recursos y energía de las redes de este tipo es de gran importancia. Actualmente existen distintos tipos de herramientas de simulación para desarrollar estos protocolos, entre las cuales se puede mencionar el simulador Castalia, utilizado en este trabajo. Este simulador permite desarrollar y probar la factibilidad de protocolos de enrutamiento para redes de sensores inalámbricos. En Castalia existe una implementación del protocolo CTP; (*Collection Tree Protocol*) que ha sido ampliamente utilizada, pero que posee limitaciones energéticas considerables. Este trabajo busca mejorar el desempeño energético del CTP mediante un algoritmo de apagado y encendido de nodos que logre mantener sus características principales como la adaptación del enrutamiento, el manejo de la congestión y eliminación de bucles, pero se contrarreste la pérdida de energía por el envío de paquetes excesivo y la permanencia del radio en estado activo. La evaluación de la propuesta planteada en este trabajo se realizó en 3 escenarios de prueba, donde se comparó su desempeño con respecto al del protocolo original. Posteriormente se analizaron los resultados para establecer conclusiones de la factibilidad del diseño de la mejora.

DEDICATORIA

A mi UCV y mi país, su existencia fue vital para llegar hasta aquí, siempre estaré agradecido y no hay maneras de demostrar cuánto.

Gracias.

AGRADECIMIENTOS

A mi país Venezuela que me permitió tener una educación gratuita y de calidad, así como un hospital en la universidad y un cuerpo de bomberos que siempre prestaron servicio gratuito y de calidad.

A la Universidad Central de Venezuela que es mi segunda casa, donde me crié, estudié y trabajé, me brindó conocimientos mucho más allá de mi carrera e incluso me alimentó de manera gratuita, es de gran importancia recalcar que ninguna universidad del mundo ofrece estos servicios de esta manera.

A la facultad de ingeniería y a la escuela de ingeniería eléctrica y sus profesores que brindan un apoyo importante al estudiante e imparten conocimientos sin importar la problemática que acontezca el país, se puede decir que la lección de vida más importante la aprendí aquí y consiste en seguir adelante sin importar que, debido a que la formación y el aprendizaje es más importante que la adversidad.

A mi tutora Carolina Regoli, la cual admiro y me ha prestado su apoyo incondicional, sin ella no hubiese podido llegar hasta aquí.

A la profesora Mercedes Arocha que me ha escuchado y me ha brindado su apoyo y consejos de vida para seguir adelante.

A mis compañeros de clase y amigos, a toda la gente que conocí en la facultad, que estudiando juntos aprendimos que es más fácil apoyarse el uno al otro y me prestó ayuda cuando más lo necesité.

A Jessica que fue mi apoyo y mano derecha es esta carrera por lo cual estaré eternamente agradecido sin importar las circunstancias.

A mi familia que siempre confió en mí y me apoyó incondicionalmente cuando lo necesité, sobre todo a mis tías y mi tío que me brindaron todo su apoyo.

A mi mamá que a pesar de nuestras diferencias siempre me ha brindado su apoyo y su aporte emocional para culminar la carrera fue esencial para mí.

A mis perros que a pesar de no entender nada, no se apartaron de mí en todas las noches de estudio y traspasado desde que inicié la carrera.

INDICE GENERAL

CONSTANCIA DE APROBACIÓN	I
DEDICATORIA.....	III
AGRADECIMIENTOS	IV
INDICE DE TABLAS	VII
INDICE DE FIGURAS.....	VIII
ACRONIMOS Y ABREVIATURAS.....	IX
INTRODUCCIÓN.....	1
CAPITULO I	3
1 FUNDAMENTOS DE LA INVESTIGACION	3
1.1 PLANTEAMIENTO DEL PROBLEMA	3
1.2 OBJETIVOS.....	3
2. MARCO REFERENCIAL.....	4
1.4 METODOLOGIA	5
CAPITULO II.....	7
2.1 REDES INALÁMBRICAS DE SENSORES.....	7
2.2 ARQUITECTURA DE COMUNICACIÓN DE LAS REDES INALÁMBRICAS DE SENSORES (WSN)	10
2.3. SIMULADOR OMNET++ (OBJECTIVE MODULAR NETWORK TESTBED IN C++)	11
2.3.1 <i>El MODELO OMNeT++</i>	12
2.4 SIMULADOR CASTALIA.....	13
2.5 ENRUTAMIENTO	13
2.6 CONSIDERACIONES PARA LA EFICIENCIA ENERGÉTICA	15
2.7 PROTOCOLOS DE ENRUTAMIENTO	15
2.7.1 <i>CONSUMO ENERGÉTICO EN PROTOCOLOS DE ENRUTAMIENTO PARA WSN</i>	19
2.7.2 <i>CTP</i>	20
CAPITULO III	24
3 <i>COLLECTION TREE PROTOCOL</i> – “PROTOCOLO DE RECOLECCIÓN EN ÁRBOL” (CTP) IMPLEMENTADO EN EL SIMULADOR CASTALIA 3.3	24
3.1 PAQUETES DE ENRUTAMIENTO Y DE DATOS DEL PROTOCOLO CTP IMPLEMENTADO EN CASTALIA 3.3.....	25
<i>Encabezado PHY (capa física) y MAC (capa de acceso control medio)</i>	26
<i>Marco de datos en CTP</i>	26
<i>Marco de enrutamiento en CTP</i>	27

<i>ACK (paquetes de reconocimiento)</i>	28
3.2 MÓDULO DE ESTIMADOR DE ENLACE	28
<i>Encabezado y pie del estimador de enlace</i>	28
<i>Calculo de 1 salto EXT</i>	29
<i>Procedimiento de inclusión y exclusión de nodos en la tabla del módulo de estimador de enlace</i>	31
3.3 MÓDULO DEL MOTOR DE ENRUTAMIENTO	32
<i>Frecuencia de envío de mensajes beacon</i>	32
<i>Actualización de tabla de rutas</i>	33
<i>Selección de los nodos padres o raíz</i>	33
3.4 MÓDULO DEL MOTOR DE REENVÍO	34
<i>Paquetes recolectados y retransmisiones</i>	34
<i>Bandera de congestión</i>	34
<i>Paquetes duplicados</i>	35
<i>Bucles en ruta</i>	35
<i>Mecanismo de fisgoneo</i>	35
<i>Temporizadores de tiempo de espera</i>	36
3.5 MÓDULO DE BUFFER DUAL	36
3.6 IMPLICACIONES DEL MÓDULO MAC DEL PROTOCOLO DE RECOLECCIÓN EN ÁRBOL REALIZADO PARA EL SIMULADOR CASTALIA	37
3.7 MÓDULO CC2420MAC	37
<i>ACK en la capa de enlace</i>	38
<i>Propagación de la información en la capa de enlace</i>	39
<i>Fisgoneo de paquetes</i>	39
<i>Mecanismo de división de fase de TinyOS</i>	39
<i>Encabezado MAC</i>	39
CAPITULO IV	42
4 PROPUESTA DE ESQUEMA DE APAGADO Y ENCENDIDO DE NODOS PARA EL PROTOCOLO CTP PROGRAMADO EN EL SIMULADOR CASTALIA 3.3	42
4.1 DISEÑO DE UN ESQUEMA DE APAGADO Y ENCENDIDO PARA EL PROTOCOLO DE RECOLECCIÓN EN ÁRBOL PARA SIMULADOR CASTALIA	42
4.2 PROGRAMACIÓN DE LA PROPUESTA DE ESQUEMA DE APAGADO Y ENCENDIDO	45
4.3 ESCENARIOS DE PRUEBA	46
4.4 ANALISIS DE RESULTADOS	47
CONCLUSIONES Y RECOMENDACIONES	59
REFERENCIAS BIBLIOGRAFICAS	61
BIBLIOGRAFIA	63
ANEXOS	64

INDICE DE TABLAS

Tabla 1. VALORES DE LOS PARÁMETROS IMPLEMENTADOS EN EL MÓDULO DEL ESTIMADOR DE ENLACE	31
Tabla 2. VALORES DE LOS PARÁMETROS RELEVANTES DEL MÓDULO DEL MOTOR DE ENRUTAMIENTO	33
Tabla 3. VALORES RELEVANTES DE LOS PARÁMETROS DEL MÓDULO DEL MOTOR DE REENVÍOS	36
Tabla 4. VALORES RELEVANTES DE LOS PARÁMETROS DEL MÓDULO MAC	38
Tabla 5. NODOS CTP PRIMER ESCENARIO	49
Tabla 6. NODOS MEJORA CTP PRIMER ESCENARIO	49
Tabla 7. COMPARACION PRIMER ESCENARIO	50
Tabla 8. NODOS CTP SEGUNDO ESCENARIO	52
Tabla 9. NODOS MEJORA CTP SEGUNDO ESCENARIO	52
Tabla 10. COMPARACION SEGUNDO ESCENARIO	53
Tabla 11. NODOS CTP TERCER ESCENARIO	54
Tabla 12. NODOS MEJORA CTP TERCER ESCENARIO	55
Tabla 13. COMPARACION TERCER ESCENARIO	55
Tabla 14 Comparación CTP vs MEJORA a diferentes tasas de paquetes por segundo	57

INDICE DE FIGURAS

Figura 1. ESQUEMA BÁSICO DE UNA RED INALÁMBRICA DE SENSORES	8
Figura 2. ELEMENTOS QUE COMPONEN EL NODO EN LA RED.....	8
Figura 3. ARQUITECTURA DE COMUNICACIÓN DE LAS WSN	10
Figura 4. CLASIFICACIÓN DE PROTOCOLOS DE ENRUTAMIENTO	16
Figura 5 DIAGRAMA DE FLUJO - FUNCIONAMIENTO CTP	23
Figura 6. COMUNICACIÓN DEL MÓDULO COMPUESTO CTPNOE CON MÓDULOS EXTERNOS	24
Figura 7. COMUNICACIÓN DE LOS MODULOS INTERNOS EN EL MÓDULO CTPNOE [2]	25
Figura 8. ESTRUCTURA DE LOS PAQUETES DE ENRUTAMIENTO, DATOS Y RECONOCIMIENTO (ACK) EN CTP [2]	26
Figura 9. ENCABEZADOS, MARCO DE DATOS Y MARCO DE ENRUTAMIENTO DEL ESTIMADOR DE ENLACE	27
Figura 10. COMUNICACIÓN CTP CON LA CAPA MAC [2].....	40
Figura 11. ALGORITMO DE APAGADO-ENCENDIDO DE LA RADIO DEL NODO PARA EL ENRUTAMIENTO DE LOS DATOS.....	44
Figura 12 ESCENARIO 1.....	48
Figura 13. ESCENARIO 2.....	51
Figura 14. ESCENARIO 2.....	51
Figura 15. ESCENARIO 3.....	54

ACRONIMOS Y ABREVIATURAS

ACK Acknowledgement

ACQUIRE Active Query Forwarding in Sensor Networks

BAN Wireless Body Area Network

CH Cluster Head

CTP Collection Tree Protocol

CPU Central Processing Unit

ETX Expected Transmissions

FIFO First in, First out

FSK Frequency Shift Keying

TTD Two -Tier Data Dissemination

Gbps Giga bits por segundo

GHz Giga Hertz

Hz Hertz

IEEE Institute of Electrical and Electronics Engineers

J Joules

LA Local Aggregator

LEACH Low Energy Adaptive Clustering Hierarchy

MA Master Aggregator

MAC Media Access Control

m Metros

Mbps Mega bits por segundo

MHz Mega Hertz

OMNet++ Objective Modular Network Test-bed in C++

OSI Open System Interconnection

PEGASIS Power-Efficient Gathering in Sensor Information Systems

QoS Quality of Service

SAR Sequential Assignment Routing

SINR signal to noise plus interference ratio

SOP Self Organizing Protocol

SPIN Sensor Protocol for Information via Negotiation

TCP Transmission Control Protocol

UDP User Datagram Protocol

VGA Virtual Grid Architecture

WSN Wireless Sensor Network

INTRODUCCIÓN

En la actualidad, las redes inalámbricas son muy importantes en el área tecnología, ya que se pueden usar en muchas aplicaciones que permiten vigilar variables (temperatura, luz, luminosidad, humedad, consumo energético), información efectiva en muchas áreas, facilitando datos para investigaciones de manera menos costosa.

Una red inalámbrica de sensores es un conjunto de dispositivos distribuidos en un espacio, que posee sensores que recogen variables físicas o ambientales para enviarlas a un centro de recolección de datos, con la finalidad de monitorizar, controlar y eventualmente tomar acciones en base a esos datos.

Las redes inalámbricas de sensores han evolucionado en los últimos años, desarrollándose, entre otros aspectos, gran cantidad de algoritmos y protocolos con la finalidad de lograr de manera óptima la transmisión y recolección de datos. En este sentido la utilización de simuladores es de gran importancia a la hora de diseñar una red de este tipo, ya que son capaces de reproducir el comportamiento de un sistema, simulando experiencias que pueden llegar a suceder o pueden estar cercanas a la realidad.

Para el desarrollo de nuevos algoritmos o aplicaciones de las redes inalámbricas de sensores existe gran variedad de simuladores, siendo uno de los más usados por desarrolladores de protocolos para redes inalámbricas de sensores y que ha demostrado muy buen desempeño el Castalia, que se basa en un entorno de simulación OMNeT++. Castalia es un simulador principalmente utilizado para la investigación y desarrollo de prototipos, algoritmos distribuidos y protocolos mediante el uso de modelos realistas de canales inalámbricos y acceso de radio; usa un modelo logarítmico para las pérdidas por propagación y de las variaciones temporales en la señal, para emular el desvanecimiento que ocurre en algunos entornos físicos. El simulador es capaz de modelar procesos físicos, ruido, protocolos MAC y de enrutamiento, así como derivas en el reloj de los nodos de la red.

Entre los protocolos de enrutamiento que se han desarrollado en los últimos años se encuentra el protocolo CTP (*Collection Tree Protocol*). Este protocolo es una referencia en recolección de datos debido a su eficiencia y rendimiento con respecto a otros protocolos, en

relación a la comunicación de datos y ubicación entre nodos (sensores) en una red. Sin embargo, una de sus limitaciones es el desgaste energético de los nodos, que limita la vida útil de la red. Castalia dentro de sus librerías no incluye el módulo del protocolo CTP por lo cual se debe añadir, este módulo se encuentra disponible gratuita y públicamente en la página: <http://code.google.com/p/ctp-castalia/>

En el presente trabajo se realizó un estudio del protocolo CTP programado en el simulador Castalia, y se planteó e implementó una propuesta para mejorar su desempeño energético, a partir de la implementación de un esquema de apagado y encendido de los nodos de la red.

La estructura de este trabajo es la siguiente: En el capítulo 1 están los fundamentos de la investigación, sus objetivos, alcances y metodología utilizada. En el capítulo 2 se encuentra una base teórica del funcionamiento y generalidades de las redes inalámbricas de sensores para ofrecer un mejor entendimiento de la investigación. El capítulo 3 presenta una descripción de la implementación y funcionamiento del protocolo de recolección en árbol (CTP) para el simulador Castalia. Seguidamente, en el capítulo 4 se presenta la propuesta planteada, así como los escenarios en los cuales se validó la propuesta, junto con el análisis de los resultados obtenidos. Por último, se presentan algunas conclusiones y recomendaciones derivadas del trabajo desarrollado.

CAPITULO I

1 FUNDAMENTOS DE LA INVESTIGACION

1. PLANTEAMIENTO DEL PROBLEMA

Las redes inalámbricas de sensores desempeñan un rol importante en la sociedad debido a sus capacidades infinitas de uso, ya que están compuestas por sensores de bajo costo capaces de recoger variables (temperatura, humedad, etc.) y transmitir las con dispositivos de baja potencia. Estos sensores están acompañados de transmisores y receptores de radio de baja potencia para comunicarse entre sí, pero tienen baterías, lo cual genera una problemática energética. Esto obliga a realizar mejoras en los protocolos de enrutamiento para optimizar el uso de la energía, logrando ampliar el tiempo de vida de la red.

En este trabajo se pretende incluir un esquema de encendido y apagado en el protocolo CTP, ya programado en el simulador Castalia, con el fin de mejorar su desempeño, buscando prolongar la vida útil de una red inalámbrica de sensores.

1.2 OBJETIVOS

OBJETIVO GENERAL

Desarrollar una propuesta de un esquema de encendido y apagado de nodos para mejorar el desempeño energético del protocolo de enrutamiento CTP (Collection Tree Protocol) para redes inalámbricas de sensores (WSN), de acuerdo con el rol de los nodos en la red.

OBJETIVOS ESPECÍFICOS

- Realizar el estado del arte sobre las redes inalámbricas de sensores, aplicaciones, ventajas, limitaciones y protocolos de enrutamiento.
- Documentar el funcionamiento del protocolo de enrutamiento CTP.
- Aprender el manejo y uso del simulador Castalia.

- Estudiar los diferentes modos de funcionamiento de los nodos sensores de una red inalámbrica de sensores
- Desarrollar esquema de ahorro energético de los nodos de una red inalámbrica de sensores, en función del rol de cada uno en la misma.
- Desarrollar el algoritmo de la mejora del protocolo CTP.
- Programar la mejora energética del protocolo CTP en el simulador Castalia.
- Definir diferentes escenarios de prueba en el simulador Castalia para evaluar el desempeño del protocolo CTP modificado.
- Realizar simulaciones sobre los escenarios definidos para validar el esquema desarrollado
- Comparar el protocolo CTP original con la mejora del protocolo.
- Validar los resultados obtenidos en comparación con resultados teóricos esperados.
- Analizar los resultados obtenidos.

2. MARCO REFERENCIAL

En Venezuela en los últimos años se han venido desarrollando algunos trabajos en el área de redes inalámbricas de sensores, que pueden clasificarse en varias áreas: aplicaciones, metodología de diseño de redes, desarrollo de protocolos de enrutamiento y finalmente estudio del desgaste energético y de escenarios de propagación.

Se puede destacar el trabajo de Sanjuan y Yuri [13], quienes plantearon un sistema de detección temprana de incendios forestales, así como el de Castillo-Effer et al. [14], orientado a la detección de inundaciones repentinas.

En el área del diseño de redes se consiguió el trabajo desarrollado por Pérez et al. [15], que plantean una metodología para el diseño de redes WSN. Así mismo, se han hecho esfuerzos desarrollando interfaces para monitorización de este tipo de redes [16].

Existen algunas publicaciones de protocolos de enrutamiento desarrollados por investigadores venezolanos [17] [18] [9], en los que se considera el desgaste energético para mejorar el desempeño de la red. Igualmente se ha estudiado el desgaste y optimización de energía en los nodos de las redes inalámbricas de sensores [19] [20].

Por otra parte, encontraron algunos trabajos donde se han estudiado diferentes escenarios de propagación [21] [22], midiendo variables de acuerdo a esos escenarios.

Finalmente, en cuanto a desarrollo de simuladores no existen grandes avances en el país, sin embargo, en [23] se observa un breve estudio de algunos de los simuladores de WSN y un desarrollo de un simulador básico para protocolos de enrutamiento de este tipo de redes.

1.4 METODOLOGIA

Fase 1. Documentación sobre las redes inalámbricas

Se realizó una investigación exhaustiva sobre las redes inalámbricas, sus ventajas, aplicaciones y limitaciones. Se buscó detalladamente en diversas fuentes acerca del tema, revisando los antecedentes de protocolos de enrutamiento desarrollados, los tipos de protocolos de enrutamiento, la importancia de los protocolos de enrutamiento en WSN, ventajas, desventajas, diseños, funcionamiento, así como otros datos de interés a la hora de implementar un protocolo de enrutamiento de redes inalámbricas de sensores.

- **Fase 2. Documentación del protocolo de enrutamiento CTP**

Se investigó toda la información necesaria para entender el funcionamiento del protocolo de enrutamiento CTP [1] [2]. Se recopiló información sobre su filosofía de funcionamiento, formas de intercambio de mensajes, establecimiento de rutas y desgaste energético del mismo, entre otros.

- **Fase 3. Estudio del uso del simulador Castalia en el sistema operativo Linux.**

Se realizó un estudio de las características, herramientas y utilidades del simulador para aprender su uso, así como su instalación con sus módulos correspondientes. Igualmente se estudió el manual de usuario del simulador [3].

- **Fase 4. Estudiar el funcionamiento de los nodos en una WSN**

En esta fase se documentó sobre las diferentes formas de funcionamiento de los nodos sensores, sus capacidades de encendido y apagado, etc.

- **Fase 5. Diseño e implementación de una mejora del comportamiento energético en el protocolo CTP.**

Se diseñó un algoritmo aplicable dentro del protocolo CTP por medio del desarrollo de un esquema de encendido y apagado de los sensores para optimizar el desgaste energético.

- **Fase 6. Realización de las simulaciones en diferentes escenarios y comparación de los resultados obtenidos en cada caso.**

Una vez realizado el diseño de la mejora energética se definieron diferentes escenarios de simulación, para realizar simulaciones del protocolo CTP modificado evaluando su desempeño y comparando los resultados obtenidos con los del CTP original.

- **Fase 7. Realización del informe final.**

Tomando en cuenta cada una de las fases anteriores, se realizó el informe que contiene detalladamente toda la información sobre lo realizado en la investigación y el desarrollo del trabajo.

CAPITULO II

En el siguiente capítulo se describirán las generalidades de las redes inalámbricas de sensores, sus funciones y componentes. Se explicarán los aspectos principales del simulador con el que se trabajará en el presente trabajo, y finalmente se hará un resumen sobre el tema de los protocolos de enrutamientos para este tipo de redes, describiendo los más importantes.

2.1 REDES INALÁMBRICAS DE SENSORES

A partir de la evolución de las redes de datos, la microelectrónica y la industrialización de la producción de componentes electrónicos (semiconductores, procesadores, tarjetas e impresoras 3D) para todo público, nacieron las redes inalámbricas de sensores o WSN (en inglés Wireless Sensor Networks) como una alternativa para aplicaciones en las áreas de monitorización de variables ambientales, para crear ambientes eficientes de trabajo, reducir el impacto de las catástrofes naturales (terremotos, inundaciones, tornados, etc.), monitorización de pacientes con diversas patologías, optimización de sistemas en general a partir de sistemas de aprendizaje (algoritmos), entre otras.

Las redes inalámbricas de sensores se componen de tres bloques físicos principales, un conjunto de sensores que serán los nodos de la red, la base principal o receptor principal que gestionará los datos de la red comunicándola con una red externa y la unidad de procesamiento, almacenamiento y administración de datos; los datos obtenidos de esta red son enrutados de manera autónoma. La ventaja de esto es que diferentes usuarios pueden hacer uso de esta información para realizar diversas investigaciones o aplicaciones, ofreciendo una diversa gama de posibilidades a la hora del uso de la información.

Así mismo como se muestra en la figura 1, en vez de un sensor A que se conecta a la estación base podría ser un campo de sensores que se conecta a la misma estación base e envía información, incluso se podría conectar otra estación base aparte en otro lado del mundo con conexión a internet y que a su vez se interconecte con la otra red de sensores, dando a entender las diferentes posibilidades que se podrían encontrar con la aplicación de estas tecnologías.

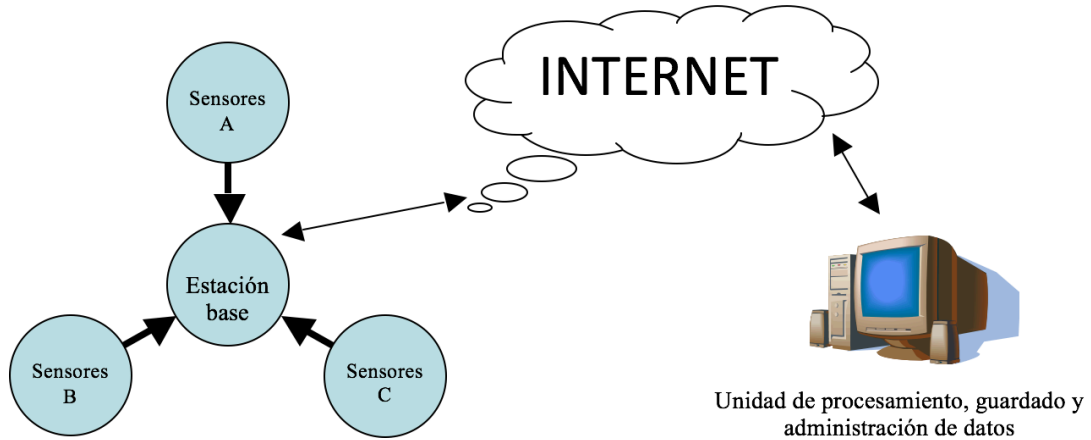


Figura 1. ESQUEMA BÁSICO DE UNA RED INALÁMBRICA DE SENSORES

Los sensores son los elementos básicos de la red, son dispositivos de bajo costo, simples y capaces de obtener información del ambiente (presión, luz, calor, etc...), sus componentes se muestran en la figura 2.

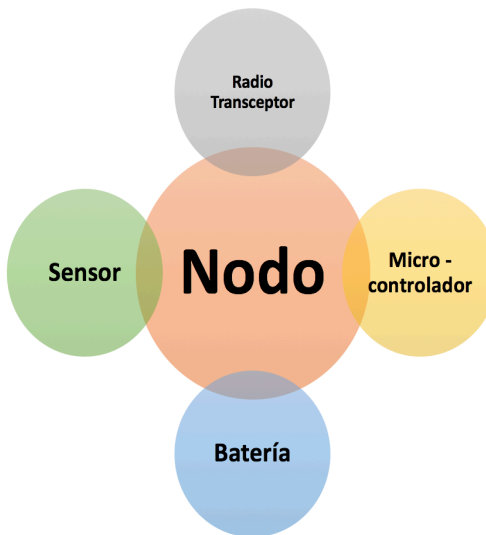


Figura 2. ELEMENTOS QUE COMPONEN EL NODO EN LA RED

Las WSN son muy similares a las redes de datos convencionales ya que pueden realizar básicamente las mismas acciones a menor escala, por eso es necesario nombrar características que las diferencien de las tradicionales como:

1. Se diseñan con un solo propósito o aplicación (ad-hoc).
2. El uso y conservación de la energía es el centro del diseño de la red.
3. Estas redes operan en ubicaciones con condiciones ambientales complicadas.
4. Normalmente acceder físicamente a estas redes de sensores es difícil o imposible.
5. Se espera el fallo y anulación de los componentes de la red y está previsto en el diseño.
6. No se puede centralizar la administración de la red, por lo que se hace de manera localizada ya que la red está sometida a cambios fortuitos e inesperados.
7. Tienen alta probabilidad de fallar.
8. Son redes con poca o ninguna intervención humana.

El funcionamiento de estas redes en sus inicios vino dado por el estándar IEEE 802.11, que ofrecía las pautas de comunicación inalámbrica iniciales donde los sensores podían intercambiar información en distintas frecuencias en la banda de 2.4 GHz (usando el estándar IEEE 802.11b/g) o la banda de 5GHz (usando el estándar IEEE 802.11a). El estándar IEEE 802.11 y sus diferentes versiones (a, b, g, n, ac, etc.), permite alcanzar altas velocidades de transmisión (hasta 54 Mbps con el estándar 802.11g) a la hora de transmitir datos. Sin embargo, las tasas de datos por segundo transmitidos por los sensores son muy bajas y comparables con la tasa de datos de los modem dial-up (56 kbps); esto significa un desperdicio innecesario de energía en el radio transmisor. Por esa razón se desarrolló un estándar apropiado para disminuir el consumo energético de estas redes, apareciendo así el estándar IEEE 802.11.1 (Bluetooth / velocidad de 1Mbps) y el IEEE802.15.4 ZigBee/ IEEE802.15.4, donde hacen vida las redes inalámbricas personales de bajas tasas de datos (LW-WPANS). Las características generales de importancia que se pueden mencionar para esta investigación son:

- Se define hasta 16 canales entre las frecuencias (2400 - 2483.5 MHz)
- Uso de transmisiones mediante MAC (*medium access control*), lo cual permite que los nodos bajen el consumo energético usando el modo *Sleep* (dormir) donde se apaga la radio de los sensores mientras no esté en uso.
- Topologías de red: tipo estrella o punto a punto.
- Arquitectura de transporte de datos donde las tramas más usadas son del tipo: data, ACK (*acknowledgment*= Reconocimiento), *beacon* y tramas del comando MAC.

- Seguridad y confiabilidad mediante protocolos como CSMA/CA, uso de retransmisión basado tiempo de espera, lista de acceso, entre otras.
- Velocidad estándar de 250 kbps.
- Permite alrededor de 64000 nodos por red.
- Permite transportar pequeños paquetes de datos.
- Alcance de 70 a 100 m según el estándar del radio que se esté usando (ZigBee IEEE 802.15.4 por ejemplo).
- Permite una duración de batería aproximada entre 100 y 1000 días dependiendo del protocolo usado y el uso del nodo.
- Para uso de control y monitorización de variables.

2.2 ARQUITECTURA DE COMUNICACIÓN DE LAS REDES INALÁMBRICAS DE SENSORES (WSN)

Siguiendo el modelo de interconexión de sistemas abierto (modelo OSI), las WSN se componen de 5 capas principales y 3 de capas de administración de la red las cuales optimizan el sistema. Como se puede observar en la figura 3.

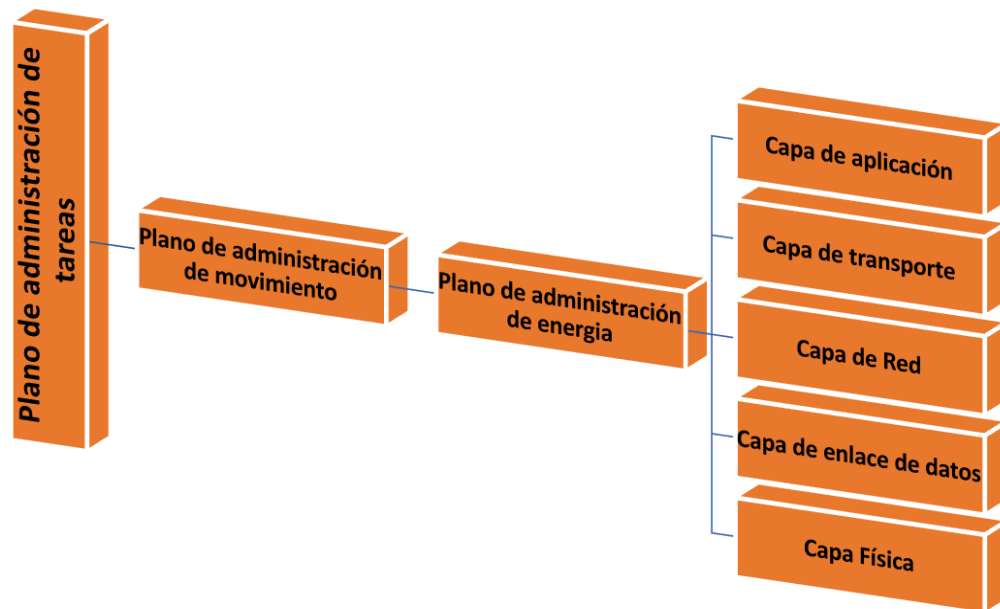


Figura 3. ARQUITECTURA DE COMUNICACIÓN DE LAS WSN

Este modelo permite un entendimiento amplio de estas redes, se observa cómo la administración de las tareas, energía y movimiento dominan la comunicación de la información a través de las capas. Esto es de suma importancia ya que debido al ambiente donde se encuentran estas redes se vuelve imposible ayudarlas físicamente, por lo que la energía se vuelve un recurso indispensable y limitado, siendo esta la base para administrar la red.

2.3. SIMULADOR OMNET++ (OBJECTIVE MODULAR NETWORK TESTBED IN C++)

Es un simulador orientado a objetos, basado en C++, para redes de comunicaciones, multiprocesadores y sistemas paralelos disponible desde 1997. Está desarrollado en *software* libre y es gratuito para fines académicos o actividades que no generen beneficios económicos. La motivación de su desarrollo fue crear un simulador con fines académicos orientado a la investigación y desarrollo de simulaciones por computadoras y sistemas paralelos. Está disponible para diversas plataformas: Windows, Mac, Linux y algunos de sus derivados.

Posee una estructura básica y librerías de simulación modular basada en lenguaje C++ creada para diseñar simuladores de redes de todo tipo, ya sean inalámbricas o alámbricas; permite añadir herramientas, ambientes gráficos; también provee de modelos de arquitectura de red, así como la capacidad de permitir agregar extensiones que permitan hacer simulaciones en tiempo real. La integración de bases de datos y todos los componentes están programados en C++. Provee la maquinaria básica y herramientas para realizar la simulación en vez de otorgar directamente los componentes de la misma; sin embargo, para muchas áreas de aplicación, existen componentes ya predeterminados que se pueden usar como los componentes de redes TCP/IP, tales como los módulos de los protocolos UDP, TCP, IP, IPv6, ARP, Ethernet, etc.

Fue diseñado para cumplir los siguientes requerimientos principales: permitir simulaciones a larga escala donde los modelos de simulación necesitan ser jerárquicos y hechos de componentes reusables tanto como se pueda; el software de simulación debe ser modular, adaptable, permitir mezclar simulaciones en aplicaciones que permitan la planificación de un red de datos; la interfaz de datos debe ser abierta y debe permitir generar y procesar archivos de entradas y salidas con herramientas del software, proveer e integrar un ambiente de desarrollo que ofrezca facilidades de modelos de desarrollo y análisis de resultado.

2.3.1 EL MODELO OMNeT++

Es el esquema en el que está programado el simulador, está basado en la integración de módulos; los módulos activos se llaman módulos simples y cuando se agrupan se llaman módulos compuestos de manera que los niveles de jerarquía son ilimitados. El uso de los módulos simples y compuestos permite generar módulos del tipo complejo; todos se implementan como objetos en lenguaje C++ utilizando las funciones que vienen de la librería del simulador. La topología de la conexión del módulo usa lenguaje NED.

El lenguaje NED (*Network Description*), es un lenguaje de declaración, definición, descripción y red de módulos, se realizan las declaraciones y descripciones de las interfaces, las puertas y los parámetros, así como las interconexiones entre módulos. La definición de la red son módulos compuestos que se califican como modelos de simulación auto contenidos.

Al modelo OMNeT++ lo definen las siguientes características:

- Los módulos se comunican por mensajes que tienen información arbitraria y atributos usuales como fecha y hora y se envían por “puertas”.
- Las puertas son las interfaces de entradas y salidas. Se pueden agregar parámetros en los módulos, permitiendo definir su topología.
- Los módulos simples se programan en C++ mediante el uso de la librería de clase de simulación de OMNeT++; esto permite que puedan reescribirse, ejecutarlos y depurarlos sin abandonar el simulador. Estos son elementos activos y no pueden ser divididos.
- El modelo del comportamiento (parte lógica de la simulación), corresponde a los archivos en código C++.
- El modelo topológico (parte gráfica) de los parámetros que lo definen están escritos en archivos NED.
- Es posible darle seguimiento a las interacciones de los módulos mientras la simulación se está ejecutando.

2.4 SIMULADOR CASTALIA

Es un simulador para redes inalámbricas de sensores (WAN) y redes de cuerpo (BAN) y en general redes de bajo consumo de energía; es capaz de simular diferentes tipos de comunicación inalámbrica. Está basado en la plataforma modular de OMNeT++, es usado por investigadores debido a que es de distribución libre y gratuita. Es capaz de simular redes de amplio alcance, escrito en lenguaje C++ y diseñado para que el usuario pruebe algoritmos o protocolos ya desarrollados o nuevos con el objetivo de evaluar su factibilidad antes de realizar pruebas en otra plataforma de nodos específica. Las características llamativas de este simulador son:

- Modelo de canal avanzado: Aquí se definen las pérdidas del canal en el mapa con conexiones difíciles entre nodos, variaciones temporales de pérdidas en el camino, movilidad en los nodos; la interferencia se maneja por medio de la potencia de la señal.
- Modelo avanzado de radios de comunicaciones de bajo consumo que permite: Probabilidad de recepción basada en SINR (relación señal a ruido más interferencia), tamaño de los paquetes, tipo de modulación (FSK y PSK), usando la curva SNR-BER, múltiples niveles potencias de transmisión en nodos distintos, estados con diferentes niveles de consumo y retraso entre ellos y sensor de portadora flexible.
- Capacidades del modelo de medición avanzado: Modelo del proceso físico altamente flexible, ruido del dispositivo sensor, bias y consumo de energía, consumo de energía del CPU, disponibilidad del uso de MAC y protocolos de enrutamiento, diseñado para expandirse y adaptarse debido a la capacidad de añadirle módulos. Como está basado en el simulador OMNeT++ es sencillo añadirle y diseñar otros algoritmos y protocolos e incluirlos con facilidad.

El simulador Castalia no trae sensores específicos, debido a que solo simula ambientes de redes inalámbricas de sensores generales con el objetivo de validar algoritmos y protocolos.

2.5 ENRUTAMIENTO

El proceso de selección de un camino en una red, elegir por donde enviar tráfico de red, se llama enrutamiento. En redes inalámbricas de sensores el enrutamiento es distinto al generalmente

usado, debido a que no existe infraestructura, los enlaces no son confiables y los nodos tienen batería limitada.

Para conocer la topología de la red, los nodos deben descubrirla, y esto sucede cuando los nodos anuncian su presencia y “escuchan” la presencia de los demás mediante transmisiones *broadcast*. Así, los sensores o nodos pueden saber qué nodos tiene alrededor y qué caminos o formas puede usar para alcanzarlos. Los algoritmos de enrutamiento deben cumplir unas normas para garantizar el buen desempeño de las redes inalámbricas de sensores:

- Tabla de enrutamiento lo más pequeña posible.
- Elegir la mejor ruta para un destino dado (evaluando la rapidez, confiabilidad, capacidad o menos costo)
- Actualización frecuente de la tabla de enrutamiento.
- Cantidad pequeña de mensajes y tiempo de convergencia

Existen diferentes criterios de los modelos de enrutamiento ya que la función de los mismos es rastrear la actividad de los objetivos en el entorno, creando así diferentes tipos de protocolos que permiten verificar transmisión de información a los nodos con las estaciones de base, algunos modelos de enrutamiento son:

1. **Modelo de un Salto:** Todos los nodos se conectan directamente a la estación base, esto puede ocurrir si la red es muy pequeña o si están al alcance del rango del radio del receptor de la base central de la red.
2. **Modelo de múltiples saltos:** Ocurre cuando todos los nodos envían su información a través de otros nodos vecinos hasta llegar al nodo más cercano que enviará la información a la estación base.
3. **Modelo Cluster:** Los nodos se agrupan y entre ellos escogen a un nodo vecino en específico que enviará la información a la estación base.
4. **Modelo Data-Centric:** Se puede aplicar cuando la red posee gran cantidad de sensores, para que no se pierda tanta cantidad de energía enviando todos los datos de la red, se solicita

solo un dato y los sensores negocian entre sí para decidir el dato más valido que se va a enviar para ahorrar energía.

5. **Modelo basado en localización:** Según la ubicación del nodo se decide qué camino va a tomar la información.

2.6 CONSIDERACIONES PARA LA EFICIENCIA ENERGÉTICA

La energía es un factor que se debe tener en cuenta en la tecnología inalámbrica porque generalmente es dependiente de una batería o acumulador de energía. Esto significa que existe energía finita, haciendo necesario considerar aspectos como:

- Consumo del CPU, radio transceptor, consumo de memoria externa y periféricos en algunos casos. Se debe asegurar que sean elementos de baja potencia.
- La optimización del consumo de energía de los nodos, ya sea con protocolos de enrutamiento o disminución de la longitud de los paquetes a transmitir.
- El uso de distintos modos de envíos ya sea de modo continuo, dirigido por eventos o condiciones o por consulta.
- El uso de sistemas híbridos combinando los protocolos de enrutamiento con técnicas de software como algoritmos genéticos y economizar las distancias de las comunicaciones.

Al tener en cuenta esto, se puede realizar esquemas de conservación de la energía en la red alargando su funcionamiento.

2.7 PROTOCOLOS DE ENRUTAMIENTO

Son de gran importancia y cada uno se puede enfocar en una variable o varias en específico, ya sea jerarquía de la red, administración de recursos, ancho de banda, energía o información; la elección correcta vendrá dada de acuerdo con la aplicación que la requiera y según la cantidad de nodos que posea la red.

Existen diversos protocolos y su uso depende exclusivamente de la aplicación y la topología de red que requiera el usuario, pueden ser clasificados como se indica en la figura 4.

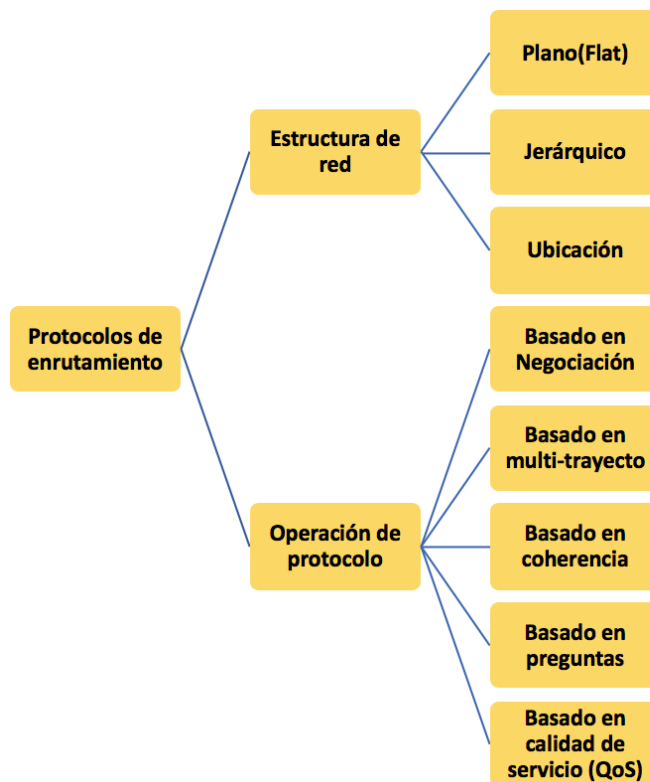


Figura 4. CLASIFICACIÓN DE PROTOCOLOS DE ENRUTAMIENTO

De acuerdo con la estructura de red se encuentran:

- Enrutamiento plano: Todos los nodos desempeñan el mismo rol en la red.
- Enrutamiento jerárquico: Existirán nodos que tendrán la capacidad de decidir y los otros de obedecer.
- Enrutamiento basado en localización: Las coordenadas de la ubicación de cada nodo y el consumo de energía dado por la distancia entre nodo, representa importancia en la red.

En la clasificación de acuerdo con la operación de protocolo se encuentran:

- Protocolos basados en negociación: Se intercambian mensajes para realizar acciones en la red ya sea el intercambio de información o el descarte de paquetes.
- Basado en múltiples trayectos: los nodos transmitirán la información, según las tablas de rutas guardadas en la memoria de los dispositivos.

- Basado en coherencia: Los nodos transmiten los paquetes de información basándose en la aplicación o en el tipo de paquete que va a enviar.
- Basado en preguntas: Para acceder a la información en un sitio específico de la red un nodo específico establece una petición y el nodo receptor decidirá si comparte los datos o no.
- Basado en calidad de servicio (QoS): Se establecen criterios de calidad ya sea de ancho de banda, baja latencia, ahorro energético entre otros, para gestionar la información.

Dentro de las clasificaciones mencionadas existe gran variedad de protocolos [10, 11], a continuación, se describen algunos de los más utilizados.

- **Difusión Directa:** Es un protocolo de negociación para redes que poseen diferentes variables, intereses, datos, etc. Cuando se establece una petición o un “mensaje de interés” en la red proveniente del usuario, que posee un mensaje que caracteriza el tipo de variable o de medición que se quiere de la red y esto genera múltiples caminos para obtener el mejor resultado a bajo consumo energético. [12]
- **PEGASIS (*Power- Efficient Gathering in Sensor information Systems*):** En este protocolo la comunicación de los nodos con una topología de cadena permite su transmisión en forma organizada y este a su vez trasmite la información al sumidero. El consumo de potencia es compartido de manera uniforme disminuyendo el ancho de banda para la comunicación, esto se debe a que trabajan de manera coordinada.
- **LEACH (*Low Energy Adaptive Clustering Hierachy*):** Este protocolo tiene la particularidad de escoger los nodos *cluster head* (nodos coordinadores) de manera aleatoria y rota esta función para distribuir la carga energética de manera uniforme a los nodos de la red. También es importante señalar que cada *cluster head* comprime los datos provenientes del grupo de nodos a los que pertenece y envía el paquete resultante a la estación base. La recolección de datos se debe realizar de manera constante y chequear de manera continua el grupo de sensores de la red.
- **Span:** Es un algoritmo basado en la posición, selecciona algunos nodos como coordinadores según su posición. Los coordinadores forman una red troncal utilizada para reenviar los mensajes. Un nodo se convierte en coordinador si dos vecinos (no

coordinadores) no pueden comunicarse entre sí o a través de uno de los coordinadores (acceso mediante tres saltos). Los coordinadores nuevos y los ya existentes no necesariamente deben ser vecinos. El diseño es poco eficiente en cuanto a energía ya que debe mantener la distancia de dos o tres saltos entre vecinos.

- **COUGAR:** Incorpora la arquitectura para el sistema de base de datos de los sensores, donde los nodos eligen un líder encargado de agregación y transmisión de datos a la estación base. La red se considera como una gran base de datos distribuida, realizándose consultas que determinan las funciones del procesamiento de la capa de la red como la selección de los sensores adecuados.
- **ACQUIRE (Active Query Forwarding in Sensor Networks):** La red es considerada como una base de datos donde las consultas complejas se pueden subdividir en varias subconsultas. La estación base envía una consulta que es retransmitida por cada nodo que la recibe; cada nodo intenta responder a la consulta parcialmente a partir de su información pre almacenada y de actualizaciones recibidas por sus vecinos en radio d saltos. Una vez resuelta la consulta completa, se envía de vuelta a través del camino más corto a la estación base. La escogencia del valor d puede afectar considerablemente el desempeño de la red.
- **VGA (Virtual Grid Architecture):** Se utiliza la agregación y procesamiento de datos dentro de la red para maximizar el tiempo de vida de la misma. Se asume una topología fija de los nodos. Trabajan con *clusters* cuadrados para obtener una topología virtual rectilínea, con *cluster* para cada zona. La agregación se realiza en dos niveles: local y global. Al conjunto *cluster* se le denomina *Local Aggregator* (LA) y se utilizan para realizar la agregación local mientras que el subconjunto de estos, llamados *Master Aggregator* (MA), realiza la agregación a nivel global y minimiza el costo de la ruta que llega a la estación base.
- **TTD (Two-Tier Data Dissemination):** Este protocolo provee la entrega de datos a varias estaciones base móviles. Cada fuente de información construye de manera proactiva una malla que se utiliza para difundir los datos a los sumideros móviles, asumiendo que los sensores son estacionarios y que su posición es conocida. Para retransmisión de datos usa algoritmos multi- salto.

- **SAR: (*Sequential Assignment Routing*):** Es uno de los protocolos de enrutamiento que introduce al Concepto QoS en las decisiones de enrutamiento.
- **SPIN (*Sensor Protocol for Information via Negotiation*):** Toda la información se distribuye a todos los nodos de la red, suponiendo que todos son posibles estaciones base. Así el usuario puede consultar cada nodo y obtener inmediatamente la información requerida. Se usa la negociación de meta- datos para evitar redundancia en la red.
- **CTP (*Collection Tree Protocol*):** La información es orientada a nodos padres o nodos raíces, y son escogidos por la red según la métrica ETX y la cercanía al nodo *sink* o sumidero. Los nodos que tengan un mayor valor de ETX deben transmitirles la información solo a los nodos con menor y mejor ETX en la red. Los paquetes con los datos recogidos por los sensores, se transmiten en forma de árbol desde los nodos hoja o hijos hasta los nodos padres o raíces hasta llegar al *sink*.

2.7.1 CONSUMO ENERGÉTICO EN PROTOCOLOS DE ENRUTAMIENTO PARA WSN

Las redes de sensores remotas inalámbricas funcionan generalmente con energía limitada, ya sean baterías AA, recargables, acumuladores, celdas solares, entre otros. Esto crea una necesidad de administrar recursos orientados al ahorro energético, para incrementar la vida útil de la red. Se han desarrollado investigaciones como por ejemplo la versión 4.1 de Bluetooth, que aumenta la duración del dispositivo en comparación con su versión Bluetooth 2.1 por indicar un ejemplo. En las redes inalámbricas de sensores los excesos en consumo energético dependen de dos cosas: el uso del radio (recepción y transmisión) y el procesamiento continuo de datos en el nodo.

Se pueden realizar modificaciones en la programación de protocolos en los “Nodos” que permitan reducir el gasto energético minimizando el funcionamiento de actividades que se excedan en el tiempo de funcionamiento.

En función de la vida útil de la red, un protocolo debería emplear las siguientes funciones:

- Uso alternado de nodos basándose en el consumo de energía de cada uno.

- Enrutamiento dinámico, basando las rutas más efectivas (comunicación exitosa) y con menos gasto energético.
- Apagado y encendido del radio cuando el nodo culmine la transmisión y recepción de paquetes.
- Transmisión a distintos niveles de potencia, adaptada a la distancia entre nodos.
- Recepción en modo *Sleep*, utilizando una señal que cambie el estado de la radio del nodo.

2.7.2 CTP

A continuación, se describe en detalle el *Collection Tree Protocol* (CTP), ya que su comprensión es de suma importancia para el desarrollo de este trabajo de grado.

El CTP es un protocolo de destino iniciado (los paquetes tienen como destino llegar al nodo *sink*), para esto se emplea difusión directa que significa que el *sink*, mandará un mensaje tipo *broadcast*; los nodos que lo reciban reenviarán este mensaje hasta que se creen las tablas de enrutamiento dentro de los nodos.

El enrutamiento en el protocolo depende de:

- El reenvío de los *beacons*.
- El algoritmo de goteo
- El valor de ETX.

Los mensajes *beacons* (mensaje de solicitud), establecen la localización de nodos adyacentes pidiéndoles información de sí mismos y sus vecinos. Para controlar el envío de estos *beacons* se emplea el algoritmo de goteo.

El valor de ETX (*Expected Transmission value* o valor de transmisión esperado), es un valor que se utiliza para estimar la calidad de un enlace, permitiéndole al nodo decidir a dónde enviar la información, la siguiente fórmula establece el cálculo de dicho parámetro y es explicado con mayor detalle en el siguiente capítulo.

$$ETX(1-salto) = \alpha ETX * Q + (1 - \alpha ETX) * ETX(actual)$$

El objetivo del protocolo CTP es proveer el mejor esfuerzo a cualquier costo en el datagrama de comunicación con los nodos raíces en la red, alargando la vida de la red gracias al ahorro de energía que se obtiene. Esto se consigue de la siguiente manera: cuando un sensor o nodo recolecta información de valor, los paquetes de datos se envían al nodo con el menor valor de ETX y así sucesivamente hasta llegar al *sink*. De esta manera no necesariamente el camino siempre será el mismo, debido a que al pasar el tiempo no todos los nodos seguirán funcionando y los caminos cambiarán gracias a este gradiente de enrutamiento (ETX), siendo siempre posible una comunicación entre los nodos restantes hasta el fin de la red.

La estructura del protocolo viene dada por:

- Tipos de paquetes, utiliza 3 tipos de paquetes distintos: los de confirmación, los de enrutamiento y los que transportan la información. Los paquetes de confirmación son mensajes de respuesta que certifican una transmisión de paquetes exitosa. Los paquetes de enrutamiento se llaman *beacons*, son capaces de modificar la topología tipo árbol de la red y los gestiona el motor de enrutamiento (*routing engine*) junto al estimador del enlace (*link estimator*). Los paquetes de datos son aquellos que transportan la información recogida por los sensores y a su vez son enviados al *sink*, los gestiona el motor de envío.
- Motor de enrutamiento: Se encarga de enviar y recibir los *beacons* y se gestiona a través del algoritmo de goteo. Mantiene la tabla de rutas actualizada y se encarga de escoger los nodos raíces en la red, recordando que en una topología de árbol pueden existir varios nodos raíces que enrutarán los paquetes al *sink*.
- Motor de Envío: Gestiona los paquetes de datos, se encarga de descartar y detectar los paquetes duplicados y los bucles, usando los paquetes ACK, que son paquetes de confirmación de recepción de los mensajes; si estos no llegan, se debe volver a enviar los paquetes, generados por el mismo nodo, como los paquetes de datos de otros nodos dentro de la red. Este módulo también se encarga de detectar y eliminar paquetes duplicados y bucles. Para hacer el envío de datos se tiene una cola de tipo FIFO de 13 posiciones, donde el nodo espera un ACK antes de remover el paquete enviado de la cola, si no llega el ACK se procede a hacer la retransmisión. El número máximo de retransmisiones para un paquete es de 30. En el caso de que el número de retransmisiones sea permanentemente grande, la

cola se va a ir llenando. Cuando la cola tiene más de la mitad de paquetes en espera se dice que el nodo está congestionado. Para detectar paquetes duplicados el nodo evalúa el *SeqNo* y el origen del paquete y lo compara con una memoria inmediata. Para detectar un bucle el nodo evalúa el ETX de sus hijos, si el ETX es menor quiere decir que existe un bucle dentro de la red. En este caso se activa un temporizador(*timerBackOff*) y se pide una actualización de topología.

- Estimador de enlace: Se encarga de determinar la calidad del enlace de entrada y salida de cada salto en la comunicación de los nodos, representándose con una métrica ETX. El estimador de enlace realiza una estadística sobre el número de *beacons* recibidos y el número de paquetes enviados exitosamente, con lo que calcula una métrica de salida utilizando una relación de envío total de *beacons* del nodo vecino y el total de *beacons* recibidos. Para el salto salida se usa la misma estimación, utilizando la relación de los *beacons* enviados al nodo vecino y los recibidos por el mismo para entregar con éxito el paquete de datos al nodo vecino.

Al recibir un *beacon* en un nodo, el estimador de enlace extrae la información del encabezado e incluye una tabla de estimación de enlace donde se incluye la información de los nodos vecinos e información relacionada a ellos como el valor de ETX de salto o la cantidad de tiempo que pasó desde la última actualización de ETX de los nodos vecinos.

El estimador de enlace genera una tabla de enrutamiento propia, donde guardará los mejores posibles enlaces que no se encuentren en la tabla del motor de enrutamiento y comparará los valores de ETX de los nodos que estén transmitiendo en ese instante con los valores ya guardados, forzando la actualización de la tabla del mismo como la del motor de enrutamiento.

La figura 5 muestra un diagrama de flujo del funcionamiento general del CTP.

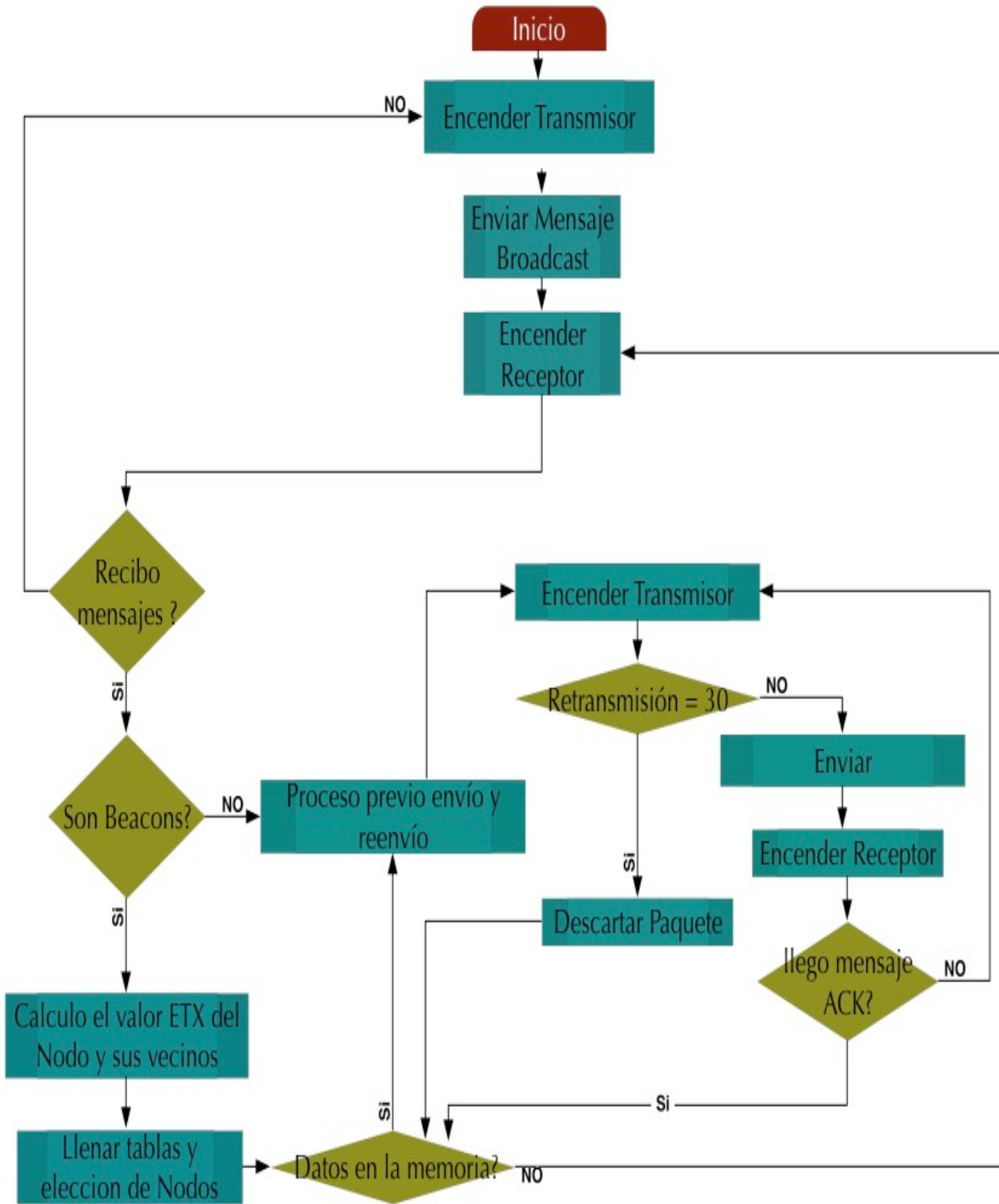


Figura 5 DIAGRAMA DE FLUJO - FUNCIONAMIENTO CTP

CAPITULO III

3 *COLLECTION TREE PROTOCOL* – “PROCOLO DE RECOLECCIÓN EN ÁRBOL” (CTP) IMPLEMENTADO EN EL SIMULADOR CASTALIA 3.3

En el capítulo anterior se explicó el funcionamiento teórico del CTP, ahora es necesario para entender la investigación y el alcance del trabajo, explicar su implementación, que fue programada inicialmente en el simulador modular Castalia 3.3 por U. Colesanti y S. Santini [2], que a su vez se basaron en la propuesta para TinyOS 2.1 de R. Fonseca y otros [1, 7].

TinyOS 2.1 es un sistema operativo de código abierto que se utiliza dentro del microcontrolador del nodo, en este va programado el protocolo que va a usar. Es de importancia recalcar que la implementación en Castalia busca simular el comportamiento del protocolo programado en el sistema operativo.

Esta implementación se realizó como un módulo compuesto (*CtpNoe*), que se compone a su vez de 5 módulos (*Ctp*, *DualBuffer*, *LinkEstimator*, *CtpRoutingEngine*, *CtpForwardingEngine*).

El primer módulo *CTP* intercambia los mensajes e información entre los módulos de aplicación y el módulo MAC (capa de acceso medio), los tres últimos módulos mencionados se encargan de la simulación del enrutamiento de los datos, en la figura 6 se puede visualizar la interacción entre módulos.

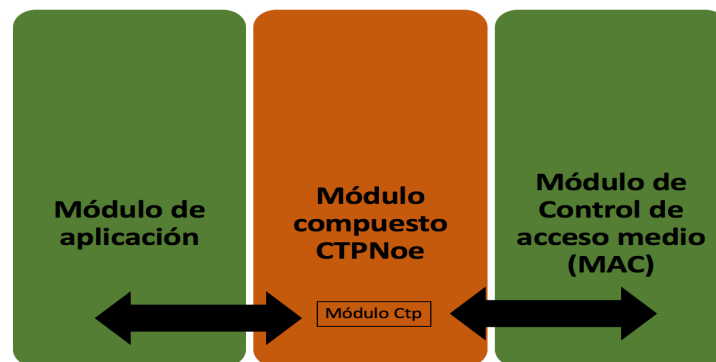


Figura 6. COMUNICACIÓN DEL MÓDULO COMPUESTO CTPNOE CON MÓDULOS EXTERNOS

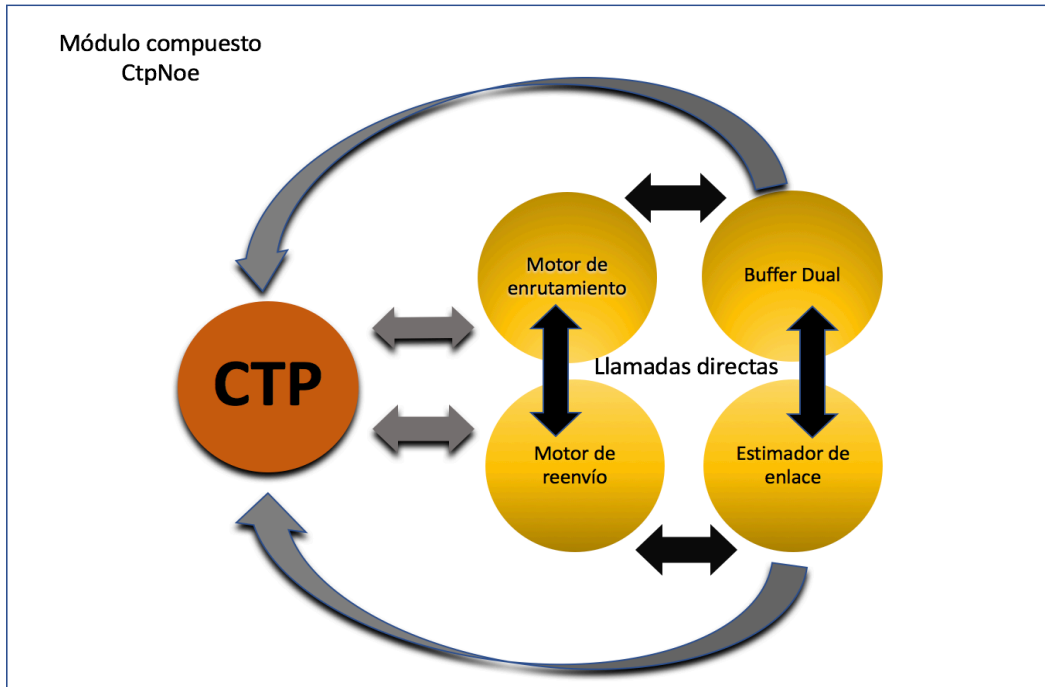


Figura 7. COMUNICACIÓN DE LOS MÓDULOS INTERNOS EN EL MÓDULO CTPNOE [2]

El módulo *CtpNoe* se comunica por medio de mensajes como se aprecia en la figura 7; estos mensajes son diferentes y están programados para funciones distintas, tanto la simulación de enrutamiento como la comunicación entre módulos.

3.1 PAQUETES DE ENRUTAMIENTO Y DE DATOS DEL PROTOCOLO CTP IMPLEMENTADO EN CASTALIA 3.3

El protocolo se basa en el intercambio de mensajes a través de un mantenimiento de una estructura tipo árbol y de reportar que la información de la aplicación de la red (*payload*) se transmitió al nodo *sink* o nodo raíz. En la implementación realizada por R. Fonseca y otros [1, 7] se realizó una estructura para los paquetes de datos y de ruta que se pueden visualizar en la figura 8; esta estructura se mantiene en la implementación en estudio.

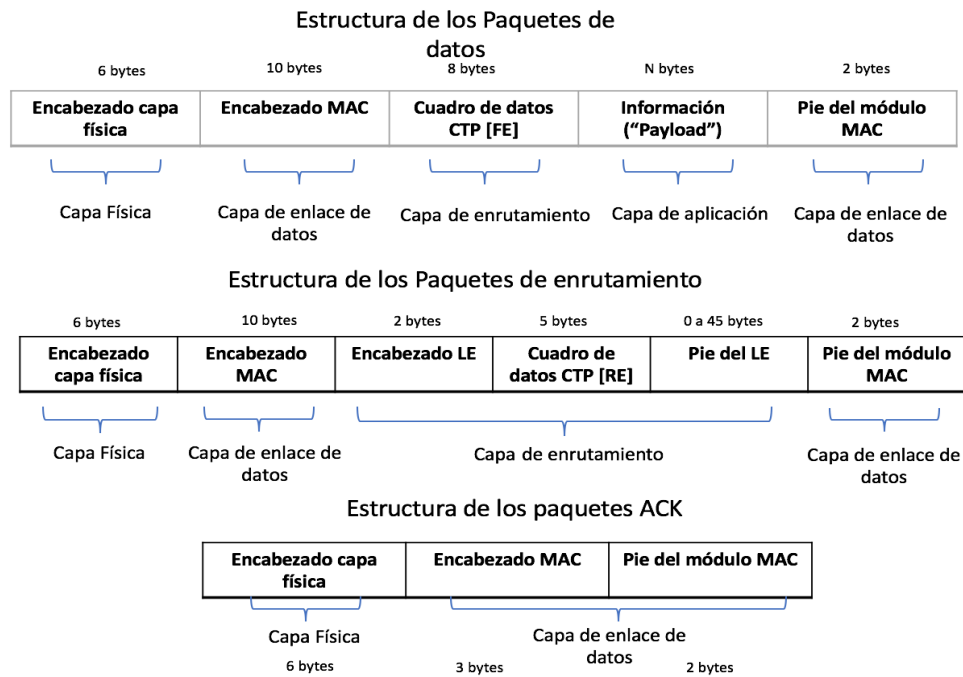


Figura 8. ESTRUCTURA DE LOS PAQUETES DE ENRUTAMIENTO, DATOS Y RECONOCIMIENTO (ACK) EN CTP [2]

Encabezado PHY (capa física) y MAC (capa de acceso control medio)

En la figura 8 se muestra que tanto el paquete de enrutamiento como el de datos tienen un total de 18 bytes de información añadidos por la capa física y la capa de acceso control medio (MAC); ese encabezado es el que acompaña a los datos cuando se envían mediante transmisor-receptor inalámbrico. En Castalia los módulos MAC se encargan de añadir los encabezados, así como su peso en bytes, en TinyOS2.2 también el controlador del radio realiza esa función.

Marco de datos en CTP.

Antes de transmitirse los paquetes a través de la radio, estos son administrados por el motor de reenvíos que programa la transmisión en la capa de enrutamiento. A su vez este mismo le añade 8 bytes de control a los paquetes de datos en tránsito como se muestra en la figura 9. Los primeros 2 bits del primer byte incluyen las banderas: P (*Pull* = empujar) y C (*Congestion* = Congestionado); la primera se usa para permitir el envío de *beacons* en busca de actualización de la topología de la red y la segunda envía la señal de congestión, por ejemplo si un nodo envía la bandera “C” a un

nodo vecino, este detendrá el envío de paquetes a este nodo hasta que se descongestione y buscará otras rutas. Los otros 6 bits están reservados para agregar más banderas con otras funciones en el futuro. El segundo byte reporta un valor THL (THL= “*Time has lived*”), es un contador que refleja cuántos paquetes ha entregado exitosamente un nodo. El tercer y cuarto byte están dedicados para multi saltos y métrica ETX, mientras que el quinto y el sexto se encargan del nodo de origen; al final del marco de encuentran los bytes para el identificador de colección, CollectId que especifica el tipo información se está recogiendo. El CTP puede manejar múltiples aplicaciones por lo tanto cada aplicación va a tener su identificador de “CollectId”, que diferenciará una aplicación de otra. En la figura 8 se visualiza que la carga o la información a transmitir no tiene un peso definido y eso se debe a que es variable según la aplicación de la red.

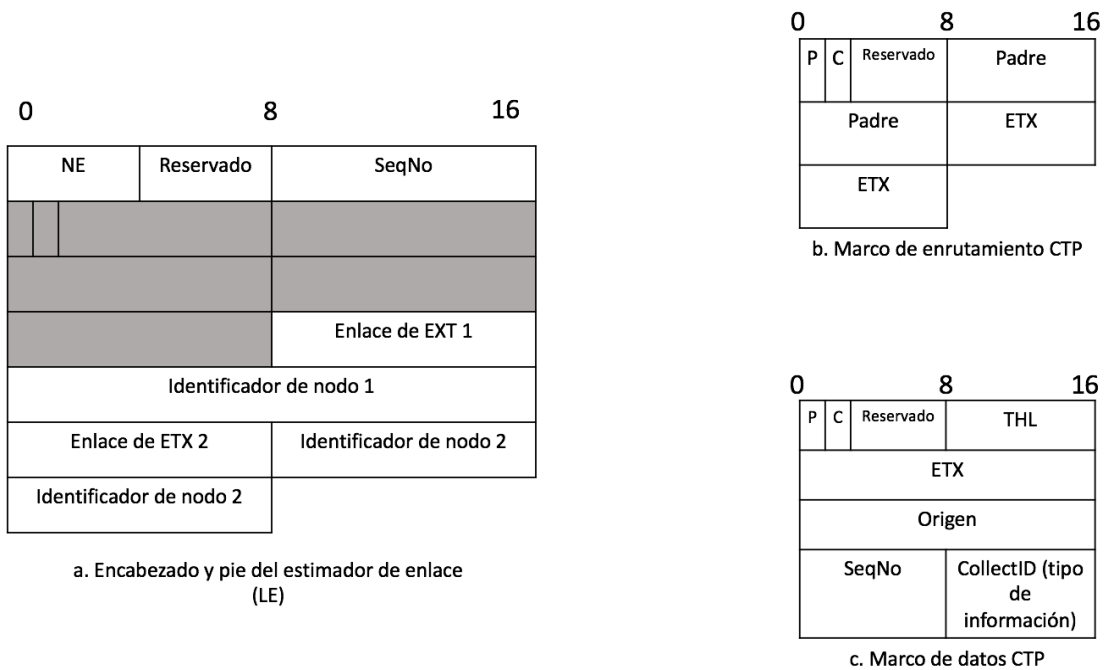


Figura 9. ENCABEZADOS, MARCO DE DATOS Y MARCO DE ENRUTAMIENTO DEL ESTIMADOR DE ENLACE [2]

Marco de enrutamiento en CTP.

Como se mencionaba anteriormente, la información es compartida con los vecinos a través del envío de paquetes sobre un enrutamiento con topología tipo árbol, por lo tanto, los paquetes serán enviados con el fin de mantener esta estructura.

Los datos son manejados por el motor de reenvíos, los paquetes de ruta son generados y procesados por el motor de enrutamiento y el estimador de enlace. En la figura 8, se muestra que los paquetes de enrutamiento cargan adicionalmente, 2 bytes de encabezado y un pie variable que son establecidos por el estimador de enlaces (LE) que se describirán más adelante. El actual marco de enrutamiento posee 5 bytes de largo y está estructurado como se indica en la figura 9a. En el marco de enrutamiento se aprecia que el primer byte incluye dos banderas la P, la C y 6 bits que no se usan al igual que el marco de datos. El segundo y tercer byte contienen el campo de padres que especifica un identificador de padre del nodo mandando un mensaje *beacon*. El cuarto y el quinto byte incluyen la métrica ETX (para múltiples saltos), a pesar de ocupar dos bytes, para realizar un solo salto ETX solo se requiere 1 byte.

ACK (paquetes de reconocimiento)

En la figura 8, se puede observar la estructura de los paquetes de reconocimiento, son las notificaciones de paquetes de datos entregados exitosamente al nodo que lo envió. El CTP usa la capa de enlace para los paquetes de reconocimiento, estos solo incluyen información de la capa física y la capa MAC generando un total de 11 bytes.

3.2 MÓDULO DE ESTIMADOR DE ENLACE

Como se establece en la propuesta de R. Fonseca y otros [1], el estimador de enlace es responsable de determinar la calidad del enlace y está implementado en sus correspondientes archivos con el nombre de *LinkEstimator*, como se establece en la documentación de la implementación [2] y con el fin de evitar repetir el nombre se le reconoce por las siglas “LE”.

Encabezado y pie del estimador de enlace

Estos encabezados se añaden antes la radio transmisión del paquete, el primer byte que se aprecia en la figura 9a. Se le asigna el valor NE (*Number of Entries*) que contendrá el número de entradas y encierra la longitud que tendrá el pie o el final del paquete. El valor está codificado en

4bits y el resto (4bits), están reservados para otros usos. El segundo byte almacena el valor del SeqNo, que se incrementa en valores de a 1 con cada transmisión de un mensaje *beacon*. Se compara con el SeqNo de los nodos vecinos, con la finalidad de estimar una relación del número de *beacons* perdidos sobre el número total de *beacons* enviados por el nodo vecino específico.

En esta implementación no se programó el pie del paquete que establece el estimador de enlace, debido a consideraciones de uso de los autores [2].

Calculo de 1 salto EXT

Para cada nodo vecino el estimador de enlace determina el salto de ETX según la calidad del enlace de entrada y de salida. La calidad del enlace de salida se calcula de esta manera:

- nu = número de paquetes enviados por un nodo a un nodo en específico (paquetes “Unicast”), incluyendo la retransmisión.
- $n\alpha$ = números de paquetes ACK recibidos.
- La calidad del enlace viene dada por la relación del número de paquetes enviados entre el número de paquetes ACK recibidos:

$$Qu = nu / n\alpha$$

Si $n\alpha = 0$, la calidad del enlace se establecerá como el número de fallos en envío desde el último mensaje entregado, por lo tanto, el estimador de enlace pondrá como valor el Qu después de un número wu (valor que viene preestablecido, se encuentra en la tabla 1) que corresponde al número de paquetes “unicast” que se han enviado. Se resetean los valores “n” para calcular otro valor de Qu . Esto se realiza para cada nodo en específico con el que se desea intercambiar los paquetes de datos.

Para que el estimador de enlace realice esta actividad, debe extraer esta información de la capa de enlace donde se encuentra la información guardada en el paquete ACK.

En la implementación, el modulo MAC envía la información al módulo de Motor de reenvío (Nombre del archivo = CtpForwardingEngine), inmediatamente después de la recepción del paquete ACK.

Es de importancia acotar que el motor de reenvío se comunica con el estimador de enlace enviándole los paquetes de recepción.

El valor Q_u se puede establecer como la calidad que tiene un enlace de salida (transmisión) con otro nodo y en toda comunicación se establece un enlace de salida y de llegada (recepción), por lo tanto, también se calcula un valor de calidad para el enlace de llegada:

$$Q_b = n_b / N_b$$

donde:

- Q_b = valor de calidad del enlace de entrada (recepción).
- n_b = número de *beacons* recibidos por un nodo vecino específico.
- N_b = número de *beacons* enviados por el mismo vecino.

Esto indica que cada vez que se recibe un *beacon* se calcula un valor de Q_b (Se calcula dentro de una ventana con valor w_b , viene preestablecido, se encuentra en la tabla 1). Por otro lado, antes de enviar la información del valor de Q_b , esta pasa por un filtro exponencial descrito por la siguiente función matemática:

$$Q_b[k] = \alpha_b * (n_b / N_b) + (1 - \alpha_b) * [k + 1]$$

Esto se especifica en los archivos de implementación del estimador de enlace de R. Fonseca y otros [1] en el archivo `file/tos/lib/4bitile/LinkEstimatorP.nc`, ahí se especifica el valor de cada variable. Esas mismas variables y fórmulas son usadas en la implementación en el simulador Castalia. Con estos valores se obtiene el valor de un salto ETX y el múltiple salto ETX y se usan para la búsqueda de padres donde se escogen aquellos con el menor valor de ETX

Cada vez que se actualiza el valor de Q_u y Q_b para un nodo vecino disponible, la métrica o medida de 1 salto EXT respecto a ese vecino se actualiza. Para esta actualización, se aplica la siguiente fórmula en el código implementado en el estimador de enlace:

$$ETX(1-salto) = \alpha_{ETX} * Q + (1 - \alpha_{ETX}) * ETX(actual)$$

Después de cada actualización el valor del salto ETX se almacena en la tabla del estimador de enlace con su correspondiente identificador de nodo vecino.

Estimador de enlace		
Parámetro	Valor	Unidad
α_{ETX}	0.9	-
α_b	0.9	-
wb	3	Paquetes
wu	5	Paquetes
Capacidad en la tabla del estimador de enlace	10	Entradas
Tamaño del encabezado	2	bytes
Tamaño del pie	< 45	bytes

Tabla 1. VALORES DE LOS PARÁMETROS IMPLEMENTADOS EN EL MÓDULO DEL ESTIMADOR DE ENLACE

Procedimiento de inclusión y exclusión de nodos en la tabla del módulo de estimador de enlace.

Cuando un nodo recibe un mensaje *beacon* de un nodo vecino, el estimador de enlace chequea si el nodo está incluido en la tabla de estimación de enlace o no; si hay espacio disponible en la tabla, se inserta la dirección en la entrada disponible. Esta tabla posee un espacio de almacenamiento hasta 10 entradas. Si está llena, pero tiene una entrada que ya no es válida, se sustituye la entrada no válida por la nueva (los nodos poseen un bit de validación que posee un tiempo de expiración).

En el caso que la tabla esté llena y todas sus entradas sean válidas, el módulo verifica el valor de 1 salto EXT y compara si la nueva entrada posee un valor más grande que alguna de las existentes. Para que este valor sea calculado los valores Q_u y Q_b tienen que estar calculados.

En el caso de estar llena la tabla y las entradas son iguales o mejores, el estimador de enlace decide si descartarlo o incluirlo. Para que se incluya una entrada nueva a pesar de lo mencionado anteriormente se tienen 2 casos:

- El mensaje *beacon* pertenece a un nodo raíz y el valor de su múltiple-salto EXT tiene valor 0. (0 es el valor ETX del SINK)

- El valor del múltiple- salto ETX es bajo comparado con el nodo que recibió el *beacon*, lo que lo convierte en un nodo raíz y tiene prioridad en la tabla del estimador de enlace.

Después de este punto el estimador de enlace envía una solicitud de comparación al motor de enrutamiento y este enviará un mensaje llamado bit de comparación; si posee valor de 1 el nodo se inserta en la tabla, pero si el valor es 0 se descarta. Sin embargo, puede ocurrir que el estimador de enlace determine que ese nodo debería estar incluido en la tabla, así que escoge a un nodo que no esté fijado en la tabla para cambiarlo por el nuevo.

3.3 MÓDULO DEL MOTOR DE ENRUTAMIENTO

Las funciones de este módulo son:

- Enviar los mensajes *beacons*.
- Llenar la tabla de rutas (distinta a la tabla de rutas óptimas del estimador de enlace).
- Mantener actualizada la tabla de rutas.
- Seleccionar el nodo padre o raíz conservando la arquitectura de árbol.

Es importante mencionar la relación que mantiene este módulo con el estimador de enlace, debido a que los marcos de ruta que genera son transferidos al estimador de enlace para agregarle el encabezado y el pie del paquete antes de ser enviado a la capa MAC.

Frecuencia de envío de mensajes *beacon*

Esta frecuencia se controla por medio del “algoritmo de goteo”, este algoritmo escoge intervalos aleatorios de valores en tiempo $[I/2, I]$, el valor de A se duplica con cada transmisión, por lo tanto, la frecuencia donde los *beacons* se reduce. Estos valores poseen valores mínimos y máximos que permitirán la sincronización entre todos sin afectar el funcionamiento (tabla 2).

Estos valores son reseteados en caso de eventos importantes como la creación de un bucle o la detección que un paquete posee el bit Pull(p), con un valor de 1, lo que establece que debe resetear el intervalo de envío *beacons*.

Motor de enrutamiento		
Parámetros	Valores	Unidades
I min	125	ms
I max	500	s
Capacidad de tabla de rutas	10	Entrada
Umbral de cambio de nodo padre	15	--
Periodo de actualización de los nodos padre	8	s
Tamaño de msj Beacon	5	Bytes

Tabla 2. VALORES DE LOS PARÁMETROS RELEVANTES DEL MÓDULO DEL MOTOR DE ENRUTAMIENTO

Actualización de tabla de rutas

La tabla de rutas no solo almacena el nodo por donde va a transmitirse la información, también posee información sobre la congestión, status y múltiples saltos EXT de los nodos vecinos por donde se está estableciendo el intercambio de información.

Posee también un máximo de 10 entradas al igual que la tabla del estimador de enlace, la diferencia es que aquí se encuentran las mejores rutas del momento o con mayor probabilidad de factibilidad.

Selección de los nodos padres o raíz

Se realiza periódicamente o de manera asincrónica cuando los siguientes eventos ocurren:

- Un mensaje *beacon* es enviado.
- Un nodo es inalcanzable o ya no es válido.
- Un nodo vecino ya no está congestionado.
- El nodo vecino está en ese momento congestionado.
- El nodo vecino no posee ruta al nodo *sink*.
- Los nodos siempre son escogidos por su valor de ETX, que siempre tiene que ser menor al del nodo en cuestión.

Es necesario un umbral de ETX para evitar que se seleccione un nodo padre que realmente sea un nodo hijo; teóricamente en diseño el umbral es 1, pero en la implementación este número es mucho mayor (15) para que la diferencia tan grande cree una estricta condición de no elegir un nodo hijo a donde enviar la información. A pesar de todo esto siempre es posible que suceda un evento de bucle, sin embargo, existe un procedimiento para eliminarlos.

3.4 MÓDULO DEL MOTOR DE REENVÍO

Se encarga de recibir todos los paquetes referentes a la información en el sistema, también detecta los eventos y avisa a los módulos que les corresponde tomar la acción, estos eventos pueden ser, congestión, bucles, paquetes duplicados. Puede identificar los paquetes que no son para el nodo y usarlo para actualizar la topología de la red. En esta implementación los archivos llevan el nombre de “CtpForwardingEngine”.

Paquetes recolectados y retransmisiones

En la implementación de este módulo se pueden recolectar hasta 13 paquetes y almacenarlos hasta retransmitirlos todos, la retransmisión estará completa con la llegada del mensaje ACK que indicará que el paquete llegó a su destino, pudiendo ser sustituido por nueva información u otro paquete de otro nodo vecino. Si el nodo vecino está congestionado, este módulo al recibir el paquete se lo comunica al módulo de enrutamiento para que junto al estimador de enlace asignen rutas temporales hasta que el nodo se descongestione. Si el mensaje ACK no es recibido, este módulo se encargará de retransmitir el paquete hasta un máximo de 30 veces, después en el caso de la no recepción del paquete ACK, el paquete es descartado.

Bandera de congestión

La bandera de congestión se activa (bit C ubicado en la figura 9c) cuando varios nodos escogieron al mismo nodo padre y gran cantidad de paquetes en un nodo se empiezan a perder o descartar. Por lo tanto, cuando esto se activa, el motor de reenvío obliga a buscar otras rutas con nodos menos congestionados o preferiblemente sin congestión, permitiendo una degradación

equitativa de la energía en todos los nodos y un reparto de paquetes equitativo en la red para garantizar que lleguen a su destino y a su vez el gasto energético en procesamiento es similar.

Este reparto equitativo de la información es uno de los puntos a favor de esta aplicación, porque se garantiza siempre una optimización en el envío, obligando que los datos lleguen a su destino a pesar de las dificultades.

Mientras el bit de congestión esté en nivel 1, se le enviarán mensajes al motor de enrutamiento para que actualice la tabla y envíe los paquetes a otro nodo vecino.

Paquetes duplicados

Para detectar los paquetes duplicados el motor de reenvío revisa los valores de origen, CollectId, SeqNo, THL, que se encuentran en los encabezados y pie de los paquetes. Por otro lado, el motor de reenvío guarda los últimos 4 paquetes enviados exitosamente.

El campo de origen permite saber qué nodo inicial está enviando el paquete, CollectId identifica la aplicación del paquete o el uso de esa información que se está recolectando, el valor de SeqNo permite saber cuántas veces se ha retransmitido o la secuencia del marco de ese nodo, el valor THL determina cuántos saltos ha dado el nodo.

Bucles en ruta

Si el módulo detecta un valor de ETX que determina que un nodo padre o de menor valor ETX está enviando información a un nodo hijo, inmediatamente se envía un paquete al motor de enrutamiento para que realice una actualización de la tabla de rutas y reinicie el proceso de envío de *beacons*, de este modo se pueden calcular nuevas rutas para los paquetes.

Mecanismo de fisgoneo

Consiste en recibir los paquetes dirigidos a otros nodos, y mediante los bits bandera (*flags*) C y P ubicados en el marco de datos, actualizar la topología de la red de manera más conveniente para que la información del nodo en cuestión llegue por el camino menos congestionado.

Se considera que la información sobre el paquete está disponible en la capa de enlace, esta función esta detallada en el módulo MAC del cual se hablará más adelante.

Temporizadores de tiempo de espera

Estos llevan los nombres de: TX_OK, TX_NOACK, CONGESTION y LOOP, el primero se activa después de recibir un paquete de transmisión exitosa (ACK), el segundo se activa solo si no recibe el paquete ACK correspondiente a la transmisión. El tiempo de congestión se activa después de recibir un paquete de un nodo con el bit de congestión en nivel lógico 1. Cuando este temporizador expira, se vuelve a verificar el status del nodo. Y finalmente el temporizador LOOP, se activa después de recibir un paquete de un nodo con menor EXT que el actual. En la tabla 3 se encuentran los parámetros usados en la implementación referente a los temporizadores.

Motor de reenvío		
parámetros	Valor	Unidades
Tamaño de recolección de reenvíos	13	Paquetes
Tamaño de LRU Cache	4	Paquetes
Max intentos de retransmisión	30	--
TX_OK	15.6 – 30.3	ms
TX_NOACK	15.6 – 30.3	ms
CONGESTION	15.6 – 30.3	ms
LOOP	65.5-124	ms
Tamaño de encabezado	8	bytes

Tabla 3. VALORES RELEVANTES DE LOS PARÁMETROS DEL MÓDULO DEL MOTOR DE REENVÍOS

3.5 MÓDULO DE BUFFER DUAL

Este módulo realiza actividades de almacenamiento y administración de los paquetes que vienen del estimador de enlace y del motor de reenvío y administra los paquetes de salida de manera separada en cada buffer.

3.6 IMPLICACIONES DEL MÓDULO MAC DEL PROTOCOLO DE RECOLECCIÓN EN ÁRBOL REALIZADO PARA EL SIMULADOR CASTALIA

La creación de este módulo fue necesaria debido a la interacción particular existente entre la capa de enlace con la capa MAC diseñada en el protocolo [1], a su vez también debe a que la implementación original fue diseñada para TinyOS 2.1 y se tuvo que emular el comportamiento del divisor de fase que utiliza este en el simulador. Las razones para diseñar este módulo en vez de usar las programadas por defecto en Castalia son:

- Capa de enlace con la adaptación de los ACK del CTP.
- Propagación de la información en la capa del enlace.
- Recepción de paquetes cuyo destino no es el nodo que los está recibiendo.
- Replicar el mecanismo de división de fase del simulador TinyOS.

3.7 MÓDULO CC2420MAC

El modulo MAC programado en Castalia se llama CC2420Mac, esto se debe a que se implementó el modelo de radio CC2420 a causa de ser uno de los más usados en el mercado por los dispositivos TelosB y MicaZ, la hoja de datos del Chipcon 2420 radio transmisor-receptor (*transceiver*) comenta la capacidad de encendido y apagado del radio [8].

La programación de su funcionamiento comienza con un temporizador llamado “initialBackoff”, el valor se escoge de manera aleatoria en un intervalo pre-programado en la tabla 4. Al terminar el tiempo, el módulo verifica que el canal esté libre para dar tiempo al segundo temporizador llamado “ack turnaround timer”, que espera que llegue el paquete de confirmación de paquete entregado (ACK). Si esto se cumple el paquete siguiente es puesto en la agenda de envío. Si el canal está ocupado se pone en la agenda de envío en un tiempo aleatorio después que culmine el tiempo de congestión llamado “congestionBackoff”.

Este módulo tiene la peculiaridad de usar los mensajes de comunicación virtual de la MAC predefinidos en el Castalia, por lo que usa los formatos de mensajes estándares (MacPacket, MacControlMessage), establecidos en el manual del simulador para la creación de módulos en el

mismo y de comunicarse mediante mensajes con el módulo de enlace de datos y el motor de reenvío como se muestra en la figura 10, ubicada al final del capítulo.

En la tabla 4 se establecen los intervalos de tiempo de espera, programados en la implementación, referidos en la hoja de datos de radio usado [8]. Por ejemplo, el máximo tiempo de espera antes de declarar que una comunicación fue fallida (variable en código = CC2420_ACK_WAIT_DELAY), el tiempo de espera mínimo o el periodo de tiempo de espera (CC2420_MIN_BACKOFF y CC2420_BACKOFF_PERIOD), estos tiempos se expresan en múltiplos de 35kHz pulsos de reloj, en la tabla se reflejan estos valores en milisegundos.

MAC		
Parámetros	Valor	Unidades
Rango InitialBackoff	0.3 -10	ms
Rango del Congestion Backoff	0.3- 2.4	ms
Encabezados PHY (capa física)	6	bytes
Encabezado MAC	12	bytes
Tamaño del paquete ACK	11	bytes
Tiempo máximo para ACK	7.8	ms
Tiempo de ACK	21	μs

Tabla 4. VALORES RELEVANTES DE LOS PARÁMETROS DEL MÓDULO MAC

ACK en la capa de enlace

Los paquetes ACK se transmiten a todos los nodos (mensaje *broadcast*) de manera que a través de ellos llegue al nodo que originó la información; es un paquete con pocos bytes de información y es la capa de enlace la que se encarga de recibirlos, pero la capa MAC se encarga de notificar si este mensaje no llegó. En caso de llegar este mensaje la transmisión concluye y se procede a calcular el valor de 1 salto de ETX y el número de paquetes ACK recibidos exitosamente.

Propagación de la información en la capa de enlace

En la implementación se usó el enrutamiento por defecto descrito en el manual de Castalia [3], esto se describe en el archivo “RoutingPacket.msg”. En la carpeta “Routing” de Castalia [2] funciona similar a los archivos LQI o RSSI (donde el nodo a comunicar se establece en el encabezado de pie del paquete recibido cuando pasa por la capa física), la información es propagada hacia las capas superiores y los campos son copiados de una estructura a otra y son incluidas por defecto en el “MacInterationControl” (interacción de control MAC) de la estructura de los “MacPacket” (paquetes de la capa MAC) y el “RoutingInteractionControl” (control de interacción de rutas) de los “RoutingPackets” (paquetes de ruta).

Fisgoneo de paquetes

A pesar de lo mencionado sobre el fisgoneo de paquetes, cuando la red está congestionada el CTP es capaz de descartar los paquetes que no son inicialmente para ese nodo, por ejemplo, en el caso de un nodo padre por el cual circule información que se dirija al *sink* directamente, si se congestiona, la capa de enlace del nodo al recibir los paquetes que no son enviados para él leyendo en el encabezado MAC donde se encuentra la dirección de destino, los descarta inmediatamente.

Mecanismo de división de fase de TinyOS

La manera en que el modulo MAC emula esta función del TinyOS, es con el envío de mensajes. Se define un mensaje llamado CC2420ControlMessage que se extiende hasta el MacControlMessage ubicado en la comunicación virtual de las MAC en el Castalia y de esta manera se simula el mecanismo de señalización del otro sistema, se simula la señal Send.sendDone(message_t,error_t) y esta señal carga con toda la información de la interfaz TinyOS, así como información útil como los resultados y mensajes ACK.

Encabezado MAC

El encabezado se deriva del formato que mantiene el marco del estándar IEEE802.15.4 que es el mismo que usa el radio implementado en el simulador.

La capa física añade 3 informaciones que ocupan un espacio máximo de 6 bytes, 4 para la secuencia, 1 para el delimitador y 1 para el largo del marco, a su vez la capa Mac añade 5 campos, 2 donde uno contiene la información o “*payload*” y el otro con una distancia variable. Se diseñó el encabezado MAC del CTP con un tamaño de 12 bytes donde 10 bytes son asignados a la cabeza del paquete y 2 al pie como en la figura 8. También se muestran los paquetes ACK que ocupan 11 bytes donde 5 representan el encabezado de la capa MAC y los otros 5 pertenecen a la capa física. Es importante mencionar que estos paquetes los envía el controlador de la radio (CC2420).

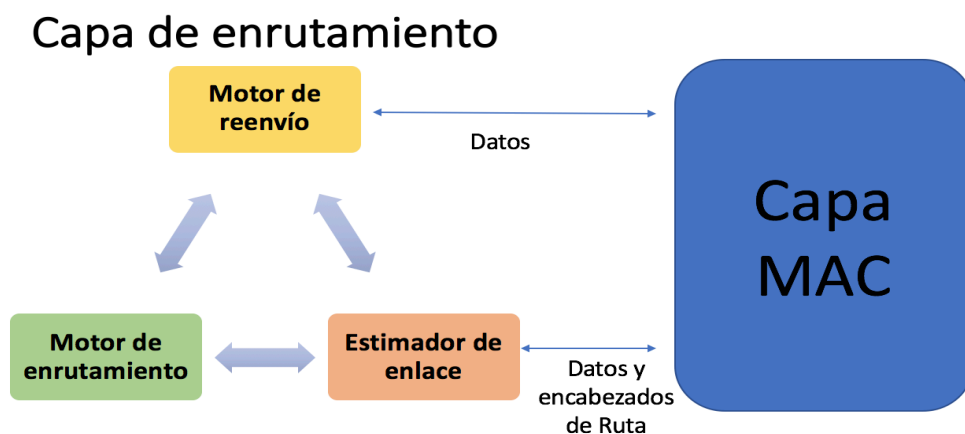


Figura 10. COMUNICACIÓN CTP CON LA CAPA MAC [2]

En la figura 10 se describe la interacción entre la capa de enrutamiento y la MAC. Esta interacción se podría decir que es la base del funcionamiento del CTP, sin embargo, se determinó que se excluyeron ciertas funciones en la programación y usos de la implementación original [1] que no se encuentran comentadas y realizadas en el trabajo realizado por U. Colesanti y S. Santini [2], y son de importancia mencionar:

1. La implementación original establece un modo de intervalos de 240 segundos que permitiría que una red de sensores opere a bajo consumo por 400 días [1].
2. Los módulos MAC y CTPNoe de esta implementación no cumplen los criterios de comunicación y programación entre módulos establecidos en el manual de usuario del

simulador [3]. Esto no permite modificaciones pertinentes a la investigación sino modificaciones directas a la programación interna del protocolo.

3. Los temporizadores de envío de mensajes poseen deficiencias e inconsistencias de programación, esto genera congestión en las redes simuladas y dificulta el análisis para escenarios de simulación con mayor cantidad de nodos.

CAPITULO IV

4 PROPUESTA DE ESQUEMA DE APAGADO Y ENCENDIDO DE NODOS PARA EL PROTOCOLO CTP PROGRAMADO EN EL SIMULADOR CASTALIA 3.3

Para realizar la mejora del protocolo y evitar problemas de compatibilidad entre los programas, se requirió el uso del sistema operativo Ubuntu 14.04 LTS (o el 16.04 LTS) junto al simulador Omnet++ versión 4.5 y Castalia versión 3.3, la mejora fue realizada sobre el protocolo CTP versión 1.1 disponible en la página: <https://code.google.com/archive/p/ctp-castalia/downloads>.

El protocolo CTP programado en Castalia, se define como un módulo compuesto, posee una comunicación entre sus módulos internos que permiten emular la implementación en TinyOS del mismo protocolo [1]. Es capaz de interactuar con algunos módulos del Castalia, sin embargo, el enrutamiento virtual para los mensajes entre módulos, no fue programado a causa de la dificultad que representaba para los autores dicha tarea, a consecuencia de esto los diferentes tipos de MAC programadas por defecto en el simulador y las funciones de distintos mensajes o variables declaradas para hacer modificaciones más profundas en el protocolo CTP, no son utilizables. Para adaptarlos se requiere una reprogramación completa del protocolo.

4.1 DISEÑO DE UN ESQUEMA DE APAGADO Y ENCENDIDO PARA EL PROTOCOLO DE RECOLECCIÓN EN ÁRBOL PARA SIMULADOR CASTALIA

El protocolo de recolección en árbol se basa en la comunicación de información desde los nodos hojas hasta los nodos raíz y finalmente al nodo principal o *sink*. En la implementación del protocolo este proceso se realiza con éxito con un gasto energético alto, haciendo que el promedio de vida de la red sea aproximadamente de 3 días incluso con el transmisor a la potencia más baja (-10 dBm), Esto se debe a que la radio se mantiene activa toda la simulación. El consumo de energía del transmisor del radio cc2420 es 17 mA a una potencia de 0 dBm [p.14, 8], si el nodo mantiene su radio encendida en todo momento, la vida útil de la red se acorta en gran proporción al compararse con una red donde se usa un esquema de apagado y encendido de nodos.

Para realizar un diseño y esquema de apagado y encendido de la radio de un nodo, se debe entender el funcionamiento completo del sistema, al saber esta información permite tener en cuenta los tiempos de funcionamiento del sistema, en este caso el proceso se realiza de esta manera:

1. Se enciende el nodo.
2. El nodo envía mensajes de reconocimiento a sus vecinos (*beacons*).
3. El nodo recibe los mensajes de respuesta (ACK).
4. Se construye una tabla de enrutamiento
5. Se verifica que existan paquetes de datos a enviar en la memoria del nodo.
6. Se envían tanto paquetes internos como externos, los internos pertenecen a los datos de las variables de medición del sensor y los externos perteneces a nodos vecinos.
7. El nodo recibe los mensajes de confirmación de llegada de los mensajes enviados.
8. El nodo se mantiene recibiendo datos de otros nodos.

Después de este proceso, se devuelve al paso 1, haciendo notar que estos procesos dependen del anterior para su realización y son los más importantes.

Un esquema de apagado y encendido de nodos basa su implementación en el ciclo de trabajo del dispositivo, donde están definidos los tiempos en que se enmarcan los procedimientos. Estos tiempos están en función del ciclo de reloj del procesador (tabla 4) y limitan el tiempo de cada acción del sistema. El CTP programado en Castalia no incluye un algoritmo de ciclo de trabajo por la complejidad de la implementación, sin esto no es posible realizar una programación de un esquema de apagado y encendido del radio de manera convencional (elaborando un algoritmo en el código en C++ y añadirlo donde se encuentre el ciclo de trabajo), y se debe analizar el código del programa antes de añadir las instrucciones del apagado y encendido de nodos.

Por otro lado, de debe entender cómo ocurre la interacción entre los módulos, esto se explica en el manual del usuario de Castalia [3], y se compone de dos lenguajes de programación, el C++ para las instrucciones del programa y el lenguaje NED que le da instrucciones a los módulos de interconexión entre ellos mismos.

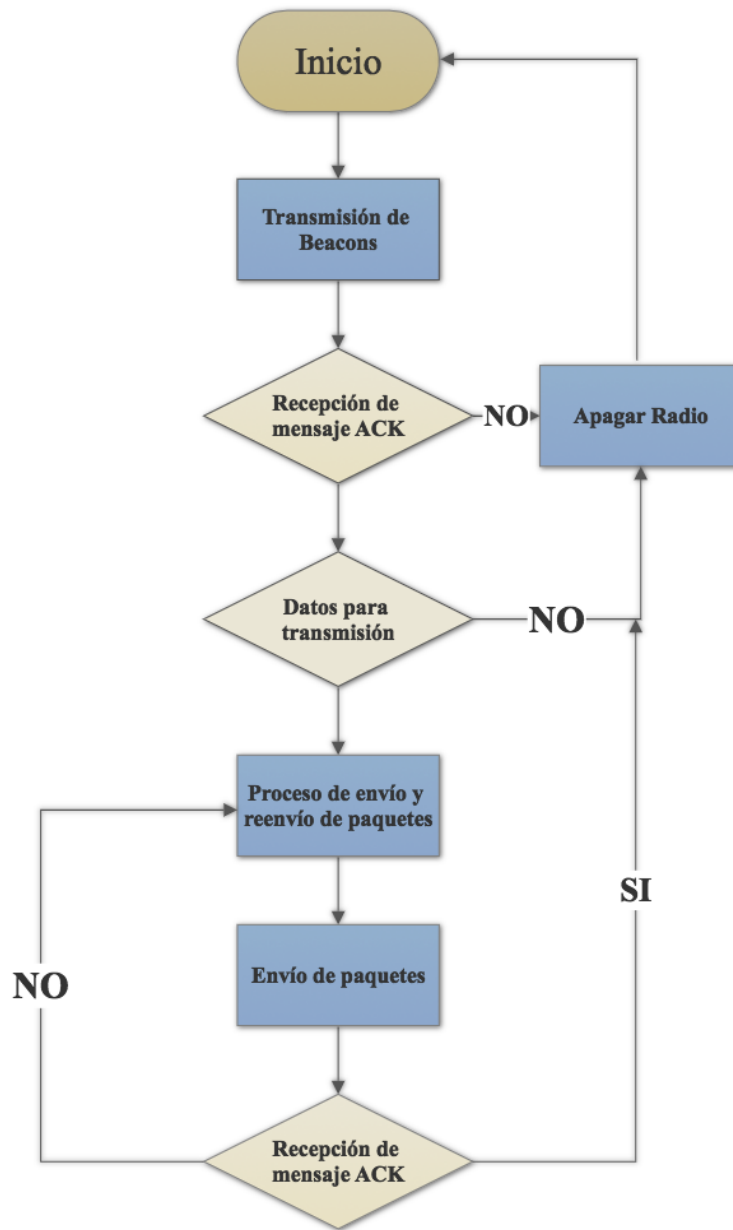


Figura 11. ALGORITMO DE APAGADO-ENCENDIDO DE LA RADIO DEL NODO PARA EL ENRUTAMIENTO DE LOS DATOS

Al analizar los procesos, se estableció un algoritmo para realizar un esquema de apagado y encendido del radio, como lo establece la figura 11. Este algoritmo se debía introducir en el módulo MAC (macCC2420) del protocolo. Debido a la modalidad en que fue programado este módulo, es incapaz de controlar la radio, ya que su realización fue considerada exclusivamente para el enrutamiento de paquetes y la interacción de la capa MAC con los mensajes ACK que

definen un comportamiento específico del protocolo, esta razón llevó a que la implementación de un apagado y encendido del radio se realizara en el módulo de aplicación del sistema.

El encendido y apagado del radio está basado en los procesos antes mencionados, cada proceso requiere un tiempo, y no ocurren sin que ocurra el proceso anterior. Si existiera la programación de un ciclo de trabajo se tendrían los tiempos definidos de los procesos mostrados en la figura 11 y se pudiese programar un esquema de apagado y encendido de los dispositivos. Por lo tanto, se busca la programación de los eventos por separado, y se establecen condiciones: cuando ocurra un evento de la simulación, el radio cambia de estado y cuando el evento termina, empieza el siguiente, se debe encender el transmisor o el receptor respectivamente a la función que deba suceder después como se muestra en la figura 5. En conclusión, la programación va a estar orientada a la ocurrencia de los eventos, debido a que no están programados los tiempos de los eventos, y los tiempos programados son orientados a la transmisión de paquetes y los tiempos de espera.

4.2 PROGRAMACIÓN DE LA PROPUESTA DE ESQUEMA DE APAGADO Y ENCENDIDO

La programación de la mejora se realizó en el módulo de aplicación, sigue el modelo del diagrama de flujo establecido en la figura 11. No se consideró alterar la programación del funcionamiento del protocolo motivado a evidenciar las cualidades del CTP con una intermitencia que conduzca a un ahorro de energía.

Originalmente se buscaba insertar la mejora en el módulo MAC ya que se encuentran las rutinas que debe seguir el dispositivo o nodo, al modificar en la capa de acceso medio se obtendrían mejores resultados porque en este se encuentra una conexión directa con la capa física (el radio, por ejemplo). Esto se intentó, pero los resultados no alcanzaban el objetivo, las razones se mencionan al final del capítulo anterior.

Finalmente se establecieron las siguientes condiciones en la capa de aplicación: si se va a enviar un paquete y este cumple los requerimientos del CTP, se procede a su envío, si no cumplen no se envía nada y se apaga el radio.

En caso de el receptor, se estableció que, si llega un paquete de datos y cumple las condiciones, este es recibido y se procede a comprobar si hay más paquetes recibiendo, si no hay más paquete y la recepción fue exitosa entonces se procede a apagar el radio.

4.3 ESCENARIOS DE PRUEBA

Para comprobar el funcionamiento de la mejora se diseñaron 3 escenarios mostrados en las figuras 12, 13 y 15, estos escenarios promueven las recolecciones de información en modo de árbol (desde el más lejano hasta el principal). A pesar del tamaño de las redes, estas topologías representan un reto para el enrutamiento de paquetes, recordando que el CTP modifica su topología de red a conveniencia de los valores ETX en la red.

Estos escenarios son simples, corresponden a una serie de nodos en línea, una topología tipo árbol, y nodos en círculo.

En el primer escenario se espera una congestión en los nodos más cercanos al nodo raíz, en el segundo se espera una recolección en árbol efectiva con distribución de paquetes uniforme, y en el tercer escenario se espera un flujo de envío de información equitativo.

En el diseño de protocolos de redes de datos, las simulaciones son un paso importante antes de la implementación. En estas se reporta el comportamiento del protocolo ante eventos aleatorios y escenarios reales; los resultados de las mismas permiten generar un criterio en el momento de tomar la decisión de implementar una estructura física de una red.

El simulador fue realizado con la visión de programar y simular protocolos de redes de sensores, esto es el paso previo de la simulación de una red con equipos específicos. Para la evaluación del esquema de encendido y apagado diseñado en este trabajo, los parámetros de simulación utilizados:

- Tiempo de simulación de 36.000 segundos.
- Potencia de transmisión a - 5 dBm.
- Modelo de radio CC2420 (programada en el CTP).
- Tasa de paquetes de 0,1; 0,5; 20.
- Retraso de encendido de 10 ms.

- Paquetes de datos de 10 bytes.
- Velocidad de transmisión de 250 kbytes/s.

Los valores mencionados previamente, a excepción de la tasa de paquetes, se usaron para todas las simulaciones presentadas en el trabajo. Cuando se estudió el CTP, se realizaron varias simulaciones de prueba donde se variaron estos parámetros, al hacer esto se pudo observar que se aumentaba el consumo de energía y la pérdida de paquetes, por lo que no se consideró pertinente cambiar estos valores debido al trabajo que era necesario realizar.

Los escenarios diseñados para realizar las simulaciones están orientados a que los nodos padres tengan un mayor consumo energético. Se apreció en simulaciones previas y no incluidas en el trabajo que, a una distancia por encima de 15 metros, la pérdida de paquetes aumentaba considerablemente, por lo tanto, esta distancia fue usada para favorecer a los nodos ubicados en la ruta deseada.

Con la finalidad de establecer una comparación más completa y detallada en este trabajo, se estudiaron varios casos de simulación. Estos escenarios son redes muy pequeñas que permiten validar el comportamiento de la red. En el simulador no es posible visualizar la ruta elegida debido a las características del protocolo que crea su topología para establecer comunicación entre los nodos, por eso los escenarios son estratégicamente diseñados, es decir, tienen nodos muy cerca y en una posición de mayor acceso al *sink* para hacer la comunicación entre ellos un poco más tangible y de esta forma lograr predecir la ruta elegida por el simulador.

4.4 ANALISIS DE RESULTADOS

Escenario 1

El escenario mostrado en la figura 11, es como se puede observar una línea recta. La comunicación establecida con el *sink* va a depender del nodo que tenga mejor ETX.

Las tablas mostradas posteriormente tabla 5, tabla 6, tabla 7, muestran los valores obtenidos con el CTP, los valores con el CTP mejorado y al final una comparación entre los nodos más relevantes.

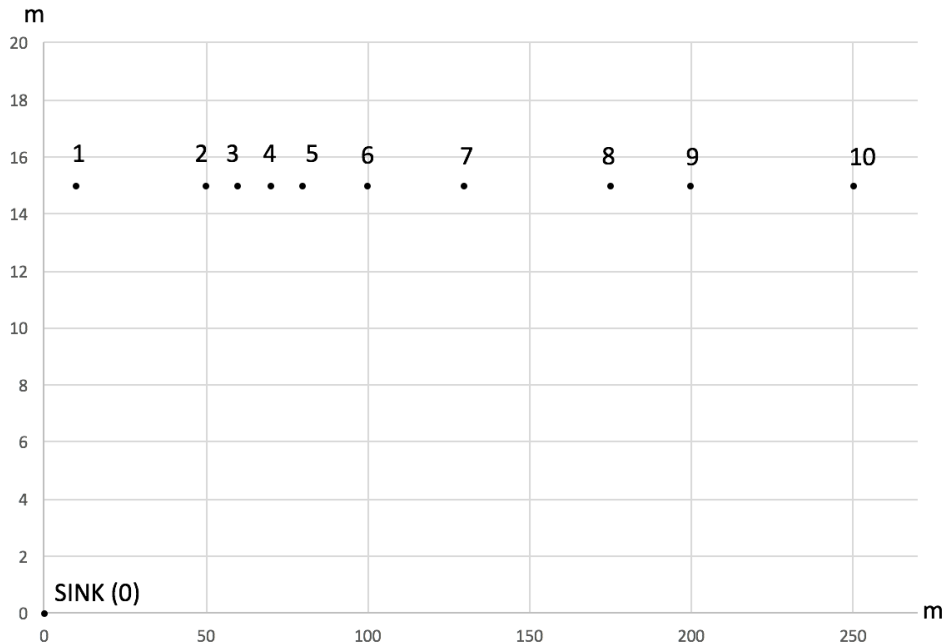


Figura 12 ESCENARIO 1

Se puede observar una línea de nodos ubicados a 15 metros de espaciado verticalmente del nodo sink, estos a su vez están espaciados horizontalmente a distancias de 10, 20, 30 y 50 metros, para lograr que los paquetes se transmitan desde el nodo 10 hasta el nodo 0 (*sink*).

La transmisión a -5 dBm en las simulaciones fue establecida para que los nodos tuviesen un rango pequeño de comunicación, obligando a que se comunicaran los nodos adyacentes. Para la tabla 5, 6 y 7 la tasa de paquetes por segundo es de 0,1.

En la tabla 5, se observa la transmisión y recepción de datos del CTP. Se muestra la cantidad de paquetes de datos transmitidos y recibidos en 36.000 segundos. Se puede diferenciar los nodos hijos como el nodo 9, nodos padres como el nodo 3 y los paquetes entregados o recibidos en el nodo *sink*.

Una vez establecido el escenario y el tiempo de simulación, el simulador emula el comportamiento del CTP programado en los archivos del simulador, junto a los valores dados en el archivo de configuración de la simulación en cuestión y al finalizar se genera una lista con todos los datos obtenidos, dando origen a los resultados mostrados en las tablas.

Nodo	Paquetes trasmitidos	Paquetes recibidos	Beacons transmitidos	Beacons Recibidos
0	420	83	420	33
1	83	0	83	0
2	166.722	212.913	166.722	203.095
3	158.439	347.666	158.439	329.846
4	137.877	452.829	427.834	137.877
5	156.299	340.554	328.767	156.300
6	166.080	186.349	166.080	182.000
7	83	0	83	0
8	181.033	178.399	181.033	178.131
9	180.368	171.532	180.368	171.262
10	83	0	83	0

Tabla 5. NODOS CTP PRIMER ESCENARIO

Nodo	Paquetes trasmitidos	Paquetes recibidos	Beacons transmitidos	Beacons Recibidos
0	389	39	389	30
1	83	0	83	0
2	166.686	198.911	166.686	198.908
3	154.463	315.480	154.463	321.478
4	128.179	430.923	128.179	430.917
5	160.703	326.295	160.703	326.286
6	171.218	187.140	171.218	187.136
7	83	0	83	0
8	189.992	186.943	189.992	186.943
9	189.233	179.779	189.233	179.779
10	83	0	83	0

Tabla 6. NODOS MEJORA CTP PRIMER ESCENARIO

En la tabla 6 se encuentran los datos transmitidos y recibidos por los nodos utilizando el esquema de apagado y encendido del radio propuesto en este trabajo.

En la tabla 7 se comparan nodos representativos de la red, estos son el nodo 9 ubicado al final, el nodo 3 a la mitad y el nodo 0 considerado el *sink*. Estos nodos se escogieron por ser un nodo hijo, un nodo padre y el nodo sumidero al cual se envía toda la información importante de la red, basándose en las explicaciones mencionadas en capítulos anteriores.

Nodos representativos	Protocolo	Paquetes Transmitidos	Paquetes Recibidos	Energía Consumida [J]
0	CTP	420	83	2.447,99
0	CTP Mejora	389	39	2.282,03
3	CTP	158.439	347.666	2.444,71
3	CTP Mejora	154.463	315.480	2.443,23
9	CTP	180.368	171.532	2.445,35
9	CTP Mejora	189.233	179.779	2.443,49

Tabla 7. COMPARACION PRIMER ESCENARIO

Como se puede observar en las tablas antes mencionadas, existe una mejora energética en el protocolo, sin embargo, las limitaciones de dicha mejora se evidencian en la transmisión y recepción de paquetes en la red, como se observa en la tabla 7, la comparación de los nodos 3 y 9 correspondientes a un nodo padre e hijo, el consumo energético no se diferencia mucho de su implementación original. La causa de esto es la congestión de mensajes y la ausencia de caminos alternos (mayor cantidad de nodos padres con un valor de ETX similar), de esta manera los mensajes pudieran transmitirse con mayor facilidad en la red.

Escenario 2

En este escenario se pretende el uso de una topología tipo árbol para favorecer el comportamiento del protocolo CTP y observar el comportamiento de la mejora y su factibilidad.

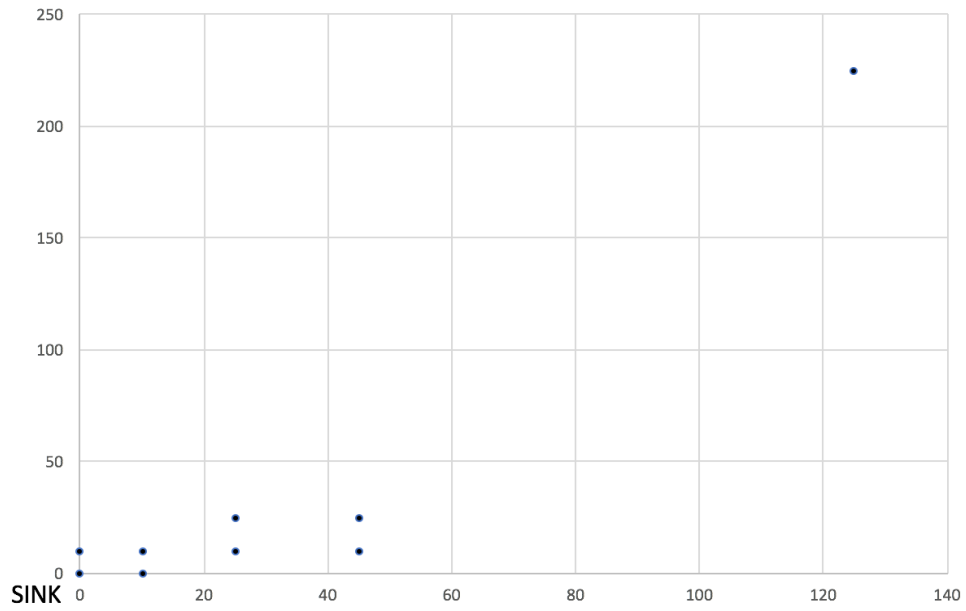


Figura 133. ESCENARIO 2

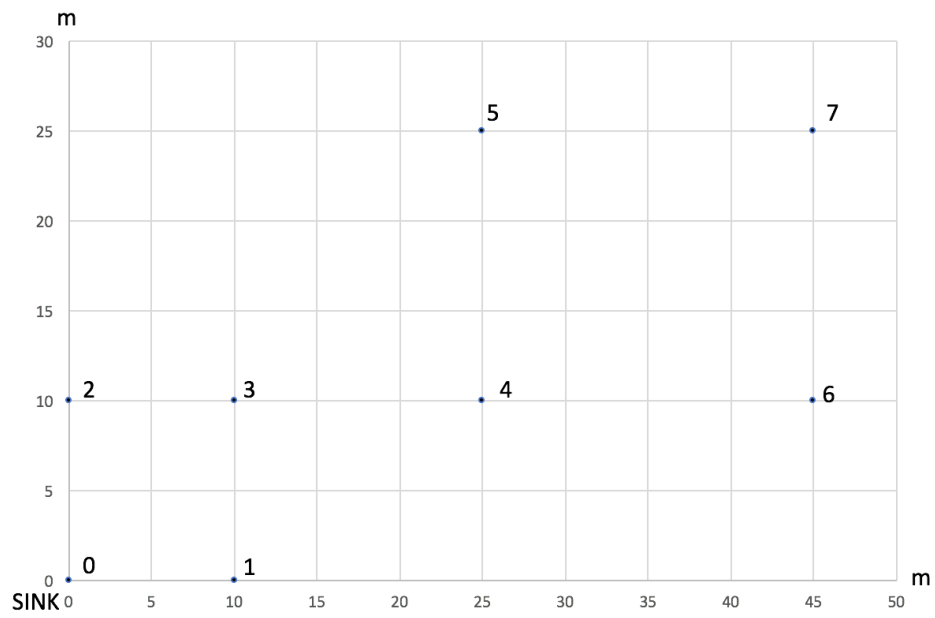


Figura 144. ESCENARIO 2

Para la siguiente tabla la tasa de paquetes por segundo es de 0,5

Nodo	Paquetes Trasmitidos	Paquetes Recibidos	Beacons Transmitidos	Beacons Recibidos
0	1.908	2.822	83	290
1	342	3.604	84	284
2	856	4.276	84	376
3	361	5.811	83	446
4	1.936	3.911	85	304
5	1.126	3.023	93	289
6	372	1908	85	262
7	370	2.278	84	205
8	83	0	83	0

Tabla 8. NODOS CTP SEGUNDO ESCENARIO

Nodo	Paquetes Trasmitidos	Paquetes Recibidos	Beacons Transmitidos	Beacons Recibidos
0	87	694	83	316
1	3.541	10.285	884	341
2	4.685	11.724	439	1053
3	5.543	12.263	396	1.233
4	2.781	10.064	240	882
5	2.199	10.339	226	808
6	2.659	10.293	297	921
7	1.420	6.345	215	532
8	83	0	83	0

Tabla 9. NODOS MEJORA CTP SEGUNDO ESCENARIO

Nodos representativos	Protocolo	Paquetes recibidos	Paquetes Transmitidos	Energía Consumida[J]
0	CTP	1.908	2.822	2.448
0	CTP Mejora	87	694	2.432,74
3	CTP	361	5.811	2448
3	CTP Mejora	5.543	12.263	2.232,13
7	CTP	370	2.278	2.448
7	CTP Mejora	1.420	6.345	2.231,86

Tabla 10. COMPARACION SEGUNDO ESCENARIO

Se observó en los resultados de las tablas 8, 9 y 10 un aumento en la transmisión de paquetes, la causa de esto es que después de la transición del modo dormir a la de transmisión se debe repetir el proceso de enviado de *beacons* para actualizar la tabla, incluso después de la transición de modo dormir a modo de recepción, al terminar de recibir los paquetes se pasa a modo transmisión para actualizar las tablas de enrutamiento y del estimador de enlace, de esta manera el protocolo asegura la comunicación entre los nodos y el envío de mensajes al nodo *sink*.

Se aprecia como el nodo sink consume menos energía en la mejora y reduce la cantidad de paquetes que transmite y que se reciben, a consecuencia del tiempo en modo dormir, se descartan gran cantidad de paquetes y un ahorro en energía de transmisión que incide directamente en el consumo.

Escenario 3

El escenario 3 está compuesto por 8 nodos, estos se encuentran en forma de circulo, con el nodo sumidero en un extremo, de manera que se generen nodos padres equidistantes entre si, dando rutas distintas a la información para llegar a su destino.

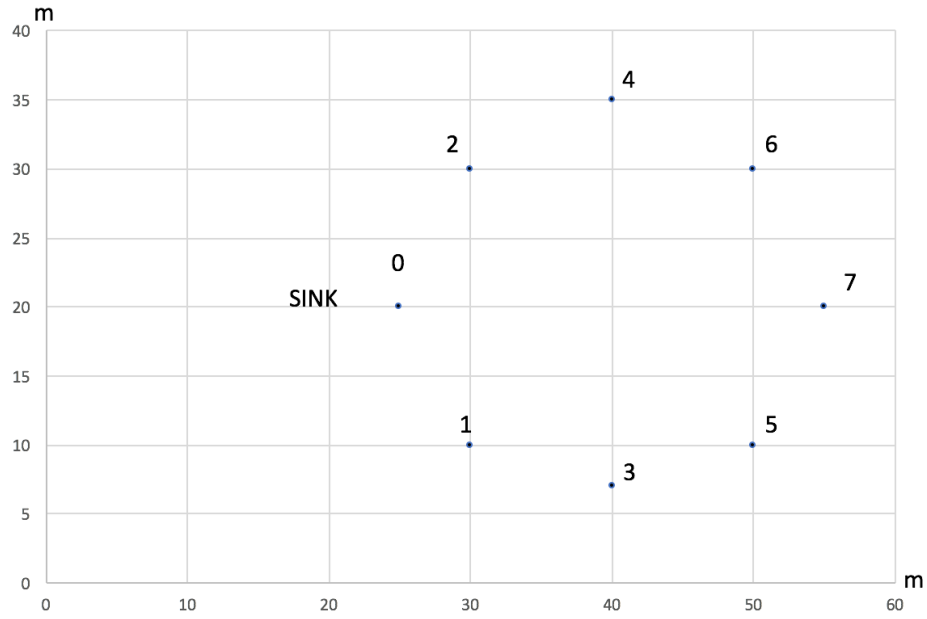


Figura 155. ESCENARIO 3

Nodo	Paquetes Trasmitidos	Paquetes Recibidos	Beacons Transmitidos	Beacons Recibidos
0	1954	2.775	82	362
1	888	3.884	83	409
2	822	4.098	83	391
3	487	4.567	83	396
4	366	3.272	83	254
5	346	2.639	83	360
6	389	3.987	83	348
7	716	3.236	83	323

Tabla 11. NODOS CTP TERCER ESCENARIO

Para la siguiente tabla la tasa de paquetes por segundo es de 0,1

Nodo	Paquetes transmitidos	Paquetes Recibidos	Beacons Transmitidos	Beacons Transmitidos
0	89	2.343	84	483
1	46.939	1.988,84	4.031	12.560
2	46.629	184.611	4.068	13.296
3	56.963	171.410	4.616	13.921
4	20.706	96.067	2.447	7.442
5	40.262	1210.32	3.602	13.065
6	49.917	161.861	4.103	12.777
7	37.551	174.193	3.543	13.077

Tabla 12. NODOS MEJORA CTP TERCER ESCENARIO

Nodos representativos	Protocolo	Paquetes Transmitidos	Paquetes Recibidos	Energía Consumida [J]
0	CTP	1.952	2.775	2.447,99
0	CTP Mejora	89	2.343	2.269,17
2	CTP	822	4.098	2.447,99
2	CTP Mejora	46.629	184.611	2.447,09
7	CTP	716	3.236	2.447,99
7	CTP Mejora	37.551	174.193	2.447,32

Tabla 13. COMPARACION TERCER ESCENARIO

En este escenario se observa un flujo de mensajes similar entre nodos, esto se debe a que es una topología tipo espejo respecto al *sink*, se puede notar que los nodos a excepción del nodo 7 tienen la misma distancia creando nodos con valor de ETX igual, esto asegura varias rutas al nodo sumidero generando una distribución del consumo de energía homogéneo entre nodos. Por otro lado la cercanía entre ellos genera un problema, al enviarse los mensajes *beacons* entre ellos se

genera un flujo de mensajes muy grande y el descarte de los mensajes duplicados trae la consecuencia de un desgaste mayor, por lo tanto el apagado y encendido de nodos incrementa el consumo energético, reflejando una de las grandes desventajas del CTP respecto al envío excesivo de mensajes para mantener la red actualizada. Este envío excesivo puede ocasionar una cantidad considerable de paquetes descartados que incomunicaría la red o no arrojaría resultados fiables, por otro lado, se pueden atribuir estos resultados a la programación implementada, que genera la repetición de un ciclo con pocos tiempos de espera entre procesos que crea una desventaja en redes con nodos relativamente cercanos entre sí.

Es notable mencionar que, a pesar del incremento del consumo energético, este escenario fue el que se desempeñó mejor a nivel de comunicación entregando más paquetes al nodo *sink*, esto se debe a que posee dos nodos padres con el mismo nivel de ETX, demostrando la importancia de tener varios nodos enrutadores cercanos al nodo sumidero.

Respuesta del CTP MEJORA respecto a diferentes relaciones de paquetes por segundo:

El protocolo CTP establece su topología a través del envío y recepción de mensajes, por lo tanto, la cantidad de mensajes por segundo que envíe cada nodo es relevante para la investigación. Para este análisis se escogió el escenario 2 de la figura 13 debido a que representa una arquitectura de red tipo árbol, donde el CTP tiene ventajas; se escogieron 3 tasas de envío de paquetes: 0.1, 0.5 y 20 paquetes por segundo y se escogieron 2 nodos representativos de la red para evaluar el desempeño de la mejora respecto a su original (CTP). Los resultados se encuentran en la tabla 14.

Los nodos escogidos representan el valor máximo y mínimo energético útiles de la red, estos valores definen el tiempo de duración de la red. Se debe mencionar que la potencia de transmisión es -5 dBm, para esta potencia el nodo *sink* solo puede transmitir a los nodos 1, 2 y 3, mientras que el 3 puede comunicarse con el 4, 5, 6 y 7 y a su vez con el 2, 1 y *sink*, esto quiere decir que el gasto energético máximo de la red lo tendrá ese nodo.

Relación paquetes por segundo	Nodos	Protocolo	Paquetes Transmitidos	Paquetes Recibidos	Energía Consumida [J]	Ahorro Energético
0.1	3	CTP	354	5.853	2.448	8,85 %
	3	MEJORA	77.001	189.703	2231,3	
	7	CTP	526	1.855	2.448	8,86 %
	7	MEJORA	19.129	101.076	2231	
0.5	3	CTP	361	5.811	2.448	8,81 %
	3	MEJORA	5.543	12.263	2.232,13	
	7	CTP	370	2.278	2.448	8,82 %
	7	MEJORA	1.420	6.345	2.231,86	
20	3	CTP	719	3.544	2.447,99	8,82 %
	3	MEJORA	2.976	6.051	2232	
	7	CTP	397	1.467	2.447,99	8,84 %
	7	MEJORA	649	2.520	2231,4	

Tabla 14 Comparación CTP vs MEJORA a diferentes tasas de paquetes por segundo

El uso de la optimización de la mejora estará sujeta a la aplicación que elija el usuario.

En líneas generales, los 3 escenarios de la mejora tienen la misma duración de la vida útil de la red en cuanto a la tasa de transmisión y recepción de mensaje los resultados son muy similares, al momento de establecer una comparación para poder determinar cuál de los escenarios involucra las mejores características de simulación, se puede observar que el escenario 3 ofrece una mayor transmisión de paquetes efectivos de la red hacia el *sink*.

Son coherentes los resultados en cuanto a la topología, debido a que por la forma del escenario de la simulación 3 (círculo), la cantidad de saltos de la ruta entre un nodo y el nodo sumidero es menor debido a su valor de ETX, haciendo que la red sea más optimizada y de esta forma más efectiva.

En los otros escenarios, los valores de ETX son más variables, ya que al este depender de la cantidad de nodos recibidos y transmitidos en cada nodo y a su vez de la posición del nodo respecto al *sink*, no se considera tan efectiva la red, debido a que hay mayor probabilidad de pérdida de paquetes, como consecuencia de los nodos poco equidistantes entre sí.

Los resultados reflejan coherencia con lo explicado en los capítulos anteriores, se recuerda que el protocolo CTP realiza la topología de red cada vez que va a enviar mensajes de datos y para esto necesita una cantidad de mensajes en tiempo real para mantener la topología lo más actualizada posible, esto genera una cantidad exagerada de paquetes que a su vez obligan al nodo

a estar en un tiempo de recepción entre el 80 y 98 % de su tiempo de simulación, en su programación indica que si está recibiendo paquetes está obligado a estar en modo de recepción. Esto causa que la acción del esquema de apagado y encendido funcione aproximadamente la primera hora, después el incremento de paquetes en la red aumenta considerablemente, esto hace que solo cambie a modo de transmisión cuando se llena el buffer del dispositivo. Es importante mencionar que la transmisión de paquetes ocupa un tiempo entre 15 y 30 mili segundos.

En las tablas 5, 6, 8, 9, 11 y 12 se observa que mientras la topología de red ofrece menos opciones de rutas, la congestión es mayor, esto se debe a que los valores de ETX en los nodos obligan a seguir la dirección de ruta como se aprecia en la figura 12 y en la tabla 5 y 6, donde la congestión incomunica al nodo 1.

Mientras mayor es la cantidad de nodos alrededor de nodo *sink* se evidencia que existe una distribución más homogénea de la energía consumida, esto se puede apreciar en las tablas 7, 10 y 13. Mientras los nodos más cerca estén la energía de cada nodo será muy parecida, pero esto aumentará la congestión de la red en esos puntos.

En los escenarios de las figuras 13 y 15 se observan topologías que ofrecen ventajas al protocolo y se reflejan en los resultados en las tablas 8, 9, 11 y 12. Mientras la distribución de los nodos sea lo más homogénea posible, estos tendrán varias rutas hacia el nodo principal (*sink*).

Se verificó que la relación de envío de paquetes por segundo es de gran importancia en el protocolo; mientras la tasa de paquetes por segundo sea más baja, el protocolo es más efectivo, la congestión disminuye, el consumo de energía disminuye y la vida útil de la red aumenta, sin embargo, se sacrifica la cantidad de información a recolectar, como se aprecia en la tabla 14.

El esquema de apagado y encendido se mejora las condiciones de la red, ahorrando una cantidad considerable de energía, se observa en la tabla 14, como para distintos valores de paquetes por segundo se ahorra un aproximado de 9% de energía transmitiendo incluso mayor cantidad de paquetes.

Es de suma importancia mencionar que cada vez que el nodo entra en modo dormir, al salir de este modo, se genera una transmisión de paquetes de reconocimiento que contribuye con el aumento de la congestión, pero se debe tener en cuenta que el estado de transmisión en un nodo

consume menos energía que el estado de recepción, lo cual al final redonda en una disminución del consumo energético de la red.

CONCLUSIONES Y RECOMENDACIONES

Luego de haber culminado el trabajo, se pueden establecer algunas conclusiones y recomendaciones:

- El CTP es un protocolo muy robusto y confiable a la hora de recolectar la información de cada nodo en una red. Esto se debe en parte a que siempre los nodos están escuchando la red por si existen mensajes a enviar, y de manera permanente se están actualizando las tablas de enrutamiento y las rutas. Igualmente considera rutas alternativas si existe congestión en alguna de las establecidas.
- Es factible implementar un esquema de apagado y encendido de nodos en el protocolo CTP, aunque por ser un protocolo que busca mantener las rutas actualizadas, esto ocasiona que el encendido de los nodos genere un gran número de paquetes, aumentando el descarte y pérdida de mensajes. En este sentido, si la variable a medir es de cambios bruscos, es mejor no activar el esquema de apagado y encendido. Si por el contrario la variable a medir puede mantener intervalos de medición largos, entonces es factible apagar los nodos durante esos intervalos, para fomentar el ahorro energético.
- En vista de la congestión y pérdida de paquetes que genera el esquema de apagado y encendido, es recomendable aplicarlo con miras a reducir el consumo en la red si existen muchas rutas factibles para transmitir los paquetes al nodo *sink*. En caso de que exista una sola ruta para transmitir la información, no es recomendable aplicar un esquema de apagado y encendido de nodos con el protocolo CTP.
- Es importante tomar en cuenta el ciclo de trabajo dentro del sistema operativo del nodo, ya que conociendo el tiempo que le toma al dispositivo realizar cada función preestablecida (transmisión, recepción, etc...) es la clave para las mejoras energéticas y aprovechar la batería al máximo.

- El modo recepción de un nodo es el que genera mayor consumo energético. En este sentido se debe limitar el tiempo que los nodos pasan en este estado, para de esta manera aumentar la vida de la red.
- Establecer en futuros trabajos un temporizador en el modo de recepción de mensajes para ampliar el tiempo en otros estados del radio, sincronizado con un tiempo en espera de los eventos de envío de mensajes de los nodos, reduciendo el consumo y la pérdida de paquetes.
- Existen discrepancias en la implementación del protocolo CTP en Castalia respecto a la documentación del trabajo realizado [2] y el trabajo en el cual se basó [1] la implementación. Se recomienda trabajar directamente con la implementación original para verificar resultados.
- Para futuros trabajos a desarrollar en base al CTP implementado en Castalia es importante tomar en cuenta que muchos módulos de esta implementación no permiten modificaciones por la manera en que fueron programados, e incluso no toman en cuenta los parámetros que considera el simulador Castalia. En general, los desarrolladores de esta implementación hicieron algunas simplificaciones del protocolo que no permiten modificar aspectos importantes del mismo, a menos que se hagan cambios significativos en la programación original del protocolo.
- Cada vez que se vaya a realizar una investigación o un proyecto en software libre o código abierto, establecer comunicación previa con los autores del proyecto en el cual se va a basar el trabajo, para verificar y considerar un panorama objetivo de la factibilidad y el alcance de la investigación independientemente de lo que diga la publicación o el trabajo de investigación base.

REFERENCIAS BIBLIOGRAFICAS

1. **Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, Sukun Kim, Philip Levis, and Alec Woo.** TinyOS Enhancement Proposal (TEP) 123: The Collection Tree Protocol (CTP).
2. **U. Colesanti and S. Santini.** "A performance evaluation of the collection tree protocol based on its implementation for the Castalia wireless sensor networks, Simulator. Technical Report No.681, Department of Computer Science, ETH Zurich, Zurich, Switzerland,
3. Castalia 3.0 User's Manual <http://castalia.npc.nicta.com.au/documentation.php>
4. **Boulis, Athanassios.** *Castalia - A simulator for Wireless Sensor Networks.* marzo 2011.
5. Omnet++ 4.5. *Simulation manual.* [Online] <https://omnetpp.org/doc/omnetpp/manual/>.
6. Wireless sensor networks : Concepts, Applications, Experimentation and Analysis \ **Fahmy, Hossam Mahmoud Ahmed** . - 1st ed. - New York, NY : Springer Berlin Heidelberg, 2016. - 614 p. : ill. ; 24 cm.
7. **Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, and Philip Levis.** CTP: Robust and Efficient Collection through Control and Data Plane Integration. Technical report, The Stanford Information Networks Group (SING), 2008. <http://sing.stanford.edu/pubs/sing-08-02.pdf>.
8. ChipCon 2420 Datasheet. <http://focus.ti.com/lit/ds/symlink/cc2420.pdf>.
9. **Regoli, C.** Diseño de un método de enrutamiento activo para redes inalámbricas de sensores (WSN). Caracas, Venezuela: Universidad Central de Venezuela, 2014.
10. **Shio Kumar Singh, M P Singh , and D K Singh.** "Routing Protocols in Wireless Sensor Networks – A Survey" International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.2, November 2010.
11. **W.Heinzelman, A.Chandrakasan, and H.Balakrishnan,**"Energy-efficient Routing Protocols for Wireless Microsensor Networks," in Proc. 33rd Hawaii Int. Conf. SystemSciences (HICSS), Maui, HI, Jan.2000.
2. **Estrin D. Heidemann J. Intanagonwiwat Ch., Govindan R. and Silva F.** Directed diffusion for wireless sensor networking. IEEE/ACM Transactions on networking, 11(1), febrero 2003.
3. **M. Sanjuan y A. Yury,** Diseño y desarrollo de un sistema de alerta temprana para la detección de incendios forestales, usando una red de sensores inalámbricos. Universidad Central de Venezuela, Caracas, Venezuela, 2010.

4. **M. Castillo-Effer, D. Quintela, W. Moreno, R. Jordan y W. Westhoff**, Wireless sensor networks for flash-flood alerting, Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems., vol. 1, pp. 142-146, 2004.
5. **J. Pérez, E. Urdaneta y A. Custodio**, Metodología para el diseño de una red de sensores inalámbricos, Universidad Nacional Politécnica Antonio José de Sucre, Caracas, Venezuela, 2014.
6. **M. Landaeta y R. González**, Interfaz de monitoreo para una red inalámbrica de sensores, Universidad Simón Bolívar, Caracas, 2008.
7. **C. B. Abbas, R. Gonzalez, N. Cardenas y L. J. G. Villalba**, A proposal of a Wireless Sensor Network Routing Protocol, de Personal wireless communications, Praga, Springer, 2007, pp. 410-422.
8. **C. Barenco, N. Cárdenas, G. Lobalsamo y N. Davila**, SHRP: A New Routing Protocol to Wireless Sensor Networks, Advances in computer science and engineering, Springer, pp. 138-146.
9. **R. Gonzalez, M. Huerta, R. Clotet y G. Figueras**, Comparación de Gasto de Energía entre Nodos Tradicionales de Redes Inalámbricas de Sensores y Teléfonos Inteligentes, de Memorias del 6to Congreso Iberoamericano de Estudiantes de Ingeniería Eléctrica (VI CIBELEC 2015), Mérida, Venezuela, 2015.
10. **M. Hernández, D. Marqués y W. Torres**, Evaluación de alternativas de Energy Harvesting para optimizar la autonomía de nodos WSN en ambientes interiores de plantas industriales, Universidad Católica Andrés Bello, Caracas, Venezuela, 2014.
11. **C. Ciammaricone, E. Muguregui y W. Torres**, Análisis del estándar de comunicaciones Zigbee en redes inalámbricas sensoriales (WSN) de alta movilidad, Universidad Católica Andrés Bello, Caracas, Venezuela, 2009.
12. **Á. Daboín, G. Verde, F. Torrealba y T. Gharbi**, Medición de RSSI, LQI y pruebas de cobertura para diferentes escenarios de propagación en una red inalámbrica de sensores, Revista Digital de Investigación y Postgrado, UNEXPO, vol. 2, nº 1, 2012.

BIBLIOGRAFIA

- **J. Al Karaki y A. Kamal**, Routing Techniques in Wireless Sensor Networks: A survey, Wireless Communications, pp. 6-28, 2004.
- **B.Nayyar y R. Singh**, A Comprehensive Review of Simulation Tools for Wireless Sensor Networks (WSNs), Journal of Wireless Networking and Communications, vol. 5, n° 1, pp. 19-47, 2015.
- **Kumar y O. Prakash**, Recent development in simulations tools for WSNs: An analytical study, de Simulations Technologies in Networking and Communications. Selecting the best tool for the test, CRC Press, 2014, pp. 495-518.
- **Dwivedi y O. Vyas**, Wireless Sensor Network: At a Glance, de Recent Advances in Wireless Communication Networks, Intechopen, 2008, pp. 299-327.
- **Akkaya K, Younis M**, A survey on routing protocols for wireless sensor networks: Ad Hoc Networks 3 (2005) Elsevier, 325–349.
- **Jingjing Zhang, Zhenqi Yang**, Baowei Wang, Huiyu Sun and Xingming Sun ,E-CTP: An Energy-balanced Collection Tree Protocol for Power Constrained Wireless Sensor Networks, International Journal of Grid and Distributed Computing. Vol.7, No.2 (2014), pp.115-126 [5] OMNeT++ User Manual (Version 4.3).
www.omnetpp.org/doc/omnetpp33/manual/usman.html.
- **Pediaditakis D, Tselishchev Y, Boulis A**, Performance and Scalability Evaluation of the Castalia Wireless Sensor Network Simulator: Proceedings of the 3rd International ICST Conference on Simulation Tools.
- **Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan**, Energy Efficient Communication Protocol for Wireless Microsensor Networks, Proceedings of the Hawaii, International Conference on System Sciences, January 4-7, 2000, Maui, Hawaii.

ANEXOS

- Imagen que reflejan el encendido y apagado de los nodos en las trazas de las simulaciones.

```
150.023912710084SN.node[7].Communication.Radio      completing transition to 2 (SLEEP)
150.024467763591SN.node[7].Communication.Radio      END signal from node 2
150.039512655194SN.node[7].Communication.Radio      START signal from node 3 , received power -94.1502dBm
150.039512655194SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 3, radio not in RX state
150.039864655194SN.node[7].Communication.Radio      END signal from node 3
150.049682464983SN.node[7].Communication.Radio      START signal from node 3 , received power -94.1502dBm
150.049682464983SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 3, radio not in RX state
150.050994464983SN.node[7].Communication.Radio      END signal from node 3
150.051014464982SN.node[7].Communication.Radio      START signal from node 4 , received power -93.3735dBm
150.051014464982SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 4, radio not in RX state
150.051366464982SN.node[7].Communication.Radio      END signal from node 4
150.052465754564SN.node[7].Communication.Radio      START signal from node 2 , received power -96.227dBm
150.052465754564SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 2, radio not in RX state
150.052818575335SN.node[7].Communication.Radio      START signal from node 4 , received power -93.3735dBm
150.052818575335SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 4, radio not in RX state
150.053777754564SN.node[7].Communication.Radio      END signal from node 2
150.054130575335SN.node[7].Communication.Radio      END signal from node 4
150.064216811568SN.node[7].Communication.Radio      START signal from node 3 , received power -94.1502dBm
150.064216811568SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 3, radio not in RX state
150.064568811568SN.node[7].Communication.Radio      END signal from node 3
150.075386321025SN.node[7].Communication.Radio      START signal from node 3 , received power -94.1502dBm
150.075386321025SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 3, radio not in RX state
150.076698321025SN.node[7].Communication.Radio      END signal from node 3
150.076718321024SN.node[7].Communication.Radio      START signal from node 4 , received power -93.3735dBm
150.076718321024SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 4, radio not in RX state
150.077070321024SN.node[7].Communication.Radio      END signal from node 4
150.083943519028SN.node[7].Communication.Radio      START signal from node 4 , received power -93.3735dBm
150.083943519028SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 4, radio not in RX state
150.085255519028SN.node[7].Communication.Radio      END signal from node 4
150.085275519027SN.node[7].Communication.Radio      START signal from node 5 , received power -89.3853dBm
150.085275519027SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 5, radio not in RX state
150.085627519027SN.node[7].Communication.Radio      END signal from node 5
150.085841160576SN.node[7].Communication.Radio      START signal from node 5 , received power -89.3853dBm
150.085841160576SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 5, radio not in RX state
150.087153160576SN.node[7].Communication.Radio      END signal from node 5
150.091408257454SN.node[7].Communication.Radio      START signal from node 2 , received power -96.227dBm
150.091408257454SN.node[7].Communication.Radio      Failed packet (WC_SIGNAL_START) from node 2, radio not in RX state
150.092720257454SN.node[7].Communication.Radio      END signal from node 2
150.096302809523SN.node[7].Communication.Radio      Buffered [CC2420 data packet] from MAC layer
150.096302809523SN.node[7].Communication.Radio      SET STATE to TX, delay=0.000194, power=62
```

- Imagen que refleja una simulación sin esquema de apagado y encendido para el mismo tiempo de simulación

```

150.029905005132SN.node[7].Communication.Radio SET STATE to TX, delay=1e-05, power=62
150.029915005131SN.node[7].Communication.Radio completing transition to 1 (TX)
150.029915005131SN.node[7].Communication.Radio Sending Packet, Transmission will last 0.001312 secs
150.031227005131SN.node[7].Communication.Radio TX finished (no more pkts in the buffer) changing to RX
150.031227005131SN.node[7].Communication.Radio SET STATE to RX, delay=1e-05, power=62
150.03123700513 SN.node[7].Communication.Radio completing transition to 0 (RX)
150.049286395107SN.node[7].Communication.Radio START signal from node 3 , received power -94.1502dBm
150.050598395107SN.node[7].Communication.Radio END signal from node 3
150.050598395107SN.node[7].Communication.Radio Failed packet (WC_SIGNAL_END) from node 3, NO interference
150.050618395106SN.node[7].Communication.Radio START signal from node 0 , received power -97.5231dBm
150.050618395106SN.node[7].Communication.Radio Failed packet (WC_SIGNAL_START) from node 0, below sensitivity
150.050970395105SN.node[7].Communication.Radio END signal from node 0
150.051519610899SN.node[7].Communication.Radio START signal from node 2 , received power -96.227dBm
150.051519610899SN.node[7].Communication.Radio Failed packet (WC_SIGNAL_START) from node 2, below sensitivity
150.052831610899SN.node[7].Communication.Radio END signal from node 2
150.052851610898SN.node[7].Communication.Radio START signal from node 0 , received power -97.5231dBm
150.052851610898SN.node[7].Communication.Radio Failed packet (WC_SIGNAL_START) from node 0, below sensitivity
150.053203610897SN.node[7].Communication.Radio END signal from node 0
150.054055829305SN.node[7].Communication.Radio START signal from node 5 , received power -89.3853dBm
150.055367829305SN.node[7].Communication.Radio END signal from node 5
150.055367829305SN.node[7].Communication.Radio Received packet (WC_SIGNAL_END) from node 5, with no interference
150.055387829304SN.node[7].Communication.Radio START signal from node 4 , received power -93.3735dBm
150.055739829303SN.node[7].Communication.Radio END signal from node 4
150.055739829303SN.node[7].Communication.Radio Failed packet (WC_SIGNAL_END) from node 4, NO interference
150.061230692284SN.node[7].Communication.Radio Buffered [CC2420 data packet] from MAC layer
150.061230692284SN.node[7].Communication.Radio SET STATE to TX, delay=1e-05, power=62
150.061240692283SN.node[7].Communication.Radio completing transition to 1 (TX)
150.061240692283SN.node[7].Communication.Radio Sending Packet, Transmission will last 0.001312 secs
150.062552692283SN.node[7].Communication.Radio TX finished (no more pkts in the buffer) changing to RX
150.062552692283SN.node[7].Communication.Radio SET STATE to RX, delay=1e-05, power=62
150.062562692282SN.node[7].Communication.Radio completing transition to 0 (RX)
150.063348691999SN.node[7].Communication.Radio START signal from node 6 , received power -87.9507dBm
150.064529022111SN.node[7].Communication.Radio START signal from node 4 , received power -93.3735dBm
150.064660691999SN.node[7].Communication.Radio END signal from node 6
150.064660691999SN.node[7].Communication.Radio Received packet (WC_SIGNAL_END) from node 6, despite interference
150.065841022111SN.node[7].Communication.Radio END signal from node 4
150.065841022111SN.node[7].Communication.Radio Failed packet (WC_SIGNAL_END) from node 4, with interference
150.094474045199SN.node[7].Communication.Radio Buffered [CC2420 data packet] from MAC layer
150.094474045199SN.node[7].Communication.Radio SET STATE to TX, delay=1e-05, power=62
150.094484045198SN.node[7].Communication.Radio completing transition to 1 (TX)

```

- Algoritmo de funcionamiento real del Radio cc2420 [8]:

