

**Centro de Investigación Científica y de Educación
Superior de Ensenada, Baja California**



**Maestría en Ciencias en
Electrónica y Telecomunicaciones con
orientación en Telecomunicaciones**

**Evaluación del desempeño en la transmisión de señales
biomédicas en un ambiente inalámbrico en redes 6LoWPAN**

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de
Maestro en Ciencias

Presenta:

Andrea Murillo Carrillo

Ensenada, Baja California, México

2017

Tesis defendida por

Andrea Murillo Carrillo

y aprobada por el siguiente Comité

Dr. Raúl Rivera Rodríguez

Codirector del Comité

M.C. Edwin Martínez Aragón

Codirector del Comité

Miembros del Comité

Dr. Salvador Villarreal Reyes

Dr. Miguel Ángel Alonso Arévalo

Dr. Heriberto Márquez Becerra



Dr. Miguel Ángel Alonso Arévalo

Coordinador del Programa de Posgrado en
Electrónica y Telecomunicaciones

Dra. Rufina Hernández Martínez

Director de Estudios de Posgrado

Andrea Murillo Carrillo © 2017

Queda prohibida la reproducción parcial o total de esta obra sin el permiso formal y explícito del autor y director de la tesis

Resumen de la tesis de **Andrea Murillo Carrillo**, presentada como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con orientación en Telecomunicaciones.

Evaluación del desempeño en la transmisión de señales biomédicas en un ambiente inalámbrico en redes 6LoWPAN

Resumen aprobado por:

Dr. Raúl Rivera Rodríguez
Codirector de Tesis

M.C. Edwin Martínez Aragón
Codirector de Tesis

En los últimos años se ha presentado una proliferación de dispositivos con recursos limitados pero capaces de medir, inferir e incluso interpretar indicadores de su entorno. Los cuales además operan en redes de gran escala que les permiten la interconexión para el intercambio de información. Para lograr esta interconexión a redes preexistentes de gran escala como las del tipo IP, comúnmente se requiere del diseño e implementación de gateways especializados. Aumentando la complejidad de la red e introduciendo retardos asociados al proceso de interconexión de las dos redes.

En este contexto, y con la finalidad de establecer una interconectividad más directa, surge el estándar conocido como 6LoWPAN, el cual provee una pila de protocolos enfocados a la adaptación entre redes basadas en IPv6 y diversas tecnologías de comunicación inalámbrica siendo la principal la IEEE 802.15.4.

Este paradigma de interconexión a gran escala de todo tipo de dispositivos, conocido como el Internet de las Cosas (IoT), no ha pasado desapercibido para el área de la e-Salud por parte de la industria y comunidad académica, aplicado especialmente para el telemonitoreo de variables fisiológicas. En publicaciones recientes se ha analizado la factibilidad de emplear redes 6LoWPAN para el monitoreo remoto de variables fisiológicas. No obstante, el análisis presentado de redes 6LoWPAN/IEEE 802.15.4 enfocadas al cuidado de la salud encontradas hasta el momento, se enfocan o bien en el análisis de aspectos concernientes a la capa de Aplicación en el modelo de referencia OSI, o a los concernientes a la capa de Acceso al Medio (MAC); dejando fuera aspectos importantes presentes en la implementación de las capas intermedias.

En este trabajo se propone evaluar la implementación del estándar 6LoWPAN con un enfoque hacia la implementación de una aplicación de telemonitoreo médico, con el fin de evaluar su desempeño y comportamiento en términos del caudal eficaz y el número de usuarios concurridos que se pueden ser servidos. Esto sin dejar fuera aspectos que emergen al momento de la implementación de las diversas capas del estándar.

Palabras clave: telemonitoreo médico, 6LoWPAN/IEEE 802.15.4, Contiki OS, Internet de las Cosas.

Abstract of thesis presented by **Andrea Murillo Carrillo**, as a partial requirement to obtain the Master of Science degree in Electronics and Telecommunications with orientation in Telecommunications.

Evaluation of performance in the transmission of biomedical signals in a Wireless environment in 6LoWPAN

Abstract approved by:

Dr. Raúl Rivera Rodríguez
Thesis Codirector

M.C. Edwin Martínez Aragón
Thesis Codirector

In recent years, there has been an increase of devices with limited resources, but capable of measure, infer and even interpret their environment variables. They also operate within wide networks that enables the interconnection for the information exchange. To achieve this interconnection into preexisting IP networks, it is required to design and implement specialized gateways. This make the network implementation more complex and increasing the delays related to the interconnection process of both networks. In this context, 6LoWPAN emerges as a standard with the purpose of stablish a more direct interconnectivity. This, through a stack of protocols that focus on the adaptation between IPv6 based networks and several communication technologies, being the main one the IEEE 802.15.4.

This model of wide interconnection of all kind of devices, known as Internet of Thing (IoT), has gained attention from the industrial and academic community dedicated to the telemonitoring of physiological variables (e-Health). Nowadays, there are several related works of 6LoWPAN networks that analyze the feasibility of its application in telemonitoring of physiological variables, however, these analysis centers their attention into aspects related to the Application Layer (OSI Model) or Media Access Layer (MAC); leaving out important aspects that can be found at the implementation of intermediate layers.

This thesis proposes the evaluation of a 6LoWPAN's implementation oriented to healthcare telemonitoring application, with the aim to evaluate the performance and behavior in terms of throughput and number of concurred users that can be served, this considering aspects found at the implementation.

Keywords: healthcare telemonitoring, 6LoWPAN/IEEE 802.15.4, Contiki OS, Internet of Things.

Dedicatoria

*A mis padres, que siempre me han apoyado incondicionalmente a cumplir todas mis metas; a mi hermano por su motivación y cariño.
A todos ellos, mis logros son los suyos.*

Agradecimientos

Al CICESE por abrir las puertas y permitirme recibir una formación académica de calidad proporcionándome las herramientas necesarias para mi crecimiento académico y profesional.

Al CONACYT por brindarme el apoyo económico para realizar mis estudios de maestría con número de becario 338047.

A todos los profesores del Posgrado en Electrónica y Telecomunicaciones que compartieron su conocimiento para mi formación académica. Quiero expresar mi sincero agradecimiento a mis codirectores de tesis, al Dr. Raúl Rivera Rodríguez y al M.C. Edwin Martínez Aragón por la confianza, así como su visión y orientación. Su previsión y percepción han ayudado a trazar el curso de mi investigación.

A los miembros de mi comité de tesis: Dr. Salvador Villarreal Reyes, Dr. Miguel Ángel Alonso Arévalo y Dr. Heriberto Márquez Becerra y adicionalmente al Dr. Gabriel Alejandro Galaviz Mosqueda por sus consejos, comentarios y observaciones que ayudaron a sacar adelante este trabajo de investigación.

A los integrantes del grupo ARTS que con su apoyo y recomendaciones fortalecieron este trabajo de investigación.

A Dios por brindarme todo lo necesario para poder alcanzar mis objetivos.

A mis padres y hermanos, con quienes sé que siempre podré contar donde quiera que se encuentren.

A Humberto, por acompañarme durante este recorrido, tanto en los triunfos como en los tropiezos.

A todos mis amigos, especialmente a Memo, Ramón y Jairo, por los excelentes momentos de vida tanto en lo académico como en lo personal.

Tabla de contenido

Resumen en español	ii
Resumen en inglés	iii
Dedicatoria	iv
Agradecimientos	v
Lista de figuras	viii
Lista de tablas.....	xi
Capítulo 1. Introducción	
1.1 Antecedentes.....	1
1.2 Planteamiento de problema.....	2
1.3 Objetivo general	4
1.4 Objetivos particulares.....	4
1.5 Metodología	5
1.6 Organización de la tesis	6
Capítulo 2. IPv6 para redes inalámbricas de área personal (6LoWPAN)	
2.1 Introducción	8
2.2 Redes LoWPAN	8
2.3 IP versión 6	11
2.4 IPv6 sobre LoWPAN (6LoWPAN)	12
2.4.1 Arquitectura 6LoWPAN	12
2.4.2 Pila de protocolos de 6LoWPAN.....	14
2.4.2.1 Capa PHY: IEEE 802.15.4.....	14
2.4.2.2 Capa Enlace de Datos: IEEE 802.15.4	15
2.4.2.3 Capa Adaptación	17
2.4.2.4 Capa de Red.....	21
2.4.2.5 Capa de Transporte	23
2.4.2.6 Capa de Aplicación	24
2.5 Sistema Operativo Contiki	25
2.5.1 Protocolos utilizados en Contiki OS.....	26
2.5.1.1 Capa Radio en Contiki	26
2.5.1.2 Capa RDC en Contiki	27
2.5.1.3 Capa Trama en Contiki	32
2.5.1.4 Capa MAC en Contiki.....	32
2.5.1.5 Capa de Red en Contiki	33
2.6 Simulador de redes Cooja.....	34
2.7 Resumen del capítulo	36
Capítulo 3. Transmisión de variables fisiológicas sobre 6LoWPAN	
3.1 Introducción	38
3.2 Variables fisiológicas en un escenario de telemonitoreo.....	39
3.2.1 Electrocardiograma (ECG)	39
3.2.2 Emulación del tráfico de ECG	40
3.3 Arquitectura de la red 6LoWPAN para el telemonitoreo de variables fisiológicas	42
3.3.1 Capa Física	43
3.3.2 Capa de acceso al medio	43

3.3.3	Capa de adaptación	43
3.3.4	Capa de red.....	43
3.3.5	Capa de transporte.....	44
3.4	Resumen del capítulo	45

Capítulo 4. Simulación e implementación de la WSN para la transmisión de señales fisiológicas sobre 6LoWPAN.

4.1	Introducción.....	46
4.2	Pila de protocolos (“ <i>netstack</i> ” Contiki OS)	46
4.2.1	Capa Radio.....	46
4.2.2	Capa RDC.....	46
4.2.3	Capa Trama.....	47
4.2.4	Capa MAC.....	47
4.2.5	Capa Red.....	47
4.3	Simulación de la red inalámbrica de sensores utilizando Cooja.....	48
4.4	Implementación de la red inalámbrica de sensores utilizando Contiki OS	51
4.4.1	Equipo utilizado	52
4.4.1.1	Texas Instruments CC2650STK	52
4.4.1.2	Debug DevPack XDS110 TI.....	53
4.4.1.3	Texas Instruments CC2531EMK.....	53
4.4.2	Descripción del escenario de prueba experimental.....	54
4.5	Implementación de la red inalámbrica de sensores utilizando TIMAC	58
4.5.1	Equipo utilizado	59
4.5.1.1	Nodos ECG	59
4.5.1.2	Nodo central.....	59
4.5.2	Descripción del escenario de prueba experimental.....	60

Capítulo 5. Resultados de simulación y experimentación

5.1	Introducción.....	63
5.2	Definición de métricas a evaluar.	63
5.3	Validación de la implementación.....	68
5.3.1	Pila de protocolos.....	68
5.3.2	Emulación de ECG.....	69
5.4	Evaluación del desempeño de la red 6LoWPAN.....	71
5.5	Conclusiones sobre la simulación y desarrollo de la red inalámbrica de sensores utilizando Contiki OS y TIMAC.	84

Capítulo 6. Conclusiones y trabajo futuro

6.1	Introducción.....	87
6.2	Conclusiones	87
6.2.1	Sobre la simulación de la red inalámbrica de sensores en Cooja.....	87
6.2.2	Sobre el desarrollo de la red inalámbrica de sensores utilizando Contiki.....	89
6.2.3	Sobre el desarrollo de la red inalámbrica de sensores utilizando TIMAC.	90
6.3	Trabajo a futuro	91

Literatura citada	92
--------------------------------	-----------

Lista de figuras

Figura		Página
1	Capas del Modelo de referencia OSI.....	9
2	Estructura de una red IPv6 conectada a los diferentes tipos de arquitecturas 6LoWPAN	13
3	Pila de protocolos del estándar 6LoWPAN.....	14
4	Formato PPDU en IEEE 802.15.4.....	15
5	Formato MPDU en IEEE 802.15.4.....	16
6	Proceso de fragmentación de un paquete IPv6 sobre IEEE 802.15.4.....	17
7	Formato del encabezado de IPv6 sin compresión.....	18
8	Formato del encabezado de IPV6 con compresión HC1.....	18
9	Formato del encabezado “Hop-by-Hop Options”.....	19
10	Ejemplos de compresión del encabezado IPv6 en 6LoWPAN.....	20
11	Modelo de enrutamiento IP.....	20
12	Modelo de enrutamiento “mesh-under” de paquetes en 6LOWPAN.....	21
13	Modelo de enrutamiento “route-over” de paquetes 6LoWPAN.....	22
14	Formato del encabezado de UDP.....	23
15	6LoWPAN utiliza un socket como interfaz para cada aplicación en específico.....	24
16	MPDU del estándar 6LoWPAN.....	25
17	“Netstack” de Contiki OS.....	26
18	Mecanismos capa RDC asíncronos. (a) Mecanismo asíncrono emitido por el emisor. (b) Mecanismo asíncrono emitido por el receptor.....	28
19	Proceso de transmisión para enviar un paquete en ContikiMAC.....	29
20	Transmisión de paquetes “broadcast” con la capa ContikiMAC.....	30
21	Distribución de tiempos en la capa RDC ContikiMAC (Dunkels, 2011).....	31
22	Modelo de propagación UDGm de una red de cuatro nodos en Cooja, el nodo 3 se comunica con el nodo 1 e interfiere con los nodos 2 y 4.....	35
23	Paquete 6LoWPAN sin empaquetamiento.....	41

24	Empaquetamiento de 48 muestras para un ECG de 3 canales.....	41
25	Paquete 6LoWPAN utilizando la técnica de empaquetamiento propuesta.....	42
26	Arquitectura y “netstack” de la red 6LoWPAN propuesta.....	44
27	Topología de la simulación de red de 6LoWPAN, en un escenario de telemonitoreo multiusuario.....	48
28	Modelo de pérdidas UDGM- Distance Loss en la red 6LoWPAN propuesta. (a) UDGM-Distance-Loss para nodo central (ID 1). (b) UDGM-Distance-Loss para nodo ECG (ID 2). (c) UDGM-Distance-Loss para nodo ECG (ID 3). (d) UDGM-Distance-Loss para nodo ECG (ID4). (e) UDGM-Distance-Loss para nodo ECG (ID 5).....	49
29	Escenario para la implementación del sniffer 6LoWPAN.....	54
30	Escenario experimental de la red 6LoWPAN propuesta en Contiki OS.....	55
31	Diagrama de flujo del algoritmo que ejecutan los nodos ECG del escenario propuesto.....	56
32	Diagrama de flujo del algoritmo que ejecuta el nodo central del escenario propuesto.....	57
33	Diagrama de flujo de transmisión y recepción ContikiMAC.....	58
34	Formato de paquete utilizado en la implementación de TIMAC.....	59
35	Formato de ACK en IEEE 802.15.4 utilizado en TIMAC.....	60
36	Escenario experimental de la red 6LoWPAN propuesta en TIMAC.....	60
37	Diagrama de flujo de transmisión y recepción en TIMAC. (a) Diagrama de flujo de los nodos ECG en TIMAC. (b) Diagrama de flujo del nodo central en TIMAC.....	61
38	Línea de tiempo en Cooja para nodos con $f_{RDC} = 8 \text{ Hz}$	66
39	Captura sniffer 6LoWPAN para una $f_{RDC} = 8 \text{ Hz}$	66
40	Línea de tiempo en Cooja para nodos con $f_{RDC} = 64 \text{ Hz}$	67
41	Captura sniffer 6LoWPAN para una $f_{RDC} = 64 \text{ Hz}$	67
42	Captura de paquete 6LoWPAN en Wireshark.....	68
43	Comparación del promedio de retransmisiones realizadas del mote Sky y mote Zolertia en el simulador Cooja.....	69
44	Total de paquetes generados de acuerdo al número de nodos ECG en la red 6LoWPAN.....	71

45	Total de retransmisiones realizadas conforme se incrementa el número de usuarios considerando diferentes valores f_{RDC} para (a) simulación. (b) cama de pruebas experimental.....	73
46	Total de paquetes transmitidos por nodo para un escenario con (a) un nodo ECG; (b) dos nodos ECG; (c) tres nodos ECG; (d) cuatro nodos ECG, considerando diferentes valores f_{RDC}	74
47	Promedio de retransmisiones realizadas conforme se incrementa el número de usuarios considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.....	75
48	Respuesta del promedio de retransmisiones durante la simulación. (a) Promedio de retransmisiones escenario con 1 nodo ECG. (b) Promedio de retransmisiones escenario con 2 nodos ECG. (c) Promedio de retransmisiones escenario con 3 nodos ECG. (d) Promedio de retransmisiones escenario con 4 nodos ECG.....	76
49	Respuesta del promedio de retransmisiones durante la implementación física. (a) Promedio de retransmisiones escenario con 1 nodo ECG. (b) Promedio de retransmisiones escenario con 2 nodos ECG. (c) Promedio de retransmisiones escenario con 3 nodos ECG. (d) Promedio de retransmisiones escenario con 4 nodos ECG.....	76
50	Total de paquetes transmitidos conforme se incrementa el número de nodos considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.....	78
51	Porcentaje de paquetes transmitidos con y sin éxito conforme se incrementa el número de nodos considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.....	81
52	Caudal eficaz conforme se incrementa el número de nodos considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.....	82
53	Captura sniffer 6LoWPAN de la transmisión de paquetes ECG.....	83

Lista de tablas

Tabla		Página
1	Cobertura de patrón de interferencia de los nodos ECG.....	49
2	Parámetros de la simulación.....	51
3	Total de paquetes generados en la red de acuerdo a la cantidad de nodos ECG.....	70
4	Porcentaje de paquetes transmitidos	79
5	Resumen de resultados de paquetes transmitidos con y sin éxito durante la simulación y la cama de pruebas experimental	81

Capítulo 1. Introducción

1.1 Antecedentes

Hoy en día, el enfoque de las Tecnologías de Información y Comunicación (TICs) está orientado a la interconexión a gran escala de todo tipo de dispositivos. Este enfoque surgió a partir de la necesidad del ser humano de interactuar y controlar su entorno. Para hacer esto posible, se tiene la unificación de dos vertientes: las redes inalámbricas de sensores (WSN, “*Wireless Sensor Networks*”), encargadas del monitoreo y censado de diversas variables de interés, por medio de dispositivos de recursos limitados; y el Internet de las Cosas (IoT, “*Internet of Things*”), enfocado en brindar la capacidad de conectar cada dispositivo a Internet. En este sentido, una de las áreas de mayor importancia en el IoT es el cuidado de la salud, la cual se enfoca en la prestación de servicios de medicina a distancia, utilizando dispositivos móviles, conocida también como “*m-Health*” para IoT (m-IoT). De esta manera, es posible ofrecer una nueva y creciente gama de servicios de atención en salud, utilizando diferentes sensores y tecnologías para el monitoreo remoto del estado de un paciente, conocido como telemonitoreo de salud.

En este contexto, actualmente existen diversas tecnologías inalámbricas que se han utilizado para el monitoreo de variables fisiológicas, como Bluetooth, ZigBee, Z-wave. Sin embargo, para habilitar soluciones basadas en este tipo de redes, es necesario el diseño e implementación de “*gateways*”¹, con el consecuente retardo adicional, para poder conectarse a redes basadas en IP, haciendo que su implementación sea más compleja. Por tal motivo, se han propuesto diversos sistemas de telemonitoreo que utilizan redes híbridas², tal y como lo muestra Laine et al. (2014), presentando un sistema de telemonitoreo que cuenta con un dispositivo “*gateway*” que permite la comunicación entre una WSN y un servidor remoto utilizando ZigBee, Bluetooth y WLAN/3G, permitiendo conectividad IP a los nodos sensores que componen la red.

No obstante, la principal desventaja de utilizar un “*gateway*” para conectar una WSN hacia Internet, es que incrementa la cantidad de procesamiento debido a la conversión de la pila de protocolos de una red a otra. Además este proceso también agrega un retardo, el cual dependiendo de la aplicación, puede ser altamente perjudicial para el desempeño de la misma. Tal es el caso de aplicaciones relacionadas con el

¹ Dispositivo que permite la traducción de los datos recibidos de la pila de protocolos de una red hacia otra.

² Una red híbrida se refiere a una arquitectura de red que transmite información utilizando tecnologías de comunicación diferentes, separadas física y lógicamente.

cuidado de la salud, las cuales pueden ser sensibles al retardo y/ a la pérdida de paquetes, por lo que se debe tener un cuidado especial en el diseño de la red para el transporte de variables médicas.

Considerando lo anteriormente mencionado, el grupo de trabajo 6lo de la IETF (*“Internet Engineering Task Force”*) propuso un estándar abierto que provee una pila de protocolos enfocados a la adaptación entre redes basadas en IPv6 y diversas tecnologías de comunicación (p. ej. IEEE 802.15.4), conocido como, 6LoWPAN (*“IPv6 over Low-Power Wireless Personal Area Networks”*). En particular se ha propuesto utilizar esta pila de protocolos para permitir a redes enfocadas a diferentes aplicaciones, entre ellas, el cuidado de la salud, extender el rango de alcance de los servicios y proporcionando movilidad, flexibilidad e interoperabilidad con las redes basadas en IP.

En este contexto es importante mencionar que, en la actualidad la tecnología IEEE 802.15.4 es la más relevante para la implementación de la capa PHY y MAC del estándar 6LoWPAN. Esto se debe principalmente a que IEEE 802.15.4 fue la primera tecnología que adoptó el grupo de trabajo 6lo para transportar paquetes IPv6 sobre WPANs³. Adicionalmente, 6LoWPAN sobre IEEE 802.15.4 soporta la formación de redes de múltiples saltos en la capa de enlace, a diferencia de otras tecnologías (p. ej. Bluetooth), lo cual permite tener una red 6LoWPAN extendida.

1.2 Planteamiento de problema

En los últimos años los sistemas m-Health basados en IoT (conocidos como m-IoT) han atraído la atención de varios investigadores debido a el potencial, los beneficios y los desafíos que se presentan este tipo de sistemas. Esto se debe principalmente a las ventajas que proporciona el que los profesionales de la salud dispongan de información inmediata, en diferentes escenarios del cuidado de la salud, ya sea dentro o fuera de un hospital, proveniente de un monitoreo constante, de los signos vitales o parámetros fisiológicos específicos de un paciente.

Adicionalmente, la transmisión de información referente a variables fisiológicas no es una tarea trivial, debido a que los datos que se transportan pueden ser críticos y con requerimientos especiales (p. ej. ECG), donde se requiere de una transmisión continua (*“streaming”*) para poder reconstruir la señal, de tal manera que a partir de ella se pueda realizar un diagnóstico al paciente. Es por ésta razón, que es necesario considerar las características del tráfico que se manejará en la implementación de un escenario de m-IoT.

³ <https://tools.ietf.org/wg/6lowpan/draft-ietf-6lowpan-format/>

En este sentido, la mayoría de los trabajos realizados al momento que utilizan redes 6LoWPAN para aplicaciones enfocadas al cuidado de la salud corresponden a simulaciones que utilizan software como OMNet ++ (Touati, Mnaouer, et al., 2015) y Cooja (Touati, Tabish, & Ben Mnaouer, 2013)(Mehmood et al., 2014); e implementaciones en Contiki OS uno de los sistemas operativos abiertos orientados a redes de sensores inalámbricos, siendo uno de los más importantes y más utilizados. Cabe mencionar, que TinyOS es un sistema operativo que también es utilizado para implementaciones del estándar 6LoWPAN. Sin embargo, no incluye la pila de protocolos correspondiente al estándar IEEE802.15.4 (Matús, 2007).

Dentro de los reportes de modelos y soluciones para aplicaciones del cuidado de la salud se tienen a (Touati, Mnaouer, et al., 2015), en donde se realiza una evaluación del rendimiento de una arquitectura de red, que integra al estándar 6LoWPAN y un sistema de telemonitoreo. La evaluación es realizada mediante simulaciones utilizando OMNet++ y posteriormente comparada con un estudio experimental desarrollado en Contiki OS, utilizando motes Zolertia.

Otros trabajos relacionados que proponen el uso del estándar 6LoWPAN en aplicaciones de telemonitoreo son introducidos por Touati et al. (2013) y por Mehmood et al. (2014). Touati et al. (2013) utilizaron la pila de protocolos que proporciona el sistema operativo de Contiki OS. Por otro lado, Mehmood et al. (2014), presentaron la evaluación del rango de escalabilidad de una simulación de una red 6LoWPAN utilizando Cooja, simulador de redes de Contiki OS. Cada nodo está compuesto por seis sensores biomédicos (ECG, acelerómetro, temperatura, ritmo cardíaco, glucosa y presión sanguínea). Es importante mencionar que en estos trabajos no se estudian las características de tráfico de las variables fisiológicas que se transmiten ni tampoco se especifica los parámetros que se utilizan dentro de Contiki OS.

Si bien se han hecho evaluaciones de sistemas de telemonitoreo utilizando el estándar 6LoWPAN, los resultados de simulación y experimentación disponibles en la literatura difieren de forma significativa. Por ejemplo, en (Touati, Mnaouer, et al., 2015), el caudal eficaz máximo en la simulación es de 70 Kbps mientras que para la simulación es de poco más de 20 Kbps. Esta diferencia puede ser asociada a que el “*testbed*” implementado en el simulador OMNet++ no tiene las mismas características que el utilizado para los experimentos físicos, en cuanto a la pila de protocolos que se manejan en cada uno. Esto puede ocasionar que las conclusiones obtenidas durante la simulación no sean completamente aplicables a un escenario más realista, es decir, donde se consideren motes físicos con la implementación de la misma pila de protocolos (6LoWPAN), pero con características diferentes.

Así pues, en este trabajo de tesis se evaluará la implementación del estándar 6LoWPAN con un enfoque hacia la implementación de una aplicación de telemonitoreo médico, de manera que se consideren las

características del sistema operativo utilizado para la implementación del estándar 6LoWPAN, así como también las especificaciones del tipo de tráfico que se manejan.

Esto permitirá tener un estudio más completo de la implementación de una red 6LoWPAN con el fin de evaluar su desempeño y comportamiento cuando se utiliza en aplicaciones y sistemas de monitoreo de m-IoT, con especial énfasis en el número de usuarios concurrencios que pueden ser servidos. La evaluación de esta red se llevará a cabo en una de los sistemas operativos más utilizados, Contiki OS, ya que presenta características atractivas en cuanto a que es de código abierto, soporte de varias plataformas y dispone la mayoría de los protocolos de comunicación que establece el estándar 6LoWPAN.

1.3 Objetivo general

El objetivo de este proyecto de investigación es evaluar el desempeño del estándar 6LoWPAN implementado sobre la tecnología IEEE 802.15.4 para la transmisión de señales fisiológicas, en un escenario multiusuario, considerando la sobrecarga introducida por protocolos de capas superiores a la física y de acceso al medio. Esto se realizará utilizando una de las herramientas más aceptadas para la implementación del estándar 6LoWPAN, tomando en cuenta sus características específicas en cuanto a la pila de protocolos que maneja para definir el estándar de 6LoWPAN.

1.4 Objetivos particulares

- Estudiar las características de las diferentes variables fisiológicas utilizadas en aplicaciones de telemonitoreo, con el fin de definir el perfil de tráfico que se manejará en la investigación.
- Estudiar la pila de protocolos que ofrece el sistema operativo Contiki OS, en específico para el transporte de manera eficiente de paquetes IPv6 sobre IEEE 802.15.4. Esto con la finalidad de tener conocimiento del funcionamiento, características y deficiencias del mismo.
- Realizar la evaluación del escenario de telemonitoreo propuesto a través de simulación en términos del número paquetes recibidos /perdidos y de usuarios que pueden ser atendidos, con base a la implementación de 6LoWPAN en Contiki OS.
- Realizar la evaluación experimental del escenario de telemonitoreo propuesto en términos del número paquetes recibidos /perdidos y usuarios que pueden ser atendidos haciendo uso de un

sniffer 6LoWPAN y la recopilación de datos por parte de los nodos que componen la red. Con base a la implementación de 6LoWPAN en Contiki OS.

- A partir de las evaluaciones realizadas, determinar si es factible utilizar al estándar 6LoWPAN en Contiki OS para aplicaciones de m-Health.
- Verificar si existe otra opción para realizar la implementación de estándar 6LoWPAN, de tal manera que se determine si es factible utilizarse para aplicaciones de m-Health.

1.5 Metodología

1. Definir el perfil de tráfico de la información a transmitir.

Debido a que se pretende que esta investigación pueda ser utilizada en el establecimiento de un escenario con una red de sensores orientados hacia el telemonitoreo, se definieron las características principales de la información a transmitir, en términos del perfil de tráfico.

2. Estudio bibliográfico de 6LoWPAN

En esta parte se realizó un estudio de la pila de protocolos que define el estándar 6LoWPAN, de tal manera que se tuviera conocimiento de las características y parámetros de cada capa. La finalidad es contar con los elementos suficientes para realizar posteriormente determinar las características en la simulación e implementación de una red 6LoWPAN.

3. Selección de sistema operativo y familiarización con la programación del mismo

Se seleccionó el sistema operativo adecuado para la implementación del estándar 6LoWPAN. Luego de ello, se realizó un estudio de las bibliotecas, macros y procesos, debido a que no se contaba con un previo conocimiento acerca de la programación dentro del sistema operativo seleccionado.

4. Definición de la arquitectura y escenarios de evaluación de la red 6LoWPAN

Se definió la topología y escenario en el cual se evaluará a manera de simulación e implementación la red 6LoWPAN. Además también se especificaron los parámetros y consideraciones que se realizan en ambos casos.

5. Simulación e implementación de la red 6LoWPAN utilizando ContikiOS

De acuerdo a los escenarios y parámetros propuestos, se simuló la red 6LoWPAN utilizando Cooja, en Contiki OS. Por otra parte, para la implementación de la misma, se programó cada uno de los motes, utilizando Contiki OS y Flash Programmer 2.

6. Prueba y evaluación de la red 6LoWPAN

Se realizaron las pruebas pertinentes para observar el funcionamiento de la red 6LoWPAN en escenarios de telemonitoreo. Se obtuvieron métricas que sirven para como base para evaluar la red 6LoWPAN propuesta.

1.6 Organización de la tesis

En el Capítulo 2 se describe las características del estándar 6LoWPAN. Así como también la pila de protocolos que ofrece Contiki OS, ya que será considerada para la simulación e implementación de la red 6LoWPAN. Además se muestra un estudio de las redes inalámbricas de sensores enfocadas en un sistema de telemonitoreo. De esta manera se encontraron las señales biomédicas más comunes en escenarios de telemonitoreo y a partir de ello se revisó la literatura para determinar la frecuencia de muestreo, caudal eficaz requerido y sensibilidad al retardo de las diferentes señales.

En el Capítulo 3 se presentan un resumen de las principales señales biomédicas que se encuentran en un escenario de telemonitoreo. Así mismo se define el perfil de tráfico y la arquitectura que se propone para habilitar la red que se utilizará para evaluar la implementación del estándar 6LoWPAN en un escenario de telemonitoreo.

En el Capítulo 4 se muestran la descripción y parámetros del ambiente de simulación e implementación red 6LoWPAN en un escenario de telemonitoreo. Así como también se especifica el equipo utilizado.

En el Capítulo 5 se presenta la discusión de los resultados obtenidos de la evaluación del desempeño de la misma, dentro de diferentes escenarios, variando el número de nodos en la red y la frecuencia de encendido del radio. Además, se propone utilizar la pila de protocolos que ofrece TIMAC, de Texas Instruments, para dar solución a los inconvenientes que surgieron al utilizar la pila de protocolos ofrecida por Contiki OS. Asimismo se muestra una perspectiva del alcance que se podría tener utilizando esta solución.

Para finalizar, en el capítulo 6 se muestran las conclusiones y trabajo a futuro que surgen como fruto de este trabajo de investigación.

Capítulo 2. IPv6 para redes inalámbricas de área personal (6LoWPAN)

2.1 Introducción

En este capítulo se detallarán las funciones más relevantes para esta tesis de la pila de protocolos del estándar 6LoWPAN. Adicionalmente, se dará un resumen de los aspectos más importantes de la implementación del estándar 6LoWPAN sobre Contiki OS y Cooja que serán utilizado para evaluar el estándar.

2.2 Redes LoWPAN

Las redes inalámbricas de sensores (WSN por sus siglas en inglés) son una parte esencial en las aplicaciones de m-IoT. Específicamente, las WSN de área personal o redes inalámbricas de área personal (WPANs, “*Wireless Personal Area Networks*”) deben ser capaces de recolectar y transmitir a un resumidero de información las variables fisiológicas. Esto de forma eficiente en términos de energía y ancho de banda considerando dispositivos con recursos computacionales restringidos. Las WPANs, especialmente para aplicaciones en el contexto de m-IoT, deben ser diseñadas para permitir la escalabilidad y la funcionalidad para proporcionar su integración con la arquitectura actual de Internet.

Las redes LoWPAN (“*Low power Wireless Personal Area Networks*”) se enfocan en la transmisión de información dentro de rangos de comunicación pequeños (p. ej. IEEE 802.15.4, maneja rangos de transmisión menores a 100 m) entre un grupo limitado de dispositivos participantes. Este tipo de comunicación involucra tasas de transmisión medianas y bajas, poca o nula infraestructura o conectividad directa con redes fuera del área de cobertura de la LoWPAN (Casillas, 2012). Esto permite reducción en costos de implementación y un bajo consumo de energía. Por lo que, diversas tecnologías de LoWPANs como por ejemplo Bluetooth Low Energy, IEEE 802.15.4, 6LoWPAN han sido propuestas para habilitar aplicaciones de telemonitoreo (Casillas, 2012) (Dodd, 2015) (Touati, et al. 2015)(Touati, et al. 2015), monitoreo de edificios, automatización, domótica, por mencionar algunos (Pham-Huu, Nguyen, Trinh, Bui, & Pham, 2015)(Nedelcu, Sandu, Machedon-Pisu, Alexandru, & Ogrutan, 2009).

Sin embargo, habilitar la transmisión de datos a través de Internet utilizando una tecnología LoWPAN no es una tarea trivial. Para abordar esta tarea compleja, los sistemas de comunicaciones tienen un modelo para la transmisión de información basado en capas. Esto permite diseñar protocolos de comunicación que rigen el proceso de intercambio de información entre capas iguales de los dispositivos conectados en una red.

El modelo de arquitectura para la interconexión de redes es el modelo de referencia OSI (“*Open Systems Interconnection*”(International Organization for Standardization (ISO), 1996)). En la Figura 1 se muestra la estructura del modelo OSI, en la cual se puede visualizar de manera detallada como viaja la información a través de un enlace de comunicaciones.

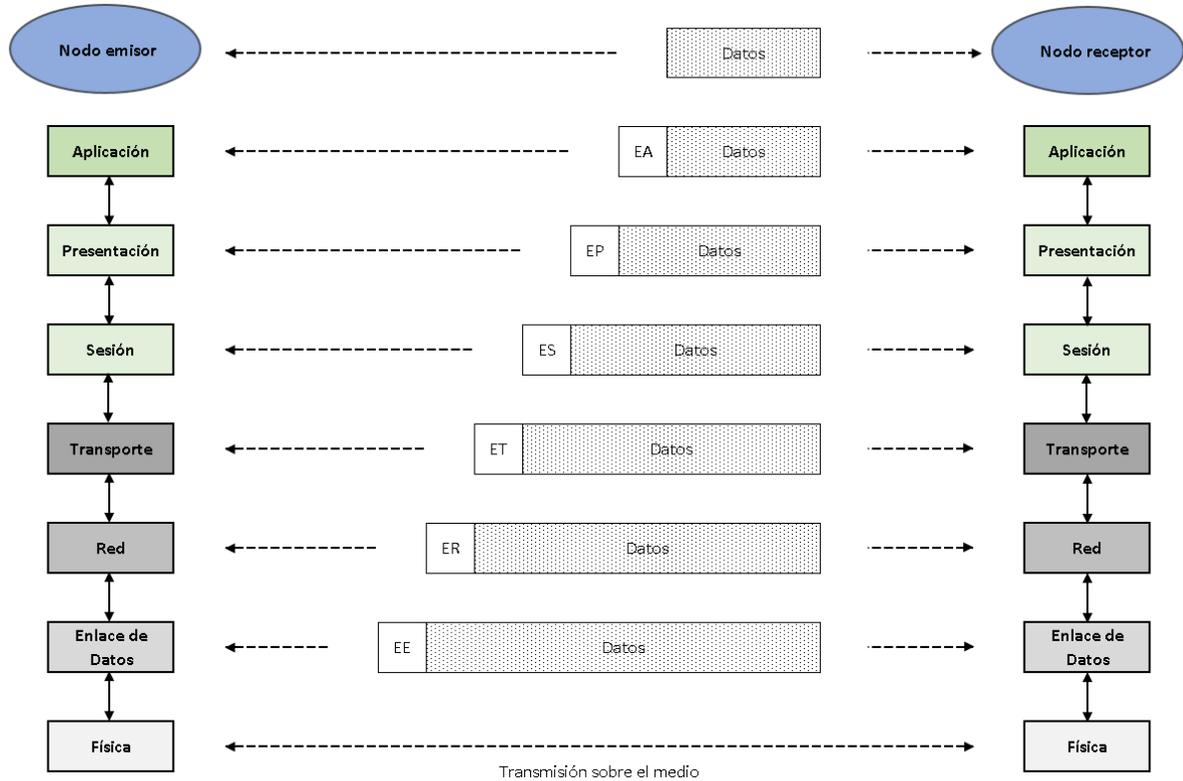


Figura 1. Capas del Modelo de referencia OSI.

Hoy en día existen varios organismos encargados de la definición de los estándares que especifican el funcionamiento y características de una o algunas capas del modelo OSI que precisan una tecnología en particular.

Uno de los organismos de estandarización más importantes es el IEEE (“*Institute of Electrical and Electronics Engineers*”). Específicamente, el grupo de trabajo IEEE 802.15 se divide en grupos especializados para satisfacer diferentes necesidades de comunicación de las capas PHY y MAC dentro de una WPAN, uno de los más importantes para este trabajo de investigación es el grupo 4, WPAN de baja tasa de datos.

Por otro lado, existen propuestas para la adaptación y/o integración de los protocolos de comunicaciones ofrecidos por las tecnologías LoWPANs a las redes IP actuales. Esto como alternativas a propuestas de utilizar protocolos totalmente específicos para cada LoWPAN y adaptarlos según las necesidades de la

aplicación. En este contexto existe un organismo denominado IEFT (*“Internet Engineering Task Force”*) que se encarga de desarrollar y promover estándares de Internet, y cuenta con el grupo de trabajo 6Lo, que ofrece una solución que tiene particular relevancia en esta investigación.

La primera versión del grupo de trabajo 6Lo, denominada 6LoWPAN, fue constituida en el año 2005. 6LoWPAN está enfocado a implementar y diseñar métodos para proveer conectividad IPv6 sobre redes de nodos con capacidades limitadas (energía, memoria, recursos de procesamiento, ancho de banda, entre otros). Particularmente trabajaba definiendo métodos para adaptar IPv6 en redes WPAN que utilizan el estándar IEEE 802.15.4 (6LoWPAN). Posteriormente en 2014 fue reemplazado por el grupo 6Lo que utiliza un mecanismo similar de adaptación para una gama más amplia de tecnologías como Bluetooth Low Energy (BLE), Z-Wave, por mencionar algunos.

Para la implementación del estándar 6LoWPAN es necesario contar con un sistema operativo que realice las funciones de gestión de memoria y acceso a los recursos de hardware por las distintas tareas que se ejecutan en la pila de protocolos del estándar. Actualmente se encuentran disponibles diversas soluciones diseñadas específicamente para dispositivos con recursos de cómputo limitados. Dos de los sistemas operativos más comunes son Contiki y TinyOS.

En particular en este trabajo se propone utilizar Contiki OS debido a que ofrece soporte para varios protocolos de Internet (p. ej. IPv6, UDP/TCP, HTTP) y LoWPAN (IEEE 802.15.4 PHY y MAC). Esto permite la implementación del estándar 6LoWPAN sobre varias plataformas. Además, Contiki OS cuenta con un simulador de redes llamado Cooja, que permite visualizar como se ejecutan los diseños de las redes a gran escala, como se detalla en la sección 2.5 de este capítulo. Es importante mencionar debido a que se utilizarán motes *“Texas Instruments”*, para la implementación del estándar, éstos recomiendan utilizar a Contiki OS como sistema operativo para dar soporte 6LoWPAN⁴. Aunado a ello Contiki OS y Cooja son de las implementaciones 6LoWPAN más utilizadas.

Es por ello que es importante revisar las capacidades de la tecnología 6LoWPAN con la finalidad de estudiar su utilización en un escenario de telemonitoreo multiusuario, así como también describir las herramientas que se utilizarán para evaluar el estándar. Por lo tanto, en este capítulo se describe las características más importantes de IPv6, 6LoWPAN, Contiki OS y Cooja.

⁴ <http://processors.wiki.ti.com/index.php/Contiki-6LoWPAN>

2.3 IP versión 6

Es conocido como el Protocolo de Internet versión 6 es la nueva versión del protocolo IP (*“Internet Protocol”*, por sus siglas en inglés), para reemplazar en forma gradual a la versión actual, IPv4. El motivo básico para crear este protocolo es la falta de direcciones. Por ello IPv6 define direcciones con una longitud mayor (direcciones de 128 bits) a la longitud de las direcciones proporcionadas por IPv4 (direcciones de 32 bits), esto significa que se tienen 2^{128} direcciones posibles. De esta manera se podrán soportar más niveles en la jerarquía de direccionamiento y un número mucho mayor de nodos bidireccionales.

El soporte de IPv6 en las redes LoWPAN no solamente se justifica por una comunicación fluida con las redes convencionales IP, sino también por el apoyo de los protocolos ya conocidos que se definen en las capas de alto nivel, tales como, TCP, UDP, HTTP, entre otros.

Los principales beneficios de utilizar IPv6 y protocolos de Internet para aplicaciones implementadas en redes LoWPAN son:

- Los dispositivos basados en IP pueden ser conectados fácilmente a otra red IP sin la necesidad de una translación realizada por un gateway.
- Las redes IP permiten el uso de la infraestructura ya existente.
- Las tecnologías basadas en IP han existido por décadas, son bien conocidas, y han sido trabajadas y escaladas.
- La tecnología IP está especificada de manera libre y abierta, con estándares y documentos disponibles para el público en general.
- Pueden utilizar las herramientas para gestión y diagnóstico de las redes basadas en IP que existen hoy en día.

Una de las soluciones que ofrece este tipo de ventajas es la proporcionada por el estándar 6LoWPAN, en la cual se pueden integrar diferentes tecnologías LoWPANs con el protocolo IPv6. Enseguida se realiza una descripción de este estándar de redes basadas en IP de bajo consumo.

2.4 IPv6 sobre LoWPAN (6LoWPAN)

6LoWPAN (“IPv6 over Low power Wireless Personal Area Networks”) es un estándar definido por la organización IETF. Este estándar permite el uso eficiente de IPv6 sobre redes inalámbricas de bajo consumo y bajas tasas de transmisión (LoWPANs) embebidas en dispositivos simples, a través de una capa de adaptación y la optimización de protocolos relacionados (Shelby & Bormann, 2011).

Como se mencionó anteriormente, originalmente 6LoWPAN fue definido para especificar el transporte de paquetes IPv6 sobre IEEE 802.15.4. Sin embargo actualmente ha sido adaptado para ser utilizado sobre varias tecnologías como Bluetooth, Sub-1 GHz low-power RF, por mencionar algunas (Olsson, 2014). Esto permite a 6LoWPAN ser más versátil en su implementación y así elegir la tecnología que más se adapte a las necesidades de la aplicación que se quiere desarrollar.

2.4.1 Arquitectura 6LoWPAN

Las redes 6LoWPAN son WPANs, en forma de malla habitualmente, capaces de comunicarse con redes IP externas. Existen tres tipos de arquitectura 6LoWPAN (ver Figura 2):

- **Ad-Hoc.** Se caracteriza por ser sin infraestructura y estar conectada a Internet. En esta topología, un router de la red ad-hoc se configura para actuar como un nodo frontera (“*Edge Router*”). Una LoWPAN Ad-Hoc cuenta con un prefijo IPv6 local en lugar de uno global. Esto impide que se tengan rutas fuera de la LoWPAN.
- **Simple.** Están conectadas mediante un “*Edge Router*” a otra red IP. A partir de múltiples LoWPANs simple se puede construir la estructura de grandes subredes.
- **Extendida.** Consiste en múltiples “*Edge Routers*” que comparten el mismo prefijo IPv6 a lo largo de un enlace de tipo “*backbone*” (p. ej. Ethernet). Tanto como la LoWPAN simple y la LoWPAN extendida, pueden comunicarse con cualquier otro servidor mediante sus “*Edge Routers*”.

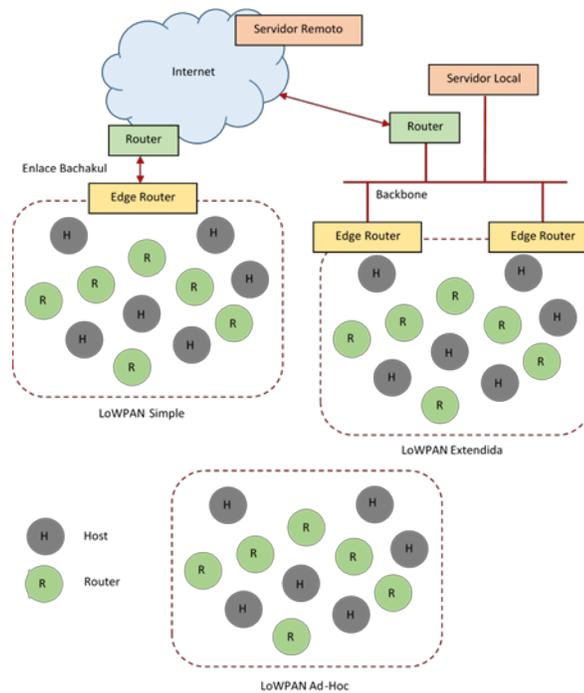


Figura 2. Estructura de una red IPv6 conectada a los diferentes tipos de arquitecturas 6LoWPAN.

A demás como se observa en la Figura 2 una red 6LoWPAN cuenta con tres tipos de dispositivos principalmente:

Edge router. Es un dispositivo frontera que se encuentra conectado a una red IPv6 y a una red LowPAN (ver **Figura 2**). Se encarga de intercambio de datos entre: los dispositivos 6LoWPAN e Internet, dispositivos dentro de la red 6LoWPAN y la generación y mantenimiento de la red 6LOWPAN.

Router. Como su nombre lo indica, son los encargados de direccionar los datos desde su destino hacia otro nodo en la red 6LoWPAN.

Host. Son dispositivos finales (“end-devices”) y no son capaces de efectuar tareas de enrutamiento de datos hacia otros dispositivos en red. Un host puede ser un dispositivo en reposo y encender periódicamente para revisar si su padre (en este caso el router) tiene datos para enviarle, esto permite tener un bajo consumo de energía.

2.4.2 Pila de protocolos de 6LoWPAN

La pila de protocolos de 6LoWPAN, está conformado por seis capas, en las que se definen diferentes protocolos ya existentes que han sido adaptados para dispositivos de bajo consumo (ver Figura 3). Enseguida se describe cada una de las capas, así como también sus características y tareas principales.

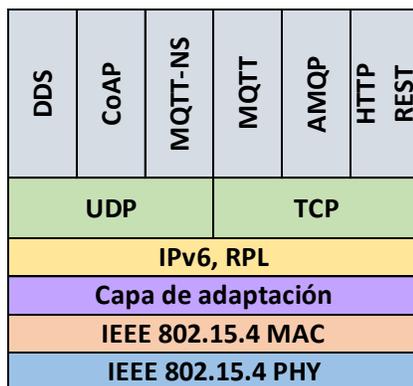


Figura 3. Pila de protocolos del estándar 6LoWPAN.

2.4.2.1 Capa PHY: IEEE 802.15.4

La capa física de 6LoWPAN está especificada para diversas tecnologías, sin embargo el estándar IEEE 802.15.4 es la más comúnmente utilizada. El estándar IEEE 802.15.4 utiliza la técnica de espectro ensanchado por secuencia directa (DSSS) con la finalidad de eliminar o reducir los efectos de la interferencia de otras redes y tener mayor robustez frente al ruido. El radio, que opera a una frecuencia de 2.400-2.4835 GHz, ofrece una tasa de transmisión de 250 Kbps. Los servicios que provee esta capa son los siguientes:

- 1) **Servicio de transmisión de datos de la capa PHY (“PHY Data Service”).** Provee la transmisión y recepción de paquetes de datos entre la capa MAC y PHY a través del canal radio.
- 2) **Servicio de gestión de la capa PHY (“PHY Management Service”).** Evalúa que el canal se encuentre libre (CCA, “Clear Channel Assessment”), detecta la presencia de energía en el canal, asigna y recupera parámetros así como también habilita y deshabilita el transmisor.

Formato PPDU

La estructura de la unidad de datos del protocolo PHY (PPDU, *“PHY Protocol Data Unit”*), se muestra en la Figura 4, como se puede observar cuenta con un campo de sincronización de encabezado (SHR, *“Synchronization Header”*), que permite al receptor la sincronización con el flujo de bits. El encabezado SHR está compuesto de dos campos: un preámbulo, que es utilizado por el transceptor para lograr la sincronía con los chips/símbolos y el delimitador de inicio de trama (SFD, *“Start of Frame Delimiter”*), el cual indica el final del SHR y el inicio del paquete de datos. Enseguida se tiene un encabezado PHY (PHR, *“PHY HEader”*), el cual especifica el número de octetos que contendrá la trama. Y por último se define el campo de carga útil de la capa PHY con una longitud variable, llamada unidad de datos para servicio PHY (PSDU, *“PHY Service Data Unit”*) con un tamaño máximo de 127 bytes.

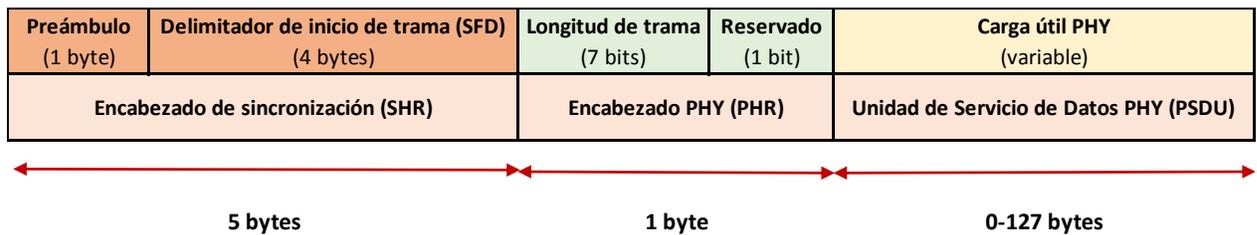


Figura 4. Formato PPDU en IEEE 802.15.4

2.4.2.2 Capa Enlace de Datos: IEEE 802.15.4

Esta capa está definida por la subcapa MAC del estándar IEEE 802.15.4 y ofrece un enlace confiable, detectando corrigiendo errores que pudieron ocurrir durante la transmisión y recepción en la capa física. Para ello se emplean una técnica de detección de errores FCS (*“Frame Check Sequence Redundancy Check”*) y un chequeo de redundancia cíclica (CRC, *“Cyclic Redundancy Check”*), utiliza un polinomio de orden 16 propuesto por la ITU-T. Además incluye una capa de control de acceso al medio (MAC, *“Media Access Control”*), que utiliza la técnica de acceso múltiple por sentido de portadora con evasión de colisiones (CSMA-CA, *“Carrier Sense Multiple Access- Collision Avoidance”*), donde el radio escucha que el canal se encuentre libre antes de transmitir sus datos. Los servicios que brinda esta capa son los siguientes:

- 1) **Servicio de datos MAC (*“MAC Data Service”*)**. Denominado como parte común de la capa MAC (MCPS, *“MAC Common Part Layer”*), el cual permite la transmisión y recepción de la unidad de datos del protocolo MAC (MPDU, *“MAC Protocol Data Unit”*) a través del servicio de transmisión de datos de la capa PHY.

2) **Servicio de gestión de la capa MAC (“MAC Management Service”).** También llamado entidad de gestión de la capa MAC (MLME, “MAC Layer Management Entity”), facilita las interfaces de servicio a través de la cual se pueden lograr las funciones de gestión de capa, tales como la asociación/desasociación, notificaciones de nodo, escaneo/inicio de red, sincronización/búsqueda de red, entre otras.

Formato MPDU

El formato general de la unidad de datos del protocolo MAC (MPDU, “MAC Protocol Data Unit”), se muestra en la Figura 5. Está encapsulado dentro en el PSDU de la capa física y está compuesto por los siguientes campos: un encabezado MAC (MHR, “MAC Header”), la unidad de servicio de datos MAC (MSDU, “MAC Service Data Unit”) y un pie de MAC (MFR, “MAC Footer”).

El primer campo del MHR es el control de trama e indica el tipo de trama MAC que se está transmitiendo, el formato del campo de dirección, y controla los acuses de recibo (ACK). Enseguida se tiene el campo de número de secuencia, y el campo de direcciones, donde se especifican las direcciones fuente y destino, el tamaño de los campos de direcciones puede variar desde 0, para el caso en que se está enviando un paquete ACK, hasta 20 bytes, ya que es posible utilizar direcciones cortas y largas.

La MSDU es la carga útil MAC y contiene la información a transmitir, es de longitud variable, sin embargo se debe cuidar no exceder la longitud máxima del PSDU. Por último se tiene el MFR, encargado de verificar la integridad de la trama MAC, por medio de un código de redundancia cíclica (CRC, “Cyclic Redundancy Code”).

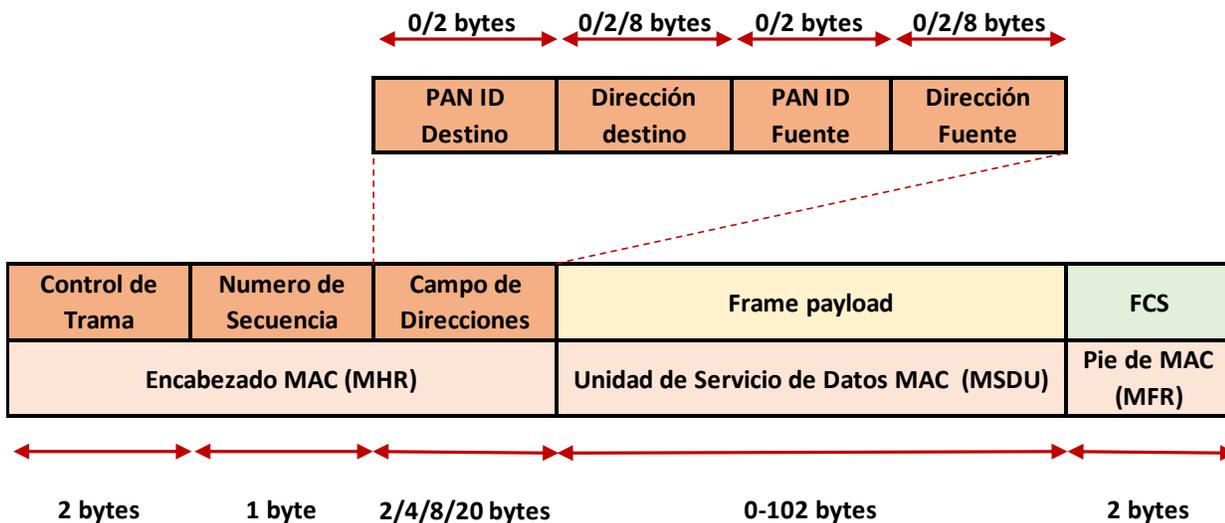


Figura 5. Formato MPDU en IEEE 802.15.4.

2.4.2.3 Capa Adaptación

Uno de los objetivos principales de 6LoWPAN, es habilitar la transmisión de información entre dispositivos basados en IP, sin la necesidad de “gateways” que traduzcan los protocolos utilizados de una red inicial a una red destino.

Es por ello que el estándar 6LoWPAN introduce dentro de su pila de protocolos una capa de adaptación ubicada entre la capa de red y la capa de enlace de datos para permitir la transmisión de los datagramas IPv6 sobre los enlaces de radio IEEE 802.15.4.

La capa de adaptación tiene tres tareas primarias:

- 1) **Fragmentación y reensamblado.** El estándar de IPv6 define 1280 bytes como longitud mínima de la unidad máxima de transmisión (MTU, “*Maximum Transmission Unit*”), sin embargo, este tamaño de trama no puede ser transportado dentro la MSDU de IEEE 802.15.4, ya que solo se tienen 127 bytes disponibles. Debido a este problema, un paquete IPv6 que se quiere transmitir sobre IEEE 802.15.4 tiene que ser dividido en varios segmentos. Por lo tanto la capa de adaptación debe encargarse de fragmentar y reensamblar el paquete según sea el caso, para posteriormente ser encapsulado dentro de la MSDU dentro del formato MPDU, como se ilustra en la Figura 6. Es importante mencionar que en caso de que sea necesaria la fragmentación se agrega un encabezado de 5 bytes en cada segmento para poder realizar una reconstrucción exacta del paquete original en el destino.

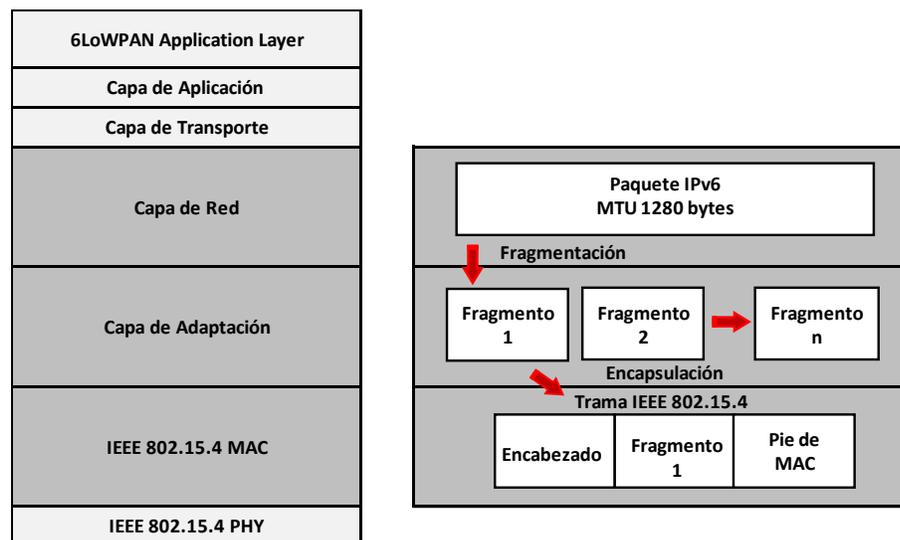


Figura 6. Proceso de fragmentación de un paquete IPv6 sobre IEEE 802.15.4.

2) **Compresión de encabezado.** Consiste en comprimir el encabezado de IPv6 y de esta manera reducir la sobrecarga que aporta al paquete que se está transmitiendo. Por lo tanto una vez que el paquete ha sido fragmentado y transmitido sobre tramas IEEE 802.15.4, cada fragmento contiene parte del paquete original de IPv6. Sin la compresión, no sería posible transmitir el paquete con efectividad. Lo anterior se puede apreciar en la Figura 7, donde se tiene el encabezado de IPV6 sin compresión, con una longitud de 40 bytes; mientras que en la Figura 8 se muestra el mismo encabezado pero está vez comprimido por la capa de adaptación de acuerdo a la norma HC1 (*“Header Compressor 1”*) y HC2 (*“Header Compressor 2”*), éste último encargado de aportar las opciones de compresión del encabezado UDP. De esta manera es posible reducir el encabezado a un tamaño de 3 bytes.

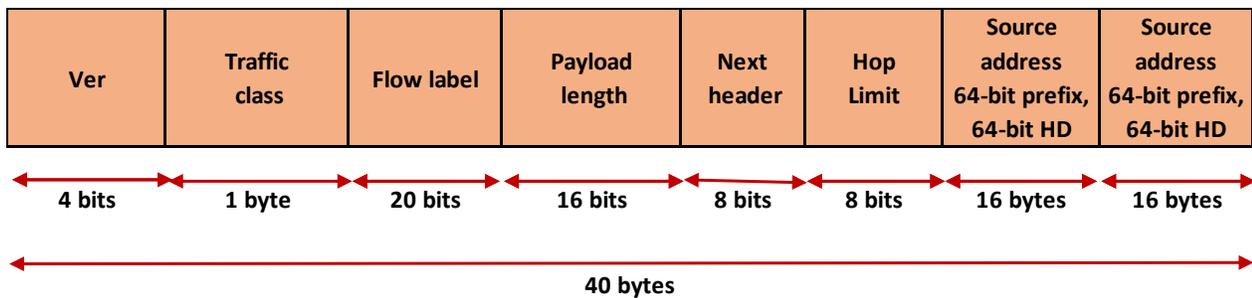


Figura 7. Formato del encabezado de IPv6 sin compresión.

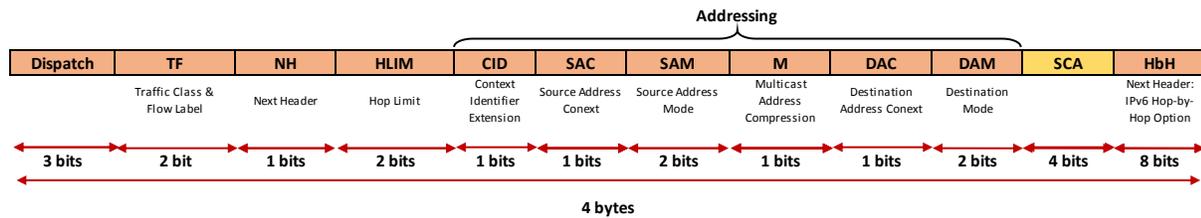
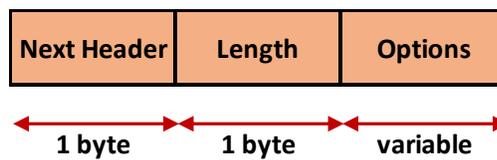


Figura 8. Formato del encabezado de IPv6 con compresión HC1.

Como se puede observar, el encabezado IPv6 comprimido (IPHC por sus siglas en inglés) elimina todos los campos que se repiten o que son fijos. En este sentido las direcciones de origen y destino pueden ser definidas por medio del contexto del encabezado especificado por la capa MAC de IEEE 802.15.4 es por ello que en el IPHC únicamente se utilizan 2 bytes, dentro de ellos también se especifica el tipo de comunicación que se establece con el destino del mensaje (*“unicast”* o *“multicast”*) en los campos M (*“Multicast Address Compression”*), DAC (*“Destination address compression”*) y DAM (*“Destination address mode”*). A demás en esta compresión se puede activar el uso del encabezado *“Hop-by-Hop Options”*, encargado de llevar la información opcional que debe ser examinada por cada nodo a lo largo del trayecto de entrega de un paquete (ver Figura

9). Está compuesto por un byte, en donde se especifica el tipo de encabezado que se encuentra enseguida de sí mismo, la longitud que posee y un campo de opciones de longitud variable para definir aspectos relacionados con el protocolo de enrutamiento que se estará utilizando.

Por otro lado se omiten campos como *versión*, ya que es un campo fijo debido a que al utilizar 6LoWPAN se estará utilizando la versión 6 del protocolo de Internet. Los campos que no pueden ser omitidos o especificados por el encabezado de otras capas de la pila de protocolos, quedan a disposición de la norma HC1, tal es el caso del campo *Next Header*, *Traffic class*, *Flow Label* y *Hop Limit*.



(a) Compresión de encabezado, FE80::CAFE:00FF:FE00:0100 → FE80::CAFE:00FF:FE00:0200

Ver	Compr. Header
-----	---------------

(b) Compresión de encabezado, 2001::DEC4:E3A1:FE24:9600 → 2001::4455:84C6:39BB:A2DD

Dispatch	Compr. Header	CID	Destination address 64-bit HD
----------	---------------	-----	----------------------------------

(c) Compresión de encabezado, 2001::DEC4:E3A1:FE24:9600 → 2001::4455:84C6:39BB:A2DD

Dispatch	Compr. Header	CID	Source address 64-bit HD	Destination address 64-bit prefix, 64-bit HD
----------	---------------	-----	-----------------------------	---

Figura 10. Ejemplos de compresión del encabezado IPv6 en 6LoWPAN.

3) **Enrutamiento.** Tradicionalmente el enrutamiento IP se lleva a cabo en la capa de red (ver Figura 11), independientemente de las capas inferiores. 6LoWPAN, gracias a su capa de adaptación, puede soportar el enrutamiento en la capa de enlace y en la capa de red, “*mesh-under*” y “*route-over*”, respectivamente.

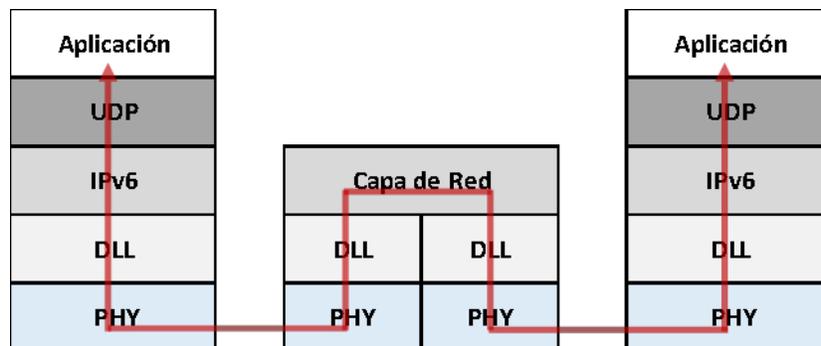


Figura 11. Modelo de enrutamiento IP.

El enrutamiento *mesh-under* se muestra en la Figura 12 y consiste en que la capa de adaptación dirige los paquetes al destino en múltiples saltos radio, es decir, el enrutamiento se hace por medio de las direcciones proporcionadas por los encabezados de la capa de enlace. Para lograr el reenvío de paquetes por múltiples saltos (“*multi-hop*”), la organización IETF 6LoWPAN ha definido un encabezado “*mesh-header*”, que tiene como función principal codificar el límite de saltos (número de segmentos que el paquete puede ser reenviado antes de ser descartado por el enrutador), así como la fuente y el destino de los paquetes en el nivel de enlace. En este tipo de enrutamiento los fragmentos pueden ser entregados a través de múltiples saltos sin la necesidad de realizar la fragmentación y reensamblado en cada salto, esto permite que la transmisión sea

rápida. Sin embargo, si se pierde algún fragmento el paquete debe ser enviado de nuevo desde el nodo original.

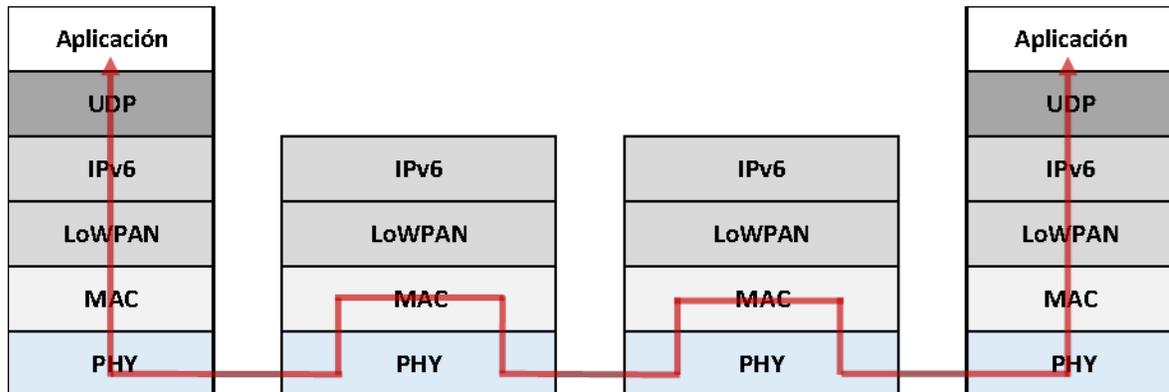


Figura 12. Modelo de enrutamiento "mesh-under" de paquetes en 6LoWPAN.

2.4.2.4 Capa de Red

La capa de red de 6LoWPAN tiene la capacidad de interconectar las redes de sensores, dentro de sus tareas más importantes se encuentran el direccionamiento así como la definición de los protocolos de enrutamiento que se utilizarán para llevar los paquetes hacia su destino. En esta capa ocurre el otro tipo de enrutamiento que soporta el estándar, "route-over", el cual se lleva a cabo como se ilustra en la Figura 13. La principal diferencia con respecto al enrutamiento mesh-under, es el proceso de envío del paquete (fragmentado/reensamblado), o mejor dicho el establecimiento de rutas, esto se debe a que un salto en la capa de enlace corresponde a un salto en IP y cada nodo actúa como router IP. El paquete es enviado salto a salto desde una dirección fuente a una dirección destino. Una vez que el paquete es fragmentado y los segmentos son enviados hacia el siguiente salto; si la capa de adaptación en el siguiente salto recibe todos los segmentos exitosamente, entonces crea un paquete IP y lo envía hacia la capa de red. Después de este al siguiente proceso, la capa de red envía el paquete hacia capas superiores, si el paquete así lo requiere. De otro modo, se envía el paquete al siguiente salto, de acuerdo a una tabla de enrutamiento. La ventaja de este tipo de enrutamiento es que si existe pérdida de fragmentos, únicamente son retransmitidos desde un salto anterior.

En la literatura existen diversos protocolos de enrutamiento adaptados para las capacidades de los nodos de las redes 6LoWPAN en términos de potencia, rango de transmisión, entre otros. Sin embargo el grupo de trabajo ROLL ("Routing Over Low power and Lossy networks") de la IETF propone al protocolo de enrutamiento RPL ("Routing, Protocol for Low Power and Lossy Networks"), como candidato principal

para actuar como protocolo de enrutamiento dentro de las redes LLN (“*Low power and Lossy networks*”) (Cama Pinto, De la Hoz Franco, & Cama Pinto, 2012).

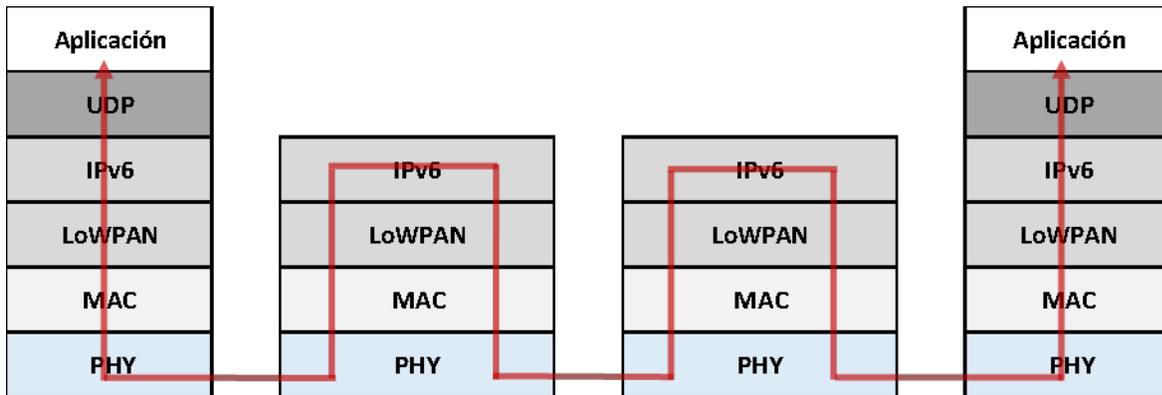


Figura 13. Modelo de enrutamiento “route-over” de paquetes 6LoWPAN.

La características clave de RPL es que está diseñado para proporcionar caminos eficientes de enrutamiento para diferentes tipos de tráfico: multipunto a punto (MP2P por sus siglas en inglés), punto a multipunto (P2MP por sus siglas en inglés) y punto a punto (P2P por sus siglas en inglés) (Kermajani & Gomez, 2011).

En resumen el funcionamiento de RPL se basa en obtener primero una dirección global IPv6, e intentar unirse a una DODAG (“*Destinated Oriented Directes Acyclic Graph*”) por medio del intercambio de mensajes de solicitud de información (DIS, “*DODAG Information Solititation*”) o mensajes de información de objetos DODAG (DIO, “*DODAG Infromation Object*”). Empleando estos mensajes de control un nodo raíz de un RPL DODAG se presenta como padre ante los otros nodos en su vecindad. Por otro lado, una vez que un nodo RPL selecciona a su padre, propaga su propio DIO hacia las ramificaciones contrarias o inferiores de jerarquía en toda la red para formar una sub-DODAG.

RPL puede ser usado con los sistemas operativos de redes de sensores inalámbricos más difundidos, TinyOS y Contiki OS, con los nombres de TinyRPL y ContikiRPL, respectivamente. De acuerdo a lo mencionado anteriormente en esta investigación se hace uso de del protocolo de enrutamiento RPL por dos razones en especial:

- 1) El grupo de trabajo ROLL propuso al protocolo RPL apto para ser implementado en dispositivos que trabajan bajo recursos limitados en potencia, procesamiento, memoria, entre otros. Después de haber obtenido resultados no satisfactorios con algunos protocolos de enrutamiento ya existentes (Felsche, Huhn, & Schwetlick, 2012).

- 2) En esta investigación se hará uso del sistema operativo Contiki OS, el cual provee las librerías necesarias para la implementación de este protocolo, esto evita la necesidad de programar un protocolo de enrutamiento y con ello poder enfocar la investigación hacia los objetivos planteados.

2.4.2.5 Capa de Transporte

Esta capa es muy similar a la capa de transporte del modelo OSI. Esta capa es la responsable del proceso de entrega de los segmentos de información a la capa de aplicación. Existen dos tipos de protocolos de transporte:

- **UDP (“User Datagram Protocol”).** Es un protocolo no orientado a conexión, ya que permite el envío de datagramas a través de la red sin que se haya establecido una conexión, esto se debe a que el propio encabezado cuenta con la información suficiente de direccionamiento. La entrega de paquetes puede no llegar en orden; y es un protocolo poco fiable debido a que no cuenta con un mecanismo de confirmación de entrega o recepción del paquete. La longitud de este encabezado en el estándar de 6LoWPAN corresponde a 8 octetos (ver Figura 14).
- **TCP (“Transmission Control Protocol”).** Es un protocolo orientado a conexión, está enfocado a ser usado para la comunicación confiable entre dos hosts, gracias al establecimiento de conexión, conocida como negociación en tres pasos (“3-way handshake”).

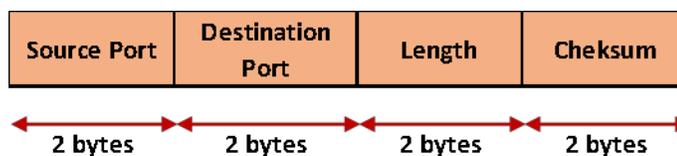


Figura 14. Formato del encabezado de UDP.

A pesar de que TCP es el protocolo de transporte dominante en Internet, genera una sobrecarga de encabezados muy grande que no siempre puede ser soportada por los dispositivos con recursos limitados. Por lo que en aspectos de desempeño, eficiencia y complejidad de implementación, TCP no es una buena opción para ser usado en redes 6LoWPAN (Olsson, 2014). De acuerdo a la literatura recabada se llegó a la conclusión de que el protocolo UDP puede ser una mejor opción en este tipo de redes, gracias a que no exige mantener una conexión, esto reduce el uso de memoria en los dispositivos (Karagiannis, Chatzimisios, Vazquez-Gallego, & Alonso-Zarate, 2015), (Babu & Dey, 2014). Sin embargo la confiabilidad

es un punto a considerar utilizando este protocolo; es por ello que existen protocolos de capa de aplicación (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015) que son los encargados de brindar esta fiabilidad en la transmisión de paquetes; de esta manera se tiene una transferencia de datos fiable a un menor costo, en término de recursos y tiempo de respuesta.

2.4.2.6 Capa de Aplicación

La capa de aplicación de 6LoWPAN utiliza un socket como interfaz, como se muestra en la Figura 15, donde cada aplicación abre un socket con el cual se puede recibir y enviar paquetes. Cada socket está asociado a un protocolo de capa de aplicación, TCP o UDP y a un puerto fuente/destino (Ismail, Hassan, & Ghazali, 2012).

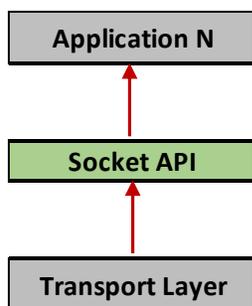


Figura 15. 6LoWPAN utiliza un socket como interfaz para cada aplicación en específico.

De igual manera que los protocolos de capa de red, han sido adaptados diversos protocolos existentes para ser implementados en redes con dispositivos de recursos limitados, algunos de ellos son: REST, CoAP, MQTT, AMPQ, entre otros (Karagiannis et al., 2015),(Foster, 2014) y (Al-Fuqaha et al., 2015).

Es importante mencionar que, para una implementación completa de una aplicación de telemedicina, es necesario considerar los protocolos de mensajero de la capa de aplicación, ya que estos se encargan de la conexión entre los dispositivos de una red distribuida. De tal manera, que dependiendo de la aplicación de telemedicina, será necesario realizar una solución de mensajero adecuada para que cumpla con los requisitos de intercambio de mensajes y datos de la misma. Aunado a ello, habrá que considerar la sobrecarga que aporta a un paquete 6LoWPAN y así cuantificar el total de bytes que se tienen disponibles para carga útil.

En resumen, después de considerar los encabezados que aporta cada una de las capas de la pila de protocolos 6LoWPAN hasta la capa de transporte, el formato de la trama MPDU del estándar 6LoWPAN está compuesta por: 23 bytes que conforman el encabezado y "footer" de la capa MAC del estándar IEEE

802.15.4, 12 bytes agregados por IPv6 (definidos de acuerdo a la norma de compresión HC1) y el encabezado “Hop-by-Hop Options” que define las características del siguiente encabezado en este caso, el encabezado UDP, enseguida se tienen los 8 bytes añadidos por el protocolo UDP y 84 bytes disponibles para carga útil de datos (ver Figura 16).

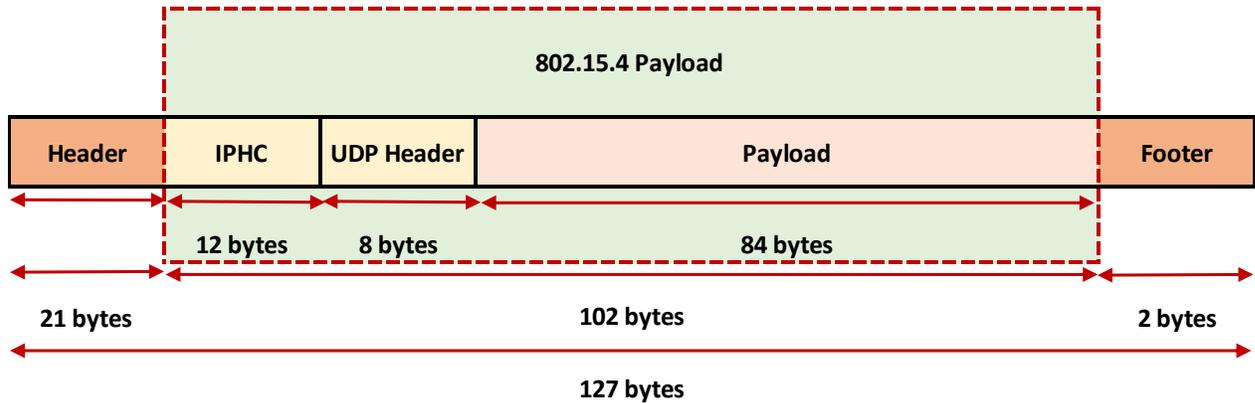


Figura 16. MPDU del estándar 6LoWPAN.

Una vez que se ha definido el formato del paquete 6LoWPAN, es posible comenzar a evaluar el estándar utilizando las herramientas antes mencionadas. A continuación se presenta una descripción más detallada de la pila de protocolos del sistema operativo Contiki y de la pila de protocolos que ofrece TIMAC, los cuales serán utilizados para realizar dicha evaluación.

2.5 Sistema Operativo Contiki

Contiki es un sistema operativo de código abierto, desarrollado en lenguaje C, portable y multitarea. Contiki está enfocado a sistemas de red embebidos de recursos computacionales limitados, como los existentes en las WSN (“Contiki: The open source operating system for the internet of things,” 2015).

Contiki consiste en un núcleo orientado a eventos que incorpora características para el ahorro de energía y memoria por medio de protohilos (“*protothreads*”), de esta manera los programas son cargados y descargados dinámicamente, y son usados para comunicar los distintos procesos. Contiki puede ser portado en varios CPU y plataformas. Su diseño le permite abstraer el sistema base y multiplexarlo para manejar diversas aplicaciones, sensores y servicios para diversos sistemas (Dunkels, Grönvall, & Voigt, 2004).

Contiki cuenta con una pila de protocolos IPv6 completamente implementada con los protocolos TCP, UDP, RPL e ICMP. Este soporta IPv6 sobre IEEE 802.15.4 y proporciona la capa de adaptación de 6LoWPAN,

además de una variedad de capas que controlan el ciclo de trabajo del radio. A continuación se describen las principales características de los protocolos utilizados en cada una de las capas del “*netstack*” de Contiki OS.

2.5.1 Protocolos utilizados en Contiki OS

Contiki está diseñado bajo el principio de la separación de capas, es decir, organiza los módulos de red en una pila de protocolos muy similar al modelo definido para 6LoWPAN (Ver Figura 3). No obstante, a diferencia del modelo de 6LoWPAN, el “*netstack*” (pila de protocolos de Contiki OS) implementado en Contiki cuenta con tres capas diferentes entre la capa de red y la capa física: Trama (“*Framer*”), Ciclo de Trabajo del Radio (RDC, “*Radio Duty-Cycle*”) y mecanismo de acceso al medio (MAC, “*Medium Access Control*”). La Figura 17 muestra la organización de las capas en Contiki OS.

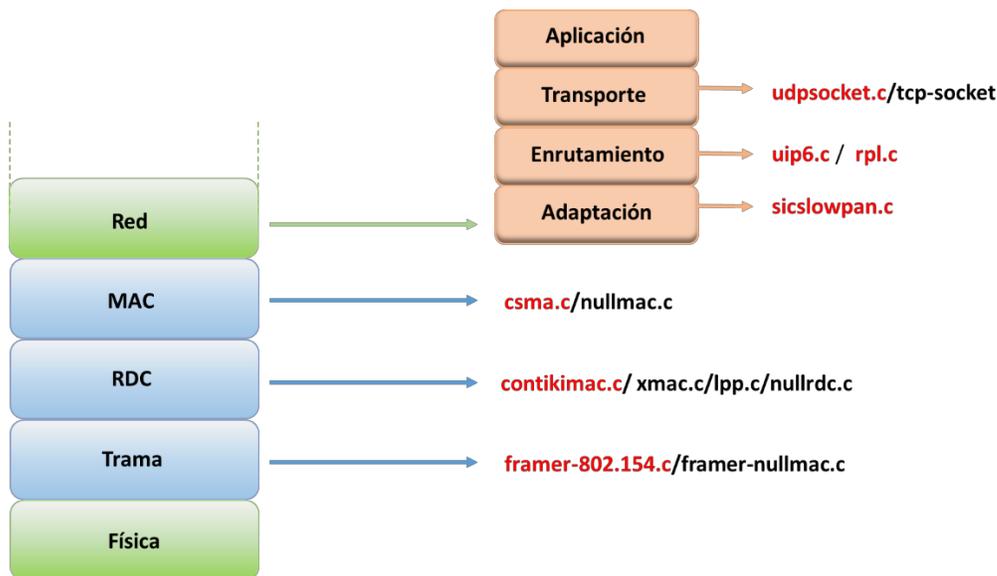


Figura 17. “*Netstack*” de Contiki OS.

2.5.1.1 Capa Radio en Contiki

Es la capa más baja del “*netstack*” de Contiki y es donde se definen los drivers para los transceptores radio que efectuaran la transmisión de manera inalámbrica de los datos hacia su destino. En resumen, cuando se reciben datos, estos llegan byte a byte o como un paquete completo a través de un controlador de interrupciones. Los datos de entrada se leen y son puestos en un buffer y se realiza un mecanismo de encuesta (“*polling*”) al proceso. El “*polling*” permite que el proceso sea dirigido hacia un evento especial y pase hacia capas superiores. En sentido contrario el radio únicamente envía los datos por medio del radio si se ha recibido un ACK del paquete que fue enviado anteriormente.

Algunos de los controladores que maneja el sistema operativo de Contiki son: *“cc2420_driver”* (para los CPU mps430), *“prop_mode_driver r”* (para la familia de CPU CC13XX), *“ieee_mode_driver r”* (para la familia de CPU CC26XX), *“cc2538_rf_driver”* (para los CPU CC2538), entre otros. En el siguiente capítulo se explicará a mayor detalle el controlador que se utilizó para definir esta capa.

2.5.1.2 Capa RDC en Contiki

En las redes inalámbricas de sensores, es deseable que los transceptores radio estén apagados tanto como sea posible reducir el consumo de energía. Es por ello que los desarrolladores de Contiki OS diseñaron una capa RDC, que controla directamente el ciclo de trabajo del radio, es decir, la frecuencia de encendido/apagado del transceptor. Es decir, esta capa es la responsable de decidir exactamente cuándo pueden ser permitidos los paquetes y asegurarse que el nodo se encuentre encendido cuando un paquete vaya a ser recibido. Un posible problema con esta capa es que agrega complejidad a la implementación de los protocolos MAC, ya que el acceso al medio depende, además del protocolo MAC, de la misma capa RDC. Algunos ejemplos de las capas RDC que maneja Contiki son: *“contikimac”*, *“xmac”*, *“lpp”*, y *“nullrdc”*.

ContikiMAC es el mecanismo RDC por default, gracias a que mejora la eficiencia en el consumo de energía, con respecto a la manera que administra el proceso de encendido/apagado del radio. Por otro lado X-MAC, utiliza un *“streaming”* de paquetes pequeños (*“strokes”*) para advertir al destinatario la llegada de una transmisión. De esta manera los vecinos pueden abstenerse de mantener sus radios encendidos si escuchaban un *“stroke”* para otro nodo. El protocolo RDC LPP (*“Low-Power Probing”*), está basado en el uso de mensajes de prueba por parte del receptor para anticipar la recepción de un paquete, pero con mejoras para aumentar el consumo de energía por medio de mecanismos para el envío de datos de tipo *“broascast”*. Por último *“nullrdc”*, como su nombre lo dice es un protocolo RDC nulo, que nunca enciende y apaga el transceptor; además la capa MAC asociada a ella es la capa *“nullMAC”*, lo cual, lo hace incompatible con IEEE 802.11.4.

En este trabajo de tesis se utilizó la capa RDC de ContikiMAC, debido a que está adaptada por Contiki OS para ser compatible con el estándar IEEE 802.15.4, además de acuerdo a (Michel & Quoitin, 2016) ContikiMAC tiene una mayor eficiencia en el ahorro de energía a pesar de tener un procesamiento de transmisión intuitivamente más consumista. Por lo que continuación se describe con mayor detalle su funcionamiento.

Descripción y funcionamiento de ContikiMAC.

Es un protocolo diseñado para ser implementado de manera fácil. Utiliza mecanismos asíncronos⁵ inicializados por el emisor, utilizados cuando se requiere que el emisor transmita un mensaje después de ejecutar su mecanismo de acceso al medio, a diferencia de los mecanismos asíncronos emitidos por el receptor, donde cuando el emisor quiere enviar un mensaje, ejecuta su mecanismo de acceso al medio y si el receptor ha enviado un mensaje de aceptación del mensaje, entonces lo envía (Ver Figura 18). Además, ContikiMAC no utiliza mensajes de señalización y no agrega encabezados adicionales en el paquete. Los paquetes de ContikiMAC son mensajes de la capa MAC (Figura 3).

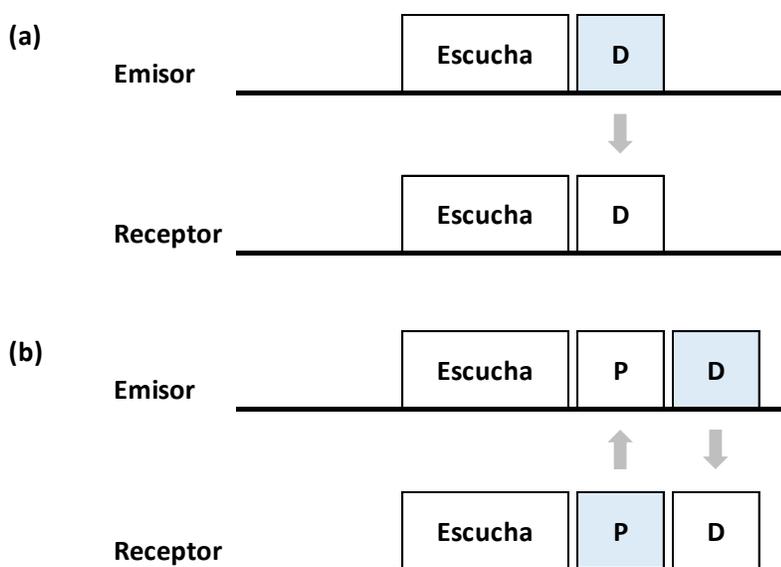


Figura 18. Mecanismos capa RDC asíncronos. (a) Mecanismo asíncrono emitido por el emisor. (b) Mecanismo asíncrono emitido por el receptor.

Una de las características principales de ContikiMAC es que los nodos se encuentran en un estado de inactividad la mayor parte del tiempo y encienden periódicamente para reavivar la actividad del radio. Este intervalo de tiempo está definido con base a una frecuencia del encendido y apagado del radio (f_{RDC}), calculado con la ecuación (1).

⁵ Son mecanismos RDC que pueden ser: iniciados por el emisor, donde el nodo emisor inicializa la comunicación entre él y el receptor; y los iniciados por el receptor, donde el receptor inicia la comunicación.

$$t_{RDC} = \frac{1}{f_{RDC}} \quad (1)$$

Si la transmisión de un paquete es detectada durante un periodo de encendido, el receptor se queda despierto para recibir el siguiente mensaje y envía a la capa de enlace un ACK. Para enviar un paquete, el transmisor envía repetidamente el mismo paquete hasta que recibe el ACK de la capa de enlace, de esta manera se tiene una transmisión exitosa del paquete (ver Figura 19). Sin embargo, cuando se tiene una transmisión fallida, es decir, que los paquetes enviados no reciben un acuse de recibo después de un intervalo de tiempo igual al definido por la ecuación (1), se puede dar por dos razones:

- 1) No se recibió un ACK después de un intervalo de tiempo t_{RDC} , por lo tanto es responsabilidad de la capa MAC programar las próximas retransmisiones de este paquete, y ejecutar este proceso se excedió el número máximo de retransmisiones.
- 2) No se recibió ACK, debido a que dos o más nodos ECG intentan enviar o retransmitir un paquete durante el mismo tiempo de contención, ocasionando que haya colisiones entre paquetes.

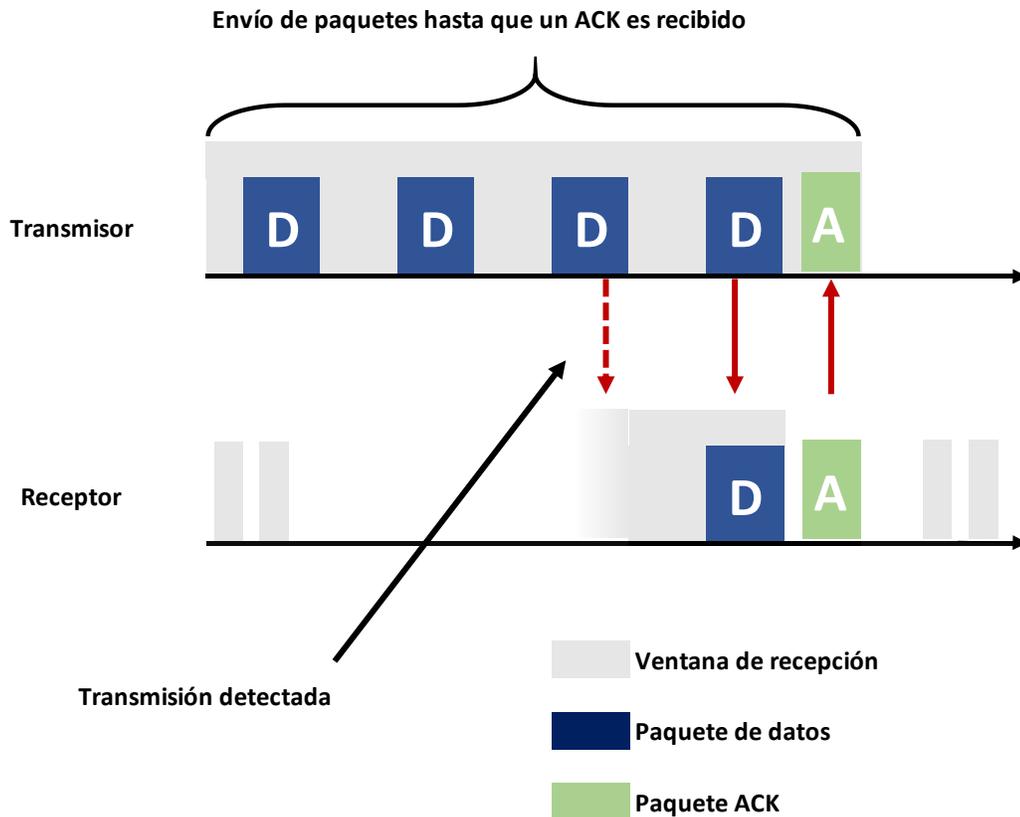


Figura 19. Proceso de transmisión para enviar un paquete en ContikiMAC.

Cabe mencionar que para los paquetes de tipo broadcast (ver Figura 20), el emisor envía repetidamente el paquete durante el intervalo de encendido para asegurarse que todos los vecinos lo han recibido, es decir, no se utilizan paquetes de ACK (Dunkels, 2011).

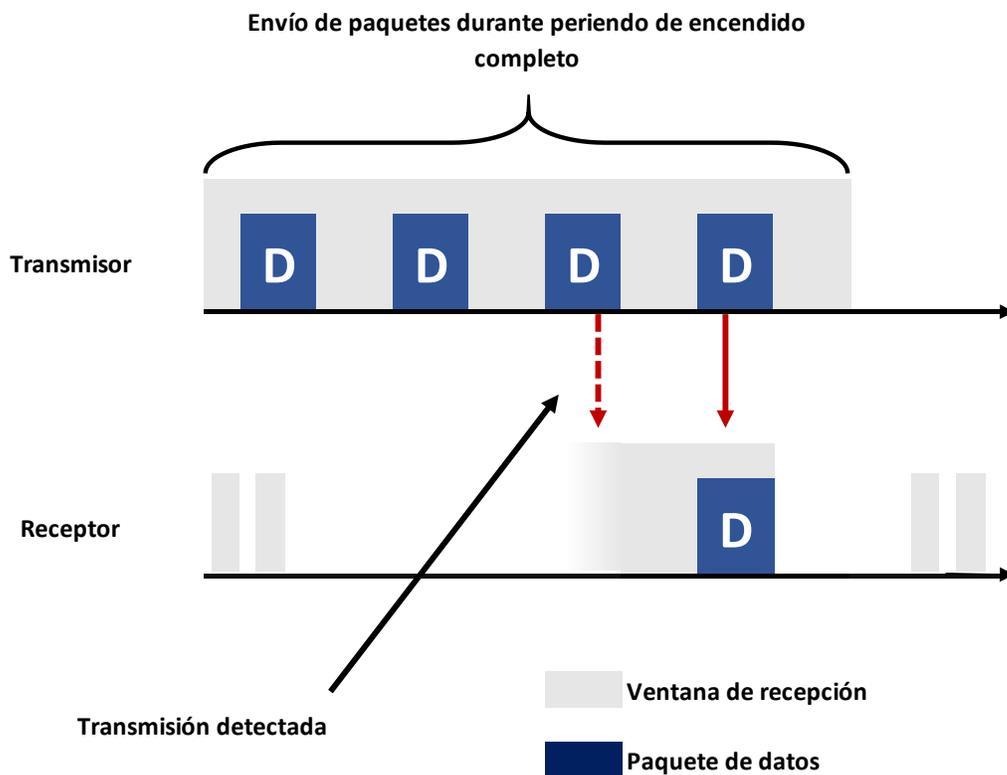


Figura 20. Transmisión de paquetes “broadcast” con la capa ContikiMAC.

ContikiMAC tiene un mecanismo de “wake-up” eficiente en potencia, y este se basa en la precisión de tiempo entre las transmisiones. Para despertar el radio, ContikiMAC realiza una evaluación del canal (CCA) utilizando el valor de potencia de la señal (RSSI, “Receiver Signal Strength Indicator”) medido en el transceptor radio para dar a conocer la actividad que hay en el canal. Si el valor de RSSI se encuentra arriba del umbral definido, CCA regresa un valor negativo, indicando que el canal está en uso; en el caso contrario interpreta que el canal se encuentra libre.

En la Figura 21 se muestra la distribución de tiempos que se emplea en ContikiMAC para la transmisión de un paquete. Cuando se utiliza tanto la trama como la capa física del estándar IEEE 802.15.4 la duración de estos tiempos queda definida de acuerdo a las especificaciones establecidas en dicho estándar. Lo cual corresponde a la implementación prevista en este trabajo de tesis.

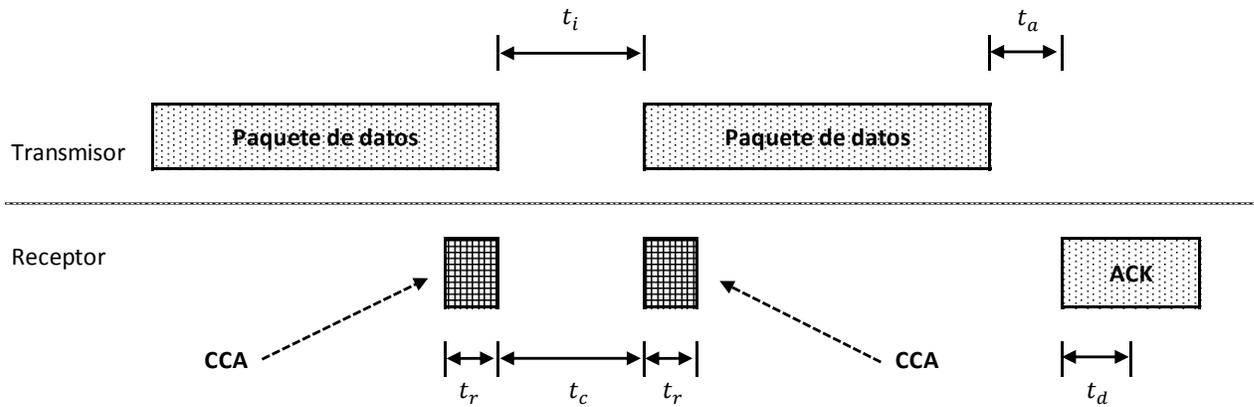


Figura 21. Distribución de tiempos en la capa RDC ContikiMAC (Dunkels, 2011) .

t_i : Intervalo entre cada transmisión de paquete. $t_i < t_c$. Esto es para asegurarse que en la primera o segunda evaluación del canal (CCA) es posible ver la transmisión paquete. Si $t_c < t_i$ las dos evaluaciones del canal no podrán ser capaces de detectar con fiabilidad una transmisión.

t_r : El tiempo requerido para obtener un RSSI estable.

t_c : Intervalo entre cada CCA.

t_a : Tiempo entre la recepción de un paquete y el envío de un paquete ACK.

t_d : Tiempo necesario para el éxito de la detección de un ACK desde el receptor.

Las consideraciones para los tiempos t_c y t_i también son tomadas en cuenta para definir el tamaño del paquete más corto que ContikiMAC puede soportar. Ya que la transmisión de un paquete no puede ser tan corta que se encuentre entre las dos evaluaciones del canal CCA's. Dicho de otro modo el tiempo de transmisión del paquete (t_s) debe ser mayor que $t_r + t_c + t_r$. Cuando se detecta la transmisión de un paquete en alguna de las evaluaciones del canal, ContikiMAC mantiene el radio encendido para recibir el paquete completo. Una vez que se ha recibido por completamente el paquete, se transmite un acuse de recibo por parte de la capa de enlace. El tiempo máximo para que un paquete ACK sea detectado es, t_d , que establece el límite inferior para el intervalo de comprobación t_c .

En resumen los requerimientos de tiempo de ContikiMAC son:

$$t_a + t_d < t_i < t_c < t_c + 2t_r < t_s$$

2.5.1.3 Capa Trama en Contiki

La capa de Trama (“*Framer*”) no es una implementación de una capa regular, de hecho es una recopilación de funciones auxiliares que son llamadas para crear una trama con datos que serán transmitidos y posteriormente analizados en la recepción. En Contiki se incluyen diferentes formatos de trama tales como: “*frame_802154*”, utilizado para el formato de MSDU de la trama del estándar IEEE 802.15.4; y “*frame_nullmac*”, este tipo de trama trabaja en conjunto con la capa MAC “*nullmac*” y está compuesto por un encabezado de dos campos, que contiene: las direcciones fuente y destino.

2.5.1.4 Capa MAC en Contiki

Esta capa se ubica en la parte superior de la capa RDC y es la responsable de evitar las colisiones que puedan presentarse en el canal radio así como la retransmisión de paquetes si hubo alguna colisión. Contiki proporciona dos capas MAC:

- **Mecanismo CSMA.** Recibe los paquetes entrantes desde la capa RDC y utiliza la capa RDC para transmitir los paquetes. Si la capa RDC o la capa de radio detectan una colisión, la capa MAC puede retransmitir los paquetes en un punto posterior en el tiempo.

A pesar de que es un mecanismo de acceso al medio por detección de portadora, la aplicación en Contiki no se basa en la detección de portadora debido a que el acceso al medio se realiza mediante el protocolo de la capa RDC. En otras palabras, la capa MAC recibe los paquetes entrantes de la capa RDC y usa esta capa para transmitir los paquetes.

Funcionamiento de CSMA

En esta técnica de acceso se realiza un sensado o detección de portadora. A través del sensado de portadora las estaciones escuchan el medio, es decir, determinan si en el momento en el que requieren transmitir existe alguna otra transmisión en proceso dentro de su alcance, con lo cual se reducen las colisiones dentro de la red. De manera que un nodo únicamente transmite su información cuando detecta libre el canal, de lo contrario espera un periodo aleatorio (determinado por el algoritmo de “*backoff*” utilizado) para volver a intentar realizar una transmisión (Ilyas & Ahmed, 2011).

El sensado del canal es llevado a cabo mediante un procedimiento de evaluación de canal libre (“*Clear Channel Assessment*” – CCA). Este procedimiento puede ser realizado en tres diferentes maneras (Bertocco, Gamba, & Sona, 2008):

- Se asume que el medio está ocupado si el nivel de potencia detectado (“*Received Signal Strength Indicator*” - RSSI) es más alto que un umbral predefinido.
- Se asume que el medio está ocupado si se detecta al menos una señal con características de modulación y esparcimiento.
- Se asume que el medio está ocupado si las dos condiciones anteriores son detectadas.
- **Mecanismo NullMAC.** Recibe los paquetes entrantes desde la capa RDC y no realiza ningún tipo de procesamiento a nivel de capa MAC, en otras palabras, solamente reenvía los paquetes desde capas superiores hacia la capa de radio y viceversa.

Funcionamiento de NullMAC

En esta técnica nunca cambia de un estado encendido a un estado apagado, es decir, el radio siempre se mantienen encendido, es por esta razón que es comúnmente utilizado para pruebas de comparación con otros drivers de la capa RDC en Contiki OS. Este protocolo no proporciona ningún control de acceso al medio.

2.5.1.5 Capa de Red en Contiki

Dentro de esta capa se engloban las subcapas de adaptación, enrutamiento, transporte y aplicación. Cada una de ellas con sus respectivos protocolos. En este caso es importante mencionar las subcapas de la capa de red y los protocolos que serán utilizados en esta investigación:

Subcapa de adaptación. En esta capa se provee la compresión del encabezado IPv6 y UDP, el proceso de fragmentación que permite transportar los paquetes IPv6 con una MTU de 1280 bytes sobre el estándar IEEE 802.154 con una MTU de 127 bytes. Contiki OS cuenta con el controlador “*sicslowpan*” que hace referencia a la descripción anteriormente mencionada.

Subcapa de enrutamiento. Contiki automáticamente forma una red inalámbrica IPv6 con la ayuda de la implementación del protocolo de enrutamiento RPL ("*ContikiRPL*") de acuerdo a la especificación del grupo de trabajo ROLL de la IETF y el Protocolo de Internet versión 6.

Subcapa de transporte. En esta subcapa Contiki OS cuenta con un conjunto de librerías que brindan soporte para dos protocolos de transporte, UDP y TCP.

En este trabajo de tesis únicamente se evaluará el estándar de 6LoWPAN hasta la capa de transporte, debido a que el alcance de esta investigación es únicamente verificar el comportamiento que añade las características del sistema operativo de Contiki OS, en específico, la capa RDC. Para ello no es necesario emplear los protocolos de mensajero de la capa de aplicación, ya que esto requeriría la implementación de un edge router y un servidor, como parte de la arquitectura de la red. No obstante es importante mencionar que la subcapa de aplicación también es proporcionada por el sistema operativo de Contiki y cuenta con algunos protocolos de mensajero (RESTful, CoAP, HTTP y MQTT) que complementan las tareas proporcionadas por la capa de transporte.

Dentro de Contiki OS se encuentra un simulador de redes denominado Cooja, el cual es utilizado para realizar una evaluación simulada del estándar 6LoWPAN que permita obtener un acercamiento al comportamiento experimental del mismo. Enseguida se describe brevemente este simulador.

2.6 Simulador de redes Cooja

Cooja es un simulador flexible basado en Java diseñado para la simulación de redes inalámbricas de sensores que corre en el sistema operativo Contiki (Österlind, Dunkels, Eriksson, Finne, & Voigt, 2006), por lo tanto soporta el mismo conjunto de protocolos.

Además Cooja permite simular cuatro diferentes modelos de propagación: UDGM con pérdidas constantes ("*Unit Disk Graph Medium- Constant Loss*"), UDGM con pérdidas en relación a la distancia ("*Unit Disk Graph Medium- Distance Loss*"), DGRM ("*Directed Graph Radio Medium*") y MRM ("*Multi-path Ray-tracer Medium*"); la elección de un modelo de propagación afectará el comportamiento que tendrá la señal que viaja por el medio y por consecuencia si está llega con éxito al receptor (Stehlík, 2011).

En este trabajo de investigación se eligió utilizar el modelo de propagación UDGM con pérdidas en relación a la distancia debido a que el rango de transmisión se define de acuerdo a un área de cobertura. En la Figura 22 se muestra una red de 3 nodos que envían sus datos hacia un nodo "*sink*" (nodo 1); como se

puede observar cuando el nodo 3 transmite sus datos causa interferencia a sus nodos vecinos ubicados dentro de la superficie de color gris, esto hará que si alguno de ellos desea transmitir un mensaje este pueda no ser enviado con éxito. Sin embargo, el nodo 3 puede comunicarse perfectamente con el nodo 1, ya que el modelo UDGm tiene un patrón de pérdidas constantes dentro del área de cobertura del radio en color verde. Este patrón de pérdidas está basado en una probabilidad de éxito, definida en el código fuente.

En general Cooja proporciona un entorno muy útil que permite visualizar los eventos de radio, capturas de paquetes, monitorea variables, se tiene una visualización de la red y muestra los registros de cada nodo. Esto facilita el desarrollo y evaluación de cada uno de los protocolos así como del diseño de un desarrollo embebido a gran escala.

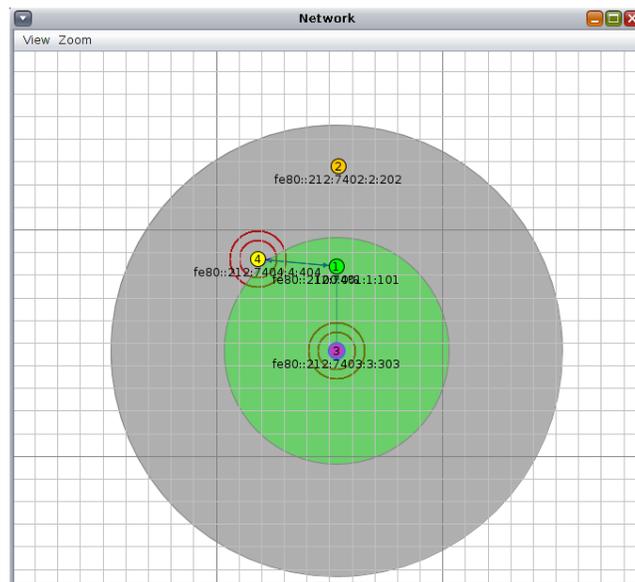


Figura 22. Modelo de propagación UDGm de una red de cuatro nodos en Cooja, el nodo 3 se comunica con el nodo 1 e interfiere con los nodos 2 y 4.

De acuerdo al “*netstack*” que ofrece el sistema operativo de Contiki que se mencionó en la sección 2.5.1, se puede percibir que no se cumple exactamente con las especificaciones que dicta el estándar de 6LoWPAN. Esto se debe a que Contiki OS, agrega una capa RDC para mejorar la eficiencia en el ciclo de trabajo del radio. Lo que ocasiona que el mecanismo de control de acceso al medio no sea controlado por CSMA sino por el control de la capa RDC. La razón principal por la cual Contiki OS agrega esta capa se debe a que con ella se puede regular el proceso de encendido/apagado del radio, lo cual permite tener un ahorro de energía. Es por ello, que el “*netstack*” de Contiki OS no se apega al funcionamiento de la capa MAC que el estándar de 6LoWPAN propone (IEEE 802.15.4 MAC), donde el control de acceso al medio es

realizado por la técnica CSMA. Esto conlleva a que con la presencia de esta capa, se puedan introducir comportamientos o modos de trabajo que no son considerados por el estándar. Por esta razón se ve en la necesidad de utilizar TIMAC, que ofrece una capa PHY y MAC más pura del estándar IEEE 802.15.4. De este modo también se podrá evaluar la implementación del estándar 6LoWPAN y así se podrá comparar con la implementación que se realice utilizando Contiki OS.

2.7 Resumen del capítulo

En este capítulo se presentaron las características de los estándares IEEE 802.15.4 y 6LoWPAN más relevantes para este trabajo de tesis. A partir del marco teórico que se presentó en este capítulo se determinó que, debido a los encabezados de capas superiores MAC, únicamente se tienen disponibles 84 bytes de carga útil en los cuales se pueden portar los datos de las variables fisiológicas a transmitir. Lo cual servirá como un parámetro para el modelo del escenario de la implementación del estándar 6LoWPAN.

Por otro lado, también se han descrito en este capítulo las características y funcionamiento del sistema operativo Contiki OS, el cual proporciona el “*netstack*” para la implementación del estándar 6LoWPAN, orientado a eventos que permiten mejorar la eficiencia en el consumo de energía. Algunas de las razones por las que en esta investigación se eligió este sistema para realizar la evaluación del estándar de 6LoWPAN fueron las siguientes:

- Permite el uso eficiente de energía del sistema, es compacto y ligero en el código de programación.
- Es un sistema de alta portabilidad a diversas plataformas. En este trabajo de investigación la plataforma que se manejará en la implementación es el kit de desarrollo CC2650STK de Texas Instruments, compañía que recomienda el uso de Contiki para el desarrollo de aplicaciones con el estándar 6LoWPAN.
- Software libre, Contiki OS cuenta con código abierto, esto lo hace ser un sistema versátil, es decir, brinda la posibilidad de diseñar y desarrollar, aplicaciones, etc.

Adicionalmente, debido al marco temporal en el que se debe desarrollar este trabajo, utilizar ContikiOS supone las siguientes ventajas frente a TinyOs:

- La implementación desarrollada por Contiki OS puede ser portada fácilmente al simulador Cooja y viceversa.

- Se tiene conocimiento del lenguaje de programación C el cual es usado por Contiki en contra posición a otros sistemas operativos que utilizan 6LoWPAN (p. ej. TinyOS) esto representado en horas de aprendizaje y facilidad al programar.
- Existe amplia documentación en Internet, tales como repositorios, artículos, foros, por mencionar algunos.

De acuerdo a lo mencionado en este capítulo la pila de protocolos que propone el estándar de 6LoWPAN oficial difiere un poco con el “*netstack*” que ofrece el sistema operativo de Contiki OS. En la mayoría de las capas Contiki OS cuenta con algunos de los protocolos existentes adaptados para dispositivos de bajo consumo, como por ejemplo: IEEE 802.15.4, IPv6, RPL, TCP, UDP, HTTP, CoAP, RESTfull, entre otros. Agrega a su “*netstack*” la capa RDC, que en conjunto con la técnica de acceso al medio CSMA/CA indican cuando se transmite o no un paquete. Por lo que es necesario evaluar el impacto que esta capa tiene sobre el desempeño de la red de telemonitoreo.

Es importante mencionar que Contiki OS agrega a su “*netstack*” la capa RDC, que en conjunto con la técnica de acceso al medio CSMA/CA indican cuando se transmite o no un paquete. Por lo que es necesario evaluar el impacto que esta capa tiene sobre el desempeño de la red de telemonitoreo.

En el siguiente capítulo se describirán los principales señales biomédicas que se encuentran en un escenario de telemonitoreo. Así mismo se definirá el perfil de tráfico y la arquitectura que se propone para habilitar la red que se utilizará para evaluar la implementación del estándar 6LoWPAN en un escenario de telemonitoreo.

Capítulo 3. Transmisión de variables fisiológicas sobre 6LoWPAN

3.1 Introducción

En las redes inalámbricas de sensores actuales circula todo tipo de tráfico, desde el envío de paquetes de control hasta paquetes de gran tamaño, donde se maneja “*streaming*” de datos. Cada tipo de tráfico tiene unas necesidades determinadas que las redes de comunicación deben cumplir para garantizar la calidad de servicio que la aplicación requiera. Es por esta razón, que la implementación de aplicaciones enfocadas en aplicaciones del cuidado de la salud, en las cuales se envía la información relacionada sobre el diagnóstico de un paciente, no es una tarea trivial, ya que es importante considerar las capacidades de procesamiento y de comunicación de las redes de sensores inalámbricas que se utilizan para implementarlas.

En escenarios de telemonitoreo en hospitales o en casa, los sensores deben enviar los datos de manera inalámbrica hacia un resumidero de información para su posterior procesamiento y análisis. Algunos de los posibles escenarios pueden ser:

- El resumidero de información se encuentra en el mismo cuarto en el que está el paciente, y el paciente no tiene mucha movilidad, es posible transmitir la información utilizando una WPAN.
- El resumidero se encuentra en una localidad remota, es decir, el paciente tiene movilidad, entonces es necesario contar con un mecanismo que permita conectar la WPAN con una red de mayor área de cobertura o en su defecto diseñar una red en la cual cada sensor tenga un alcance de transmisión lo suficientemente amplio para conectarse con el resumidero.

En cualquiera de los casos mencionados anteriormente, para realizar un diseño adecuado del sistema de telemonitoreo es necesario conocer las características particulares de las señales a transmitir. El conjunto de variables puede ser diferente dependiendo de la aplicación deseada y situación clínica del paciente, es decir, cada enfermedad requiere del monitoreo de ciertas variables fisiológicas específicas.

Es por ello que en la siguiente sección (Ver sección 3.2) se describe brevemente algunas de las variables fisiológicas más comunes en un escenario de telemonitoreo.

3.2 Variables fisiológicas en un escenario de telemonitoreo

Las variables fisiológicas, siendo análogas por naturaleza, son caracterizadas por medio del conjunto de valores que pueden tomar y su correspondiente espectro en frecuencia. Cada variable fisiológica debe ser muestreada a una tasa mayor o igual al doble de su frecuencia más alta en el espectro y enviada en forma de muestras digitalizadas hacia un resumidero de información a través de la red inalámbrica.

Dependiendo de la aplicación, el conjunto de variables fisiológicas a monitorear varía. De acuerdo a la revisión de literatura, las variables fisiológicas comúnmente utilizadas en escenarios de telemonitoreo dentro de hospitales y en el hogar son la temperatura, presión sanguínea, saturación de oxígeno en la sangre, gasto cardiaco y la actividad eléctrica del corazón (Casillas, 2012) (Alinejad, Philip, & Istepanian, 2011) (Jelena Misic & Misic, 2010) (Paksuniemi, Sorvoja, Alasaarela, & Myllyla, 2005).

La mayoría de las variables para el telemonitoreo no se consideran sensibles al retardo (Malindi, 2011); a excepción de la actividad eléctrica del corazón (medida con electrodos, obteniendo con ello un electrocardiograma- ECG), con una sensibilidad al retardo máxima de 1 segundo (Skorin-Kapov & Matijasevic, 2010). Adicionalmente, transmitir de manera continua el ECG requiere un ancho de banda mayor en relación a las otras variables fisiológicas mencionadas. Aún más, es importante mencionar que la variable fisiológica ECG se encuentra presente dentro de la mayoría de los perfiles de tráfico de los escenarios de telemonitoreo en la actualidad, esto se debe a que las enfermedades cardíacas forman parte de las principales causas de mortalidad dentro del perfil epidemiológico de los adultos mayores en nuestro país⁶ según la Dirección General de Epidemiología (DGE) en México. Por lo tanto, en esta investigación se decidió evaluar la implementación del estándar considerando un ECG como variable fisiológica a transmitir.

A continuación se muestra una breve descripción del ECG y la metodología para la generación de tráfico correspondiente a un ECG que será enviado sobre un paquete 6LoWPAN.

3.2.1 Electrocardiograma (ECG)

El ECG es una variable fisiológica que se obtiene a partir de la medición superficial del potencial eléctrico generado por la actividad eléctrica que controla la acción de bombeo de las fibras musculares cardíacas (Hartmann, 2003). Es una de las herramientas que se utiliza para el diagnóstico de enfermedades

⁶ <http://www.beta.inegi.org.mx/proyectos/registros/vitales/mortalidad/>

cardiovasculares, alteraciones del ritmo cardíaco y crecimiento de las cavidades cardiacas. La información obtenida del ECG es recolectada por electrodos ubicados en posiciones específicas en el cuerpo del paciente. Cada derivación obtiene el potencial eléctrico entre dos puntos corporales. Para una aplicación mínima, solamente es necesario utilizar 3 electrodos en ubicados en el pecho del paciente, de esta manera se pueden localizar un amplio número de arritmias cardiacas y así obtener las mediciones necesarias para tener un buen diagnóstico.

El ancho de banda útil de una señal de ECG, dependiendo de la aplicación, puede variar, sin embargo para una aplicación de monitoreo estándar se considera un ancho de banda de 0.05 Hz hasta 100 Hz (Hartmann, 2003). De acuerdo a este intervalo de ancho de banda y al teorema de muestreo de Nyquist se puede concluir que la frecuencia de muestreo de un ECG es de al menos 200 Hz. Sin embargo se han realizado estudios donde se demuestra que no existe diferencia significativa al muestrear una señal de ECG a más de 250 Hz (Pizzutti, Cifaldi, & Nolfi, 1985). Por lo tanto si se utiliza una frecuencia de muestreo de 300 Hz y cada muestra es digitalizada con 16 bits, se tendrá una tasa de datos por cada electrodo de ECG de $R_{ECG} = (300 \text{ Hz})(16 \text{ bits}) = 4800 \text{ bps}$.

En la siguiente sección se describe la manera en que se emuló el tráfico producido por el ECG. Además se define una técnica de empaquetamiento para reducir el porcentaje de sobrecarga en un paquete 6LoWPAN en relación al número de muestras en cada paquete.

3.2.2 Emulación del tráfico de ECG

Con el objetivo de realizar una mejor evaluación del desempeño, es importante considerar las características del perfil de tráfico de la variable a transmitir. Para este fin, a continuación, se detalla la metodología utilizada en la formación del paquete correspondiente a la variable fisiológica de ECG generada por cada nodo.

Para este trabajo se considera la transmisión de 3 canales de ECG. La tasa de muestreo para cada canal es de 300 Hz, con una resolución de 16 bits por muestra, debido al ancho de banda de la señal de ECG estos parámetros son comúnmente utilizados en la literatura (Casillas, 2012).

$$t = \frac{1}{f_m} \quad (2)$$

$$t_{ECG} = \frac{1}{300 \text{ Hz}} = 3.333 \text{ ms}$$

Esto equivale a una tasa de transmisión total de 14 400 bps para el ECG. Dado que se generarán en total 48 bits de información cada 3.33 ms, calculado con la ecuación (2). De acuerdo a lo descrito en la sección 2.4.2), se tienen disponible 84 bytes de carga útil. Si se envían las muestras de ECG conforme se generan, la sobrecarga de cada paquete (ver Figura 23) sería de cerca del 89.09%.



Figura 23. Paquete 6LoWPAN sin empaquetamiento.

Por lo tanto, considerando que el mecanismo acceso al medio de IEEE 802.15.4 es basado en contención, si no se empaquetan las muestras se estaría inundando el canal con paquetes 6LoWPAN que transportan pocos datos. Por lo que se considerará el empaquetamiento de varias muestras de ECG para establecer un escenario que provea las mejores condiciones para evaluar el funcionamiento de las métricas seleccionadas. Específicamente, se busca empaquetar el mayor número de muestras posibles en un solo paquete. Para esto, por operaciones simples se puede determinar que caben un máximo de 42 muestras (14 por canal) de 2 bytes cada una si se asume una carga útil de 84 bytes. La técnica de empaquetamiento descrita se muestra de manera gráfica en la Figura 24.

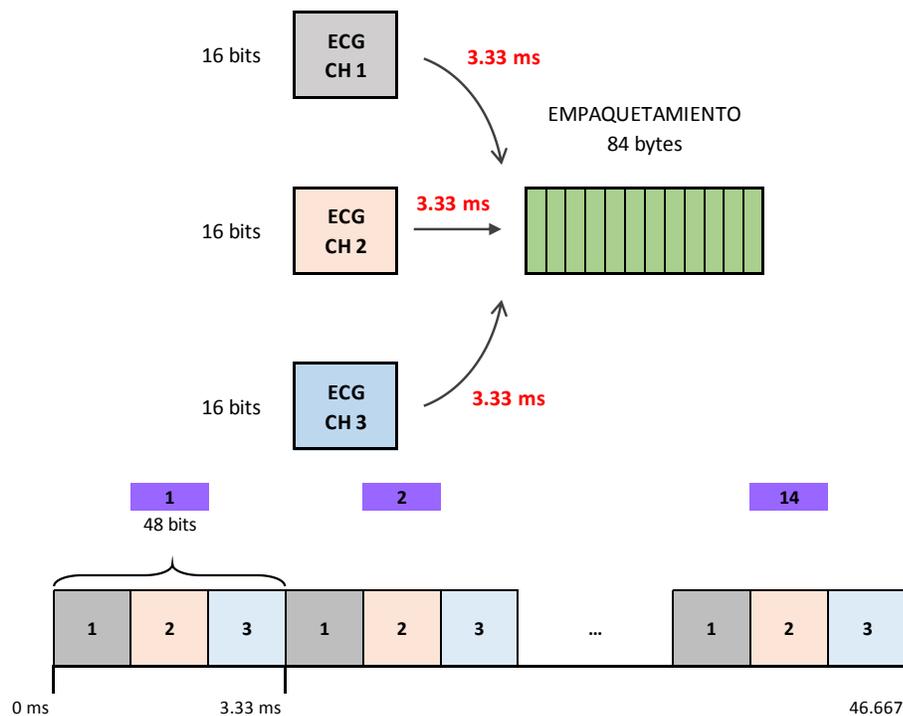


Figura 24. Empaquetamiento de 48 muestras para un ECG de 3 canales.

3.3.1 Capa Física

De acuerdo a lo mencionado en el capítulo la capa física de 6LoWPAN puede estar definida por cualquier tecnología que habilite redes LoWPAN. Es por esta razón que se seleccionó a IEEE 802.15.4 PHY ya que en la actualidad este estándar es el más relevante en la implementación del estándar 6LoWPAN, debido a que fue la primera tecnología que adoptó el grupo de trabajo 6lo para transportar paquetes IPv6 sobre WPANs.

3.3.2 Capa de acceso al medio

La capa MAC está definida por el estándar IEEE 802.15.4, de tal manera que se tiene presente al protocolo CSMA, el cual tiene como principal tarea escuchar el canal antes de transmitir sus datos, utilizando la técnica CCA. Con ello, evita que se tengan colisiones entre paquetes. No obstante, dentro del sistema operativo de Contiki OS la capa MAC trabaja en conjunto con otra capa (capa RDC) encargada de decidir cuándo se transmitirá el paquete con base al estado de actividad (encendido/apagado) del radio.

Cabe mencionar que para este trabajo de investigación se utilizó IEEE 802.15.4 sin “*beacons*”, debido a que Contiki OS aún no cuenta con la implementación de éste con “*beacons*”. No obstante, es importante mencionar que una de las prioridades de esta tesis es cuantificar el número de usuarios que pueden ser servidos. Por lo cual, es innecesario contar con un formato de supertrama e intervalos de tiempo definidos (utilizado en IEEE 802.15.4 con “*beacons*”)⁷.

3.3.3 Capa de adaptación

Esta capa brindará el soporte entre IPv6 y la tecnología definida en la capa PHY y MAC, en este caso el estándar IEEE 802.15.4 PHY/MAC. De esta manera se tendrá comunicación fluida con las redes convencionales basadas en IP. Esta capa es gran relevancia, ya que dentro de ella se realizan las tareas de compresión de los encabezados IPv6 y UDP, así como la fragmentación y reensamblado de los datagrama para posteriormente ser encapsulados dentro de la MSDU del paquete 6LoWPAN.

3.3.4 Capa de red

La capa de red, está compuesta por los protocolos de red y enrutamiento. De manera que se pueda brindar conectividad a Internet y se provean habilidades de direccionamiento y enrutamiento de los paquetes

⁷ Utilizar una supertrama sería favorable si lo que se requiere es servir señales biomédicas con la utilización de ranuras de tiempo garantizadas (J Misic & Misic, 2009).

desde una fuente hacia un destino. Es por ello, que los protocolos que se definieron en esta capa son: IPv6 y RPL, respectivamente.

3.3.5 Capa de transporte

En esta capa se define la manera en que se entregarán los segmentos de información hacia la capa de aplicación. Para la red 6LoWPAN propuesta, se eligió al protocolo UDP, debido a que reduce el uso de memoria en dispositivos, la sobrecarga de encabezados y la complejidad en su implementación, en comparación con TCP.

Asimismo, la arquitectura de red propuesta es una LoWPAN de tipo *“ad-hoc”*, en donde no se tiene conectividad a Internet. Esto debido a que las pérdidas y retardos introducidos por la transmisión del ECG a través de Internet podrían introducir un sesgo en la evaluación del desempeño del estándar 6LoWPAN para habilitar la WPAN. Por lo tanto, como se muestra en la Figura 26 los nodos ECG y el nodo coordinador, están compuestos por *“netstack”* del estándar 6LoWPAN de Contiki OS (ver sección 2.5.12.5.1), hasta la subcapa de transporte. Es decir, no se considera la implementación de los protocolos de la capa de aplicación, cuya función precisamente es especificar la manera en la cual se accede a la información de un cliente y realizar las actualizaciones correspondientes de la misma. No obstante, es importante mencionar que si se considera la sobrecarga que los protocolos de la capa de aplicación introducen tendrá un impacto en la reducción del tamaño de la carga útil, sin embargo, evaluar este aspecto no está dentro de los alcances de esta investigación.

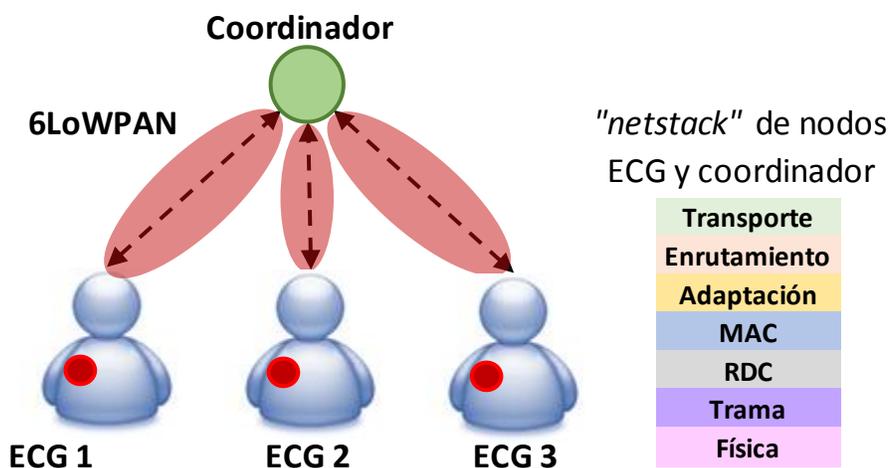


Figura 26. Arquitectura y *“netstack”* de la red 6LoWPAN propuesta.

El desempeño de la implementación de dicha red 6LoWPAN será evaluado con base a métricas de pérdida de paquetes y retransmisiones realizadas.

3.4 Resumen del capítulo

Durante este capítulo se presentó la relevancia que tiene el envío de paquetes con información relacionada al cuidado de la salud así como los posibles escenarios de telemonitoreo que se pueden presentar. Esto se puede definir con base al nivel de gravedad del paciente, ya que existen escenarios más críticos que otros.

Además a partir de la información presentada en este capítulo es posible concluir que los requerimientos de transmisión del tráfico producido por un ECG pueden ser servidos por el estándar 6Lo WPAN. Aunado a ello, se definió una técnica de empaquetamiento que permite enviar la máxima carga útil posible dentro de un paquete de 6LoWPAN (14 muestras de ECG), reduciendo la sobrecarga de 89.09 % a un 36.84 %. Esto permitirá evitar que el canal sea inundado de paquetes 6LoWPAN que envían un número reducido de muestras ECG.

Por otro lado, dentro de este capítulo, también se definió la arquitectura de la red 6LoWPAN, es decir, se describieron las características y protocolos de comunicación que permitirán habilitar tanto en la simulación como en la implementación física de la red 6LoWPAN. Con base a ello, se pudo concluir que para los fines de esta investigación es innecesario contar con una red de tipo extendida, debido a que la conectividad a Internet puede introducir la presencia de otros aspectos (p. ej. pérdidas, retardos, por mencionar algunos) que pueden introducir un sesgo en la evaluación del desempeño del estándar.

En el siguiente capítulo se describirán las consideraciones del escenario tanto de la simulación como la implementación física de una red de sensores utilizando el estándar de 6LoWPAN, para el transporte de variables fisiológicas.

Capítulo 4. Simulación e implementación de la WSN para la transmisión de señales fisiológicas sobre 6LoWPAN.

4.1 Introducción

Dentro de este capítulo se presentan los parámetros, el equipo utilizado y la metodología seguida para la creación de la red 6LoWPAN utilizando el simulador Cooja y Contiki OS, respectivamente. De igual manera, se muestran las características y equipo utilizado de la implementación de la misma red 6LoWPAN utilizando la pila de protocolos que ofrece TIMAC, la cual servirá de referencia para observar una implementación más pura de las capa PHY y MAC de 6LoWPAN.

4.2 Pila de protocolos (“*netstack*” Contiki OS)

A continuación se presenta el “*netstack*” de Contiki OS que define la pila de protocolos de cada uno de los nodos que conforman la red 6LoWPAN propuesta. Esto fue posible definirlo a partir de las capas y protocolos definidos en la sección 3.3.

4.2.1 Capa Radio.

Al no existir el emulador del microcontrolador CC2650STK en el simulador Cooja, se utilizó el mote Sky (“*cc2420_driver*”) predeterminado, además posteriormente se realizaron las mismas pruebas para otro mote (Zolertia), esto para corroborar que no existe discrepancia en el tipo de mote que se utiliza. Este controlador conforma el protocolo correspondiente a la capa física del estándar IEEE 802.15.4 del nodo coordinador y los nodos ECG, con las especificaciones del mote Sky.

4.2.2 Capa RDC.

Con base a la literatura revisada durante el desarrollo de esta investigación, la mayoría de los artículos mencionan el uso de la capa RDC ContikiMAC, en aplicaciones de diversos tipos, incluyendo el telemonitoreo médico, es por ello que se decidió utilizar este controlador (*contikimac_driver*) para evaluar el escenario propuesto.

En conjunto la capa Trama y Capa MAC especifican el protocolo de la capa MAC del estándar IEEE 802.15.4, en la cual se define el mecanismo de acceso al medio que utilizará cada uno de los nodos que conforman la red para utilizar el canal radio.

4.2.3 Capa Trama.

El tipo de trama que se maneja durante la simulación e implementación física es un conjunto de funciones que crean el formato de trama del estándar IEEE 802.15.4, que corresponde a la capa PHY y MAC del estándar 6LoWPAN, en Contiki está definida por el formato “*frame_802154*”.

4.2.4 Capa MAC.

De igual manera que el formato de trama, se eligió la capa MAC que define el estándar de 6LoWPAN, CSMA. En Contiki OS es definida por el controlador *csma_driver*. Sin embargo como ya se mencionó en el capítulo anterior la aplicación de Contiki no se basa en la detección de portadora, ya que el acceso al medio lo realiza la capa RDC, en este caso, “*contikimac_driver*”. Además, se definió un número máximo de retransmisiones para programar el reenvío de los paquetes que fueron mandados al canal radio y no pudieron ser entregados al nodo central durante el intervalo de tiempo t_{RDC} o porque se envió un acuse de recibo y el nodo remitente del paquete no lo escuchó. Este valor corresponde a 4.

A su vez dentro de la capa de red se define una única capa, en la cual se engloban las capas correspondientes a la capa de adaptación, enrutamiento y transporte, especificadas por los protocolos de 6LoWPAN, IPv6-RPL y UDP, respectivamente.

4.2.5 Capa Red.

En este trabajo de tesis únicamente se evaluará el estándar de 6LoWPAN hasta la subcapa de transporte, debido a que hasta el momento no se ha realizado un análisis de una red 6LoWPAN en el que se considere la carga que agrega cada uno de los encabezados de la pila de protocolos del estándar, así como las propiedades que atribuye el “*netstack*” de Contiki al rendimiento de la red.

Para definir esta capa se eligió el controlador “*sicslowpan_driver*”, el cual desempeña las tareas de fragmentación/reensamblado y compresión de encabezado. Además como ya se explicó anteriormente la capa de red de Contiki OS se subdivide en varias subcapas, sin embargo, para esta investigación únicamente es necesario resaltar la subcapa de enrutamiento en donde se define el uso del protocolo RPL y la subcapa de transporte en donde se especifica a UDP como protocolo. Este último se seleccionó de acuerdo a (Ismail et al., 2012), donde se indica que UDP es el protocolo más comúnmente aplicado en redes 6LoWPAN, debido a que tiene un mejor desempeño, eficiencia y menor complejidad que TCP.

4.3 Simulación de la red inalámbrica de sensores utilizando Cooja

Antes de implementar de manera física la arquitectura de la red 6LoWPAN propuesta, se realizó una simulación de dicha red. Para realizar la evaluación de la implementación del estándar 6LoWPAN en un escenario de telemonitoreo multiusuario, haciendo uso de la herramienta de simulación Cooja.

Cooja es un simulador flexible el cual tiene la capacidad de simular no solo software sino hardware requerido. Puede ejecutar el código de la aplicación como si fuera nativo el sistema o ejecutando instrucciones a nivel de emulador del nodo hardware seleccionado. Contiene la pila de protocolos utilizados en el estándar 6LoWPAN permitiendo observar de manera práctica el tráfico de los paquetes, los tiempos utilizados, la ruta tomada y el código software utilizado.

El modelo de simulación quedó definido de la siguiente manera:

En la Figura 27 se muestra el ambiente de la simulación, donde se tienen cinco nodos sensores fijos: cuatro nodos ECG conectados inalámbricamente hacia un nodo raíz, este nodo es el encargado de recibir los datos de cada uno de los sensores. Debido a que Cooja está regido por las bibliotecas y el “*netstack*” de Contiki OS, cada uno de los nodos que componen la red hace uso de estas herramientas, donde se especifican los protocolos y controladores que se utilizan en las capas de Radio, RDC, Trama, MAC y Red, descritas en la sección 4.2.

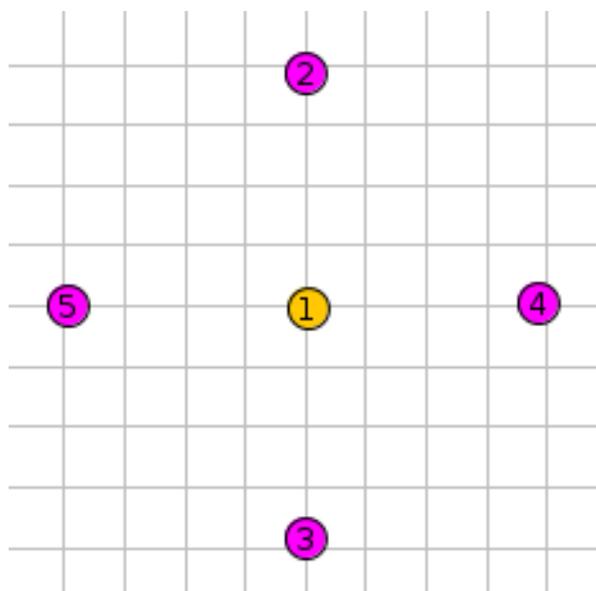


Figura 27. Topología de la simulación de red de 6LoWPAN, en un escenario de telemonitoreo multiusuario.

La simulación tiene un modelo de pérdidas UDGM, con pérdidas en relación a la distancia. Como se muestra en la Figura 28, este tipo de modelo tiene un patrón de pérdidas constantes dentro del área de cobertura (círculo verde) y un patrón de interferencia que se extiende hasta fuera de la misma (círculo gris). Si hay paquetes que se transmitan dentro del área de interferencia entre los paquetes se pierden. En cambio cuando los paquetes se transmiten dentro del área de pérdidas constantes se definen las tasas de éxito en la transmisión y la recepción. En la Tabla 1, se especifican los nodos que se ven afectados por los patrones de interferencia de cada nodo ECG, donde por ejemplo, el nodo ECG con el ID 2 presenta un patrón de interferencia con los nodos ECG con ID 3, 4 y 5.

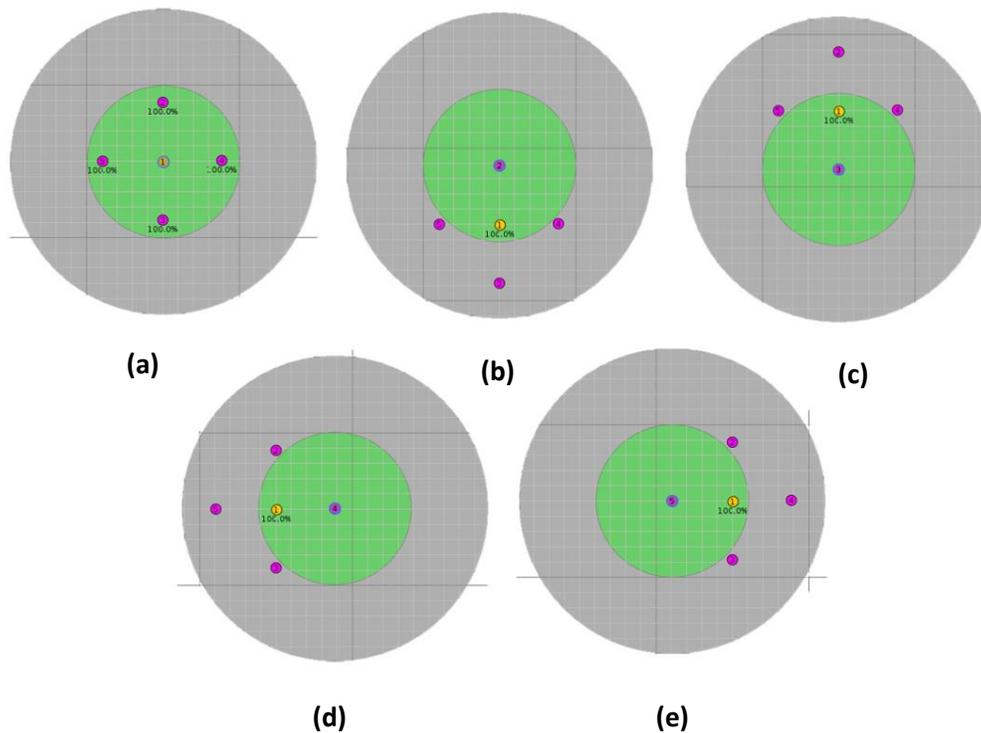


Figura 28. Modelo de pérdidas UDGM- Distance Loss en la red 6LoWPAN propuesta. (a) UDGM-Distance-Loss para nodo central (ID 1). (b) UDGM-Distance-Loss para nodo ECG (ID 2). (c) UDGM-Distance-Loss para nodo ECG (ID 3). (d) UDGM-Distance-Loss para nodo ECG (ID 4). (e) UDGM-Distance-Loss para nodo ECG (ID 5).

Tabla 1. Cobertura de patrón de interferencia de los nodos ECG.

Nodo ECG	Interfiere con:
ID 2	ID 3, ID 4 y ID5
ID 3	ID 2, ID 4 y ID 5
ID 4	ID 2, ID 3 y ID 5
ID 5	ID 2, ID 3 y ID 4

Por medio del patrón de interferencia se igualan las condiciones de los parámetros de la implementación física de TIMAC, que servirá como validación de los resultados obtenidos utilizando Contiki OS. De esta

manera, se asegura que los paquetes 6LoWPAN serán únicamente direccionados por medio de un salto hacia el nodo central (ver sección 0).

Por otro lado, los nodos que componen la red 6LoWPAN propuesta, se encuentran fijos durante toda la simulación. Además, los nodos se inicializan al mismo tiempo, de tal manera que cada nodo al comenzar se asocia con el nodo central. El proceso de asociamiento se ejecuta antes de comenzar a transmitir los paquetes correspondientes a las muestras de ECG. La metodología que se utilizó para asegurar que cada nodo se encuentre asociado con el nodo central, fue por medio de la ventana “*Mote Output*” del simulador Cooja, donde se imprimía un mensaje indicando que este proceso ya había sido ejecutado.

Adicionalmente, como uno de los propósitos de esta investigación es evaluar el envío de paquetes IPv6 sobre el estándar IEEE 802.15.4, en un escenario de telemonitoreo multiusuario, entonces durante la simulación se realizó un barrido de 1 hasta 4 nodos transmitiendo la señal de ECG. De esta manera se podrá verificar cuantos usuarios concurrencios para el monitoreo de ECG se pueden atender. Por otro lado, como ya se mencionó en el Capítulo 2, Contiki OS implementa una capa para el control de encendido y apagado de radio, esto se debe a que este sistema está diseñado para ahorro de energía, es por ello que el radio revisa periódicamente la actividad del canal a una frecuencia específica. En Contiki OS dicha frecuencia puede tomar valores de potencias de dos, desde 2 hasta 64Hz. En este trabajo de tesis se propone evaluar el rendimiento de la red 6LoWPAN propuesta con los valores de frecuencia de revisión del canal de 8, 16,32 y 64 Hz.

La razón por la cual se decidió utilizar estos valores de frecuencias radica principalmente en un aspecto muy importante en las redes inalámbricas de sensores, el ciclo de trabajo. Este concepto consiste en que el transceptor se encienda o apague en función de si el sensor espera recibir o no un algún paquete. Conforme a (Dron, Duquennoy, Voigt, Hachicha, & Garda, 2014), al probar con diferentes valores de frecuencia, con lo cual se varía el ciclo de trabajo del radio, se puede obtener un mejor rendimiento y a su vez una optimización en el consumo de energía. Además de acuerdo a (Dunkels, 2011), los valores de frecuencias por encima de 8 Hz presentan un comportamiento favorable en cuestiones de ahorro en el consumo de energía y la congestión del radio.

Por último el stack de Contiki OS quedó definido de la siguiente manera:

Tabla 2. Parámetros de la simulación.

Parámetro	Valor
Radio	cc2420_driver
RDC	contikimac_driver
Trama	frame_802154
MAC	csma_driver
Red	sicslowpan_driver
No. de nodos	1, 2, 3, 4
Canal	25
Carga útil	84 bytes
Modelo de pérdidas	UDGM- Distance Loss
Tiempo	300 s
No. de iteraciones	5
Topología	Estrella

4.4 Implementación de la red inalámbrica de sensores utilizando Contiki OS

Del mismo modo que en la simulación, se realizó la implementación física de la red 6LoWPAN propuesta en un escenario de telemonitoreo, para transportar los datos de la variable fisiológica ECG. Para cumplir con el objetivo general de esta investigación que corresponde a la evaluación del estándar 6LoWAN, se cargó manualmente el stack de 6LoWPAN que proporciona Contiki OS en cada uno de los nodos de la red y se utilizaron herramientas para el monitoreo de los paquetes en el canal.

De igual que en las consideraciones de la simulación inalámbrica de sensores utilizando Cooja, los experimentos de la implementación física los nodos se inicializaron al mismo tiempo, inmediatamente se ejecuta el proceso de asociación con el nodo central. Para comprobar que cada uno de los nodos se asoció exitosamente con el nodo central, se imprime un mensaje indicando la realización de este proceso. Cabe mencionar que cada uno de los nodos ECG se encuentran conectados vía puerto serial hacia la computadora, de tal manera que el mensaje de la ejecución del proceso de asociamiento pudo ser visualizado en la ventana de comandos.

A continuación se describe el equipo utilizado y la metodología seguida para la creación de la red 6LoWPAN.

4.4.1 Equipo utilizado

Una parte fundamental de esta investigación fue la elección de una plataforma de comunicación 6LoWPAN, para la implementación del estándar 6LoWPAN en cada uno de los nodos que conforman el escenario de telemonitoreo propuesto. La elección de la plataforma se realizó de acuerdo a una serie de especificaciones, tales como:

- Posibilidad de cargar la pila de protocolos del estándar 6LoWPAN de manera manual.
- Plataforma soportada por Contiki OS.
- Disponibilidad del producto en el mercado.
- Documentación hardware y software suficiente.
- Comunicación serie disponible por medio de una interfaz UART.
- Antena incorporada a una frecuencia de trabajo de 2.4 GHz.

A partir de estas características se optó por utilizar la plataforma SensorTag Development Kit⁸, debido a que además cumple con las especificaciones antes descritas, se tiene experiencia con el manejo de esta tecnología. Enseguida se describe brevemente el equipo utilizado para la implementación de la red 6LoWPAN en un escenario de telemonitoreo, sus características, alcances y trabajo realizado con cada uno de ellos.

4.4.1.1 Texas Instruments CC2650STK

El dispositivo CC2650STK, también conocido como SensorTag, es un kit de desarrollo que tiene disponibles diversas opciones en términos de herramientas y pilas de protocolos (Bluetooth, ZigBee o 6LoWPAN). Por lo tanto permite la integración de una pila de protocolos de manera manual a través de una actualización de firmware. Provee comunicación vía puerto serial por medio de Debug DevPack XDS110 TI. Para su programación se utiliza Contiki OS (Lenguaje C).

⁸ <http://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG>

Cuenta con la implementación de 10 sensores, los cuáles son: luz, micrófono digital, sensor magnético, humedad, presión, acelerómetro, giróscopo, magnetómetro, temperatura de un objeto y temperatura ambiental.

Trabajo realizado: Se establecieron 4 nodos como dispositivos “*end-device*” y se desarrolló código para habilitar la transmisión de un mensaje 6LoWPAN correspondiente a 42 muestras de ECG cada 46.667 ms (ver Figura 26). Además se estableció un nodo central, encargado de recolectar todos los paquetes enviados por los nodos sensores, de igual manera se desarrolló código para iniciar con el establecimiento de la red 6LoWPAN. Así mismo se programó uno de los botones de cada sensor, para controlar la inicialización de cada uno de ellos.

4.4.1.2 Debug DevPack XDS110 TI

Es un dispositivo USB fabricado por Texas Instruments que tiene la capacidad de cargar aplicaciones desarrolladas por el usuario en el SensorTag y posteriormente depurarlas. Para ello requiere del apoyo de otras herramientas en software, como SmartRF Flash Programmer 2 o Code Composer Studio, ambos de Texas Instruments e IAR Embedded Workbench, que también soporta esta plataforma.

Por otro lado la depuración realizada por el DevPack suministra energía al SensorTag mediante una interfaz USB JTAG, lo que significa que también es una solución práctica para aplicaciones que requieren una alimentación permanente. En caso de que los 10 sensores proporcionados por el SensorTag no sean suficientes, o se desee agregar algunas salidas personalizadas, este dispositivo cuenta con un puerto GPIO.

4.4.1.3 Texas Instruments CC2531EMK

Es un módulo de evaluación que provee un Dongle USB CC2531 y soporte para una interfaz en la PC de aplicaciones de 802.15.4/ZigBee. El Dongle permite capturar los paquetes intercambiados en una red IEEE 802.15.4 y visualizarlos directamente en la PC.

Trabajo realizado. Se trabajó en la implantación de un sniffer 6LoWPAN basado en el Dongle USB CC2531 para contar con una herramienta que capture los paquetes que se intercambian dentro de la red 6LoWPAN propuesta y de esta manera corroborar los resultados obtenidos. El sniffer está compuesto por un túnel

desde el puerto USB de una PC hacia Wireshark⁹ por medio de una aplicación desarrollada en Python (ver Figura 29) (Dodd, 2015).

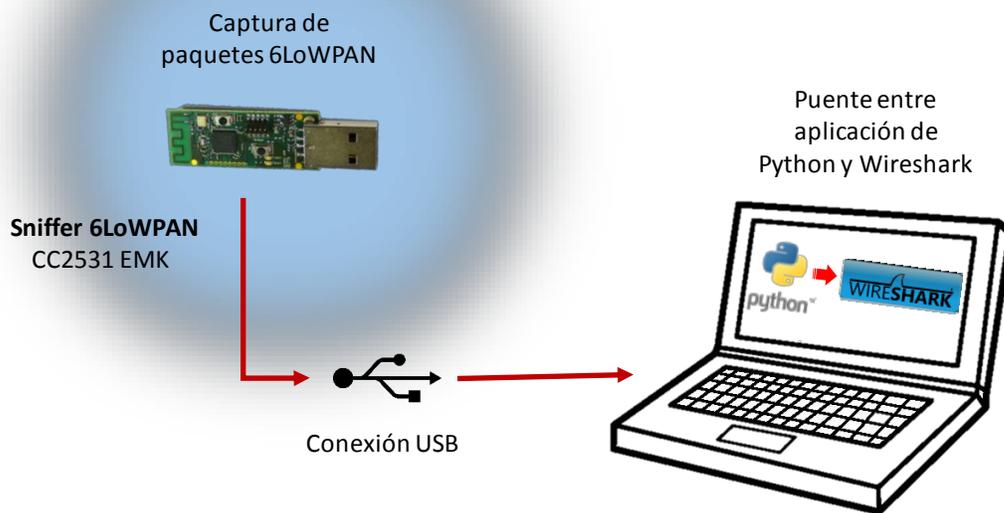


Figura 29. Escenario para la implementación del sniffer 6LoWPAN.

El funcionamiento del sniffer 6LoWPAN utiliza una aplicación desarrollada en Python, por medio de la cual los paquetes capturados por el dispositivo CC2531EMK, se adaptan a una versión de una captura de Wireshark. De tal manera que los paquetes 6LoWPAN pueden ser desglosados en base a los protocolos que lo componen (IEEE 802.15.4, IPv6, UDP) observando los datos en formato hexadecimal.

4.4.2 Descripción del escenario de prueba experimental

Para la implementación física de la red 6LoWPAN propuesta se utilizaron los SensorTags, cuatro de ellos actúan como nodos que emulan la carga de 14 muestras de cada uno de los canales de ECG. La información de cada sensor va dirigida hacia un nodo central (SensorTag), que actúa como único resumidero de información. Este nodo se encarga de capturar cada uno de los paquetes que provienen de los otros sensores.

⁹ Wireshark, es un analizador de protocolos, que permite el monitoreo de datos de una red en tiempo real y además brinda una gran variedad de filtros para organizar la información. Está distribuido bajo licencia GPL de GNU.

En la Figura 30 se muestra el escenario experimental de la red 6LoWPAN propuesta con el equipo utilizado para su formación.

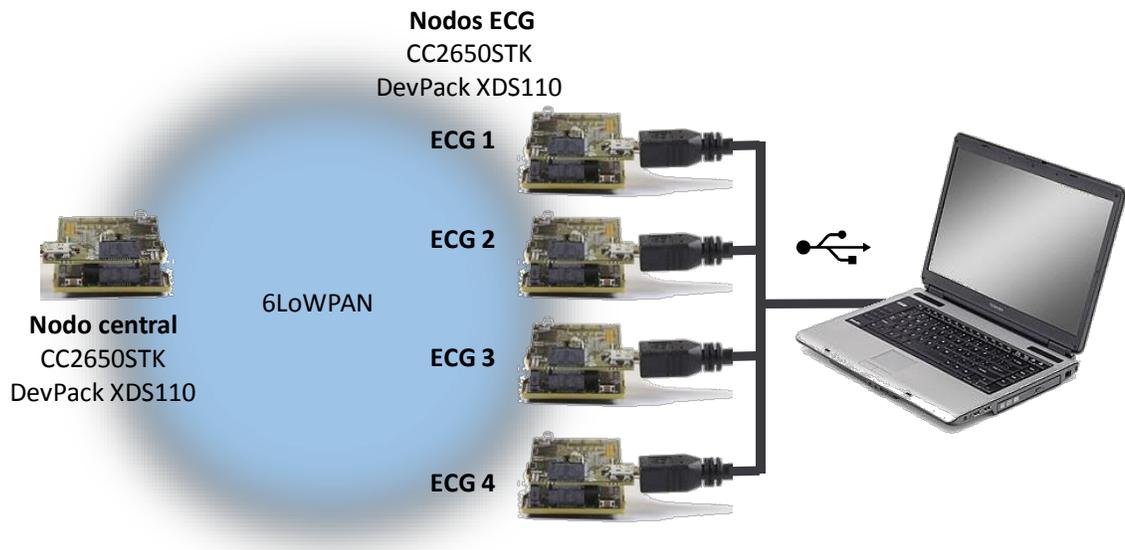


Figura 30. Escenario experimental de la red 6LoWPAN propuesta en Contiki OS.

El funcionamiento de todos los nodos de la red 6LoWPAN, es dictado por un código que especifica el “*netstack*” de 6LoWPAN en Contiki OS, hasta el nivel de la capa de transporte (Ver Figura 17). A excepción del nodo resumidero, todos los nodos ECG ejecutan el diagrama a bloques que se muestra en la Figura 31. Como se puede observar en primera instancia se asigna la dirección global del nodo, y se establece una conexión vía UDP (sección 2.4.2.5). Enseguida se inicializa el reloj SEND_TIME, con una duración de 10 segundos, este periodo es el necesario para permitir que el nodo central forme la DODAG y se asignen las direcciones correspondientes a los nodos que formaran la red. Una vez que expira SEND_TIME, se inicializa el reloj INTERVAL_TIME, encargado de controlar el intervalo en que se estará enviando el paquete predeterminado con las 14 muestras de los 3 canales de ECG. Sin embargo, antes de enviar el paquete de datos, el nodo se asegura de tener alcance hacia el nodo central, es decir, que se encuentre dentro del rango de cobertura definido en el modelo de propagación UDGM-Distance-Loss. En caso de estar dentro del rango de cobertura, el paquete es transmitido, en caso contrario, el nodo reporta que no se pudo establecer una conexión y reporta un error.

Por otro lado el nodo central es regido por un código fuente que se encarga de construir la DODAG mediante el intercambio de mensajes de control, este proceso de asociación entre los nodos, se lleva a cabo durante los 10 segundos establecidos por el reloj SEND_TIME (definido en el código ejecutado por los nodos ECG). Al mismo tiempo se ejecuta el proceso para el establecimiento de una conexión UDP, en

este proceso el nodo central se encuentra alerta para la recepción de los paquetes de datos que envían los nodos asociados. Cuando un paquete fue recibido por el nodo central es procesado y notificado en pantalla en donde se indica la dirección fuente del mensaje recibido, los puertos UDP de entrada y salida así como la longitud y los datos del mensaje recibido. El diagrama a bloques de este proceso se puede observar en la Figura 32.

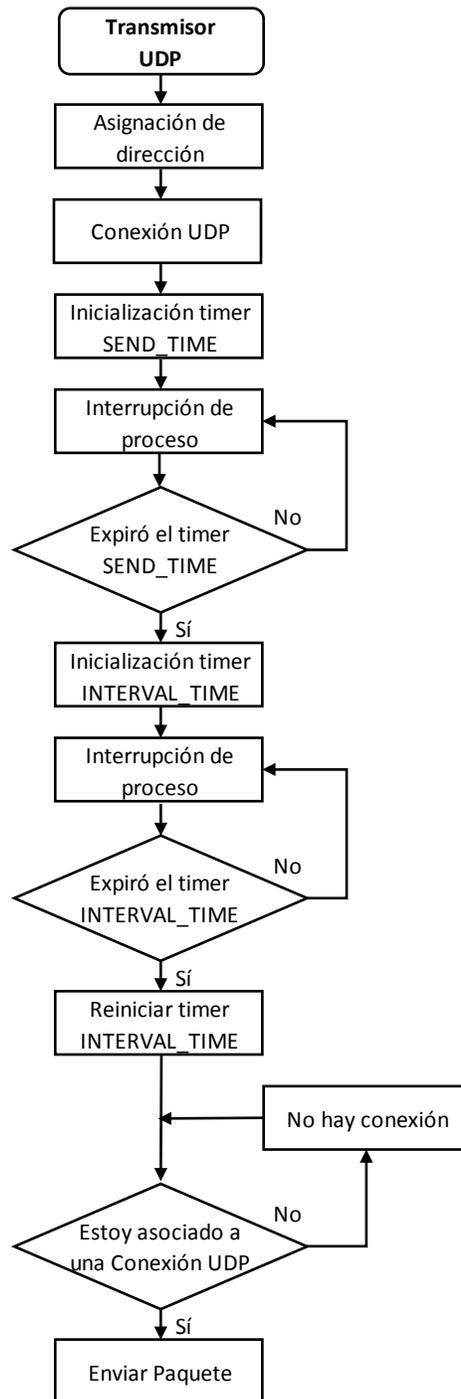


Figura 31. Diagrama de flujo del algoritmo que ejecutan los nodos ECG del escenario propuesto.

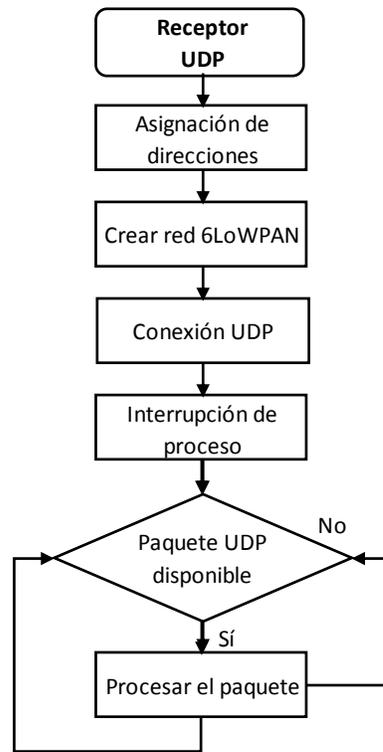


Figura 32. Diagrama de flujo del algoritmo que ejecuta el nodo central del escenario propuesto.

En la Figura 33 se muestran los diagramas de flujo de los mecanismos de acceso al medio CSMA unificados con la capa RDC ContikiMAC que se ejecutan en los nodos ECG y el nodo central. En ellos se resaltan los estados y aspectos más importantes de cada uno, tales como, sensado CCA, transmisión y recepción de paquetes, confirmación de recibido, etc. En la Figura 33 (a) se muestra el diagrama correspondiente al nodo central que al inicializarse el nodo se enciende y realiza una evaluación del canal (CCA) para conocer la actividad que se tiene en el canal. Considerando que es la primera vez que el nodo central se enciende y se detecta que el canal está libre, entonces el radio se mantiene en un estado inactivo y comienza el proceso del algoritmo de la Figura 32, enseguida el nodo pasa a un estado activo y nuevamente ejecuta el CCA para verificar si el canal se encuentra ocupado, de ser pasa a un estado de interrupción para procesar el paquete. Caso contrario el nodo se apaga y después de un tiempo correspondiente a $\frac{1}{f_{RDC}}$ el radio se vuelve a encender (donde f_{RDC} es la frecuencia con la que se enciende y apaga el radio).

Por otro lado en la Figura 33 (b) se tiene el diagrama correspondiente a los nodos transmisores, en este caso los nodos ECG, que de igual manera que el nodo central al inicializarse por primera vez permanecen encendidos y ejecutan el CCA, así mismo pasan a un estado inactivo para continuar con el proceso del algoritmo de la Figura 31. El envío del paquete en cada nodo se lleva a cabo siempre y cuando se encuentre el canal libre y de esta manera evitar las colisiones, además para continuar con la transmisión del siguiente

operación similar, en este trabajo se emula la carga de los encabezados IPv6 y UDP, dentro de la misma carga útil de la MSDU (104 bytes) del formato de la MPDU (127 bytes) (Ver Figura 34).

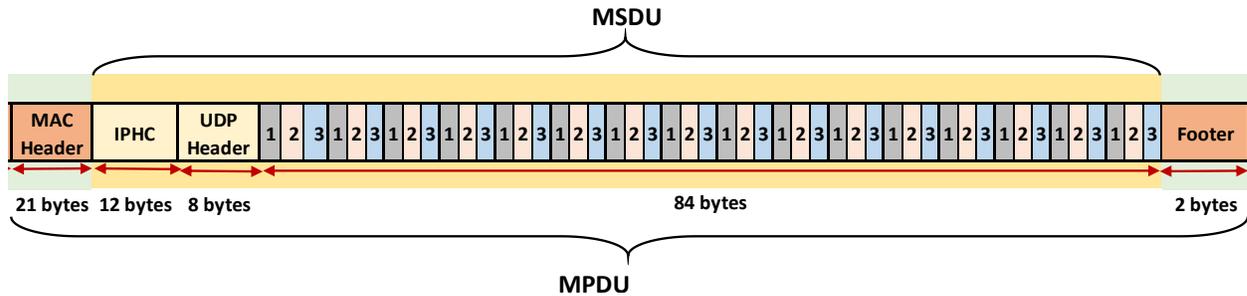


Figura 34. Formato de paquete utilizado en la implementación de TIMAC.

4.5.1 Equipo utilizado

Del mismo modo que en la simulación e implementación de Contiki OS, se realizó la implementación física de la red 6LoWPAN propuesta en un escenario de telemonitoreo (Ver sección 4.4.2), para transportar los datos de la variable fisiológica ECG. El equipo que se utilizó tanto para los nodos ECG y el nodo central fueron los motes CC2650STK, los mismos que se emplearon en la implementación usando Contiki OS. Enseguida se describe brevemente, el alcance y trabajo realizado con cada uno de los nodos.

4.5.1.1 Nodos ECG

Se establecieron 4 nodos como dispositivos RFD y se modificaron algunos parámetros de las librerías proporcionadas por TIMAC para habilitar la transmisión de un mensaje con las características de un paquete 6LoWPAN (Ver Figura 34) cada cierto intervalo de tiempo al presionar un botón. El mensaje predeterminado tiene una carga útil 104 bytes y se enviará cada 43.667 ms para emular el tráfico de los encabezados IPHC, UDP y los datos de las muestras de ECG.

4.5.1.2 Nodo central

Se estableció un nodo central como dispositivo FFD, encargado de recolectar todos los paquetes enviados por los nodos sensores, de igual manera se agregó y modificó código para generar un paquete ACK (Ver Figura 35) con el mismo formato que se utiliza en la implementación de Contiki OS. El paquete ACK, es enviado por el nodo central hacia el nodo ECG correspondiente, indicando que se ha recibido un paquete.

Este paquete consta de 5 bytes, en donde se especifican los campos *“Frame Control Field”*, *“Sequence Number”* y *“Frame Check Sequence”*. El campo *“Sequence Number”*, es muy importante debido a que por

medio de éste se puede identificar cuál fue el paquete que fue recibido por el nodo central. Es importante mencionar que el proceso de asociamiento de la misma manera en que se realiza en la implementación de Contiki OS.

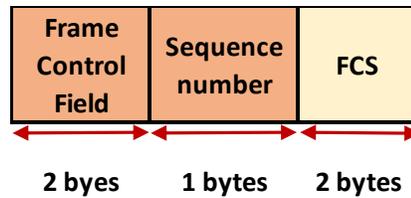


Figura 35. Formato de ACK en IEEE 802.15.4 utilizado en TIMAC.

4.5.2 Descripción del escenario de prueba experimental

Para realizar la implementación física de la red 6LoWPAN propuesta se utilizaron como motes los SensorTags, que se describieron en la Secciones 4.5.1.1 y 4.5.1.2. En la Figura 36 se muestra el escenario experimental, donde cuatro de los sensores ejercen el papel de nodos ECG, cuya información va dirigida hacia un nodo central, que actúa como único resumidero de información. El tráfico que se presenta por esta red es capturado por un sniffer 6LoWPAN, que fue descrito en la Sección 4.4.1.3 utilizando el dispositivo CC2531 EMK.

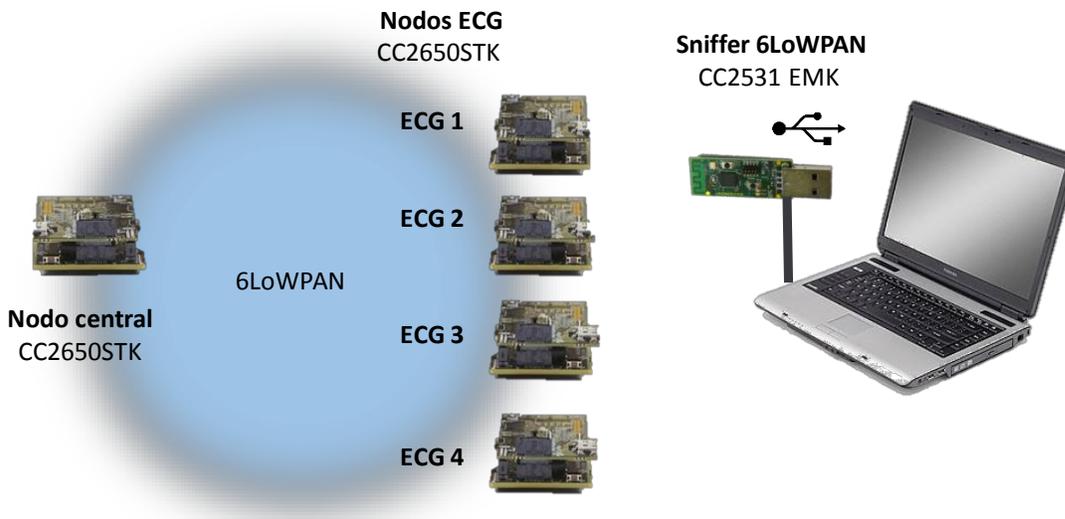


Figura 36. Escenario experimental de la red 6LoWPAN propuesta en TIMAC.

El funcionamiento de todos los nodos de la red 6LoWPAN, está establecido por un código que se especifica en la pila de protocolos del estándar IEEE 802.15.4 en TIMAC, hasta el nivel de capa MAC. No obstante, como se explicó anteriormente se emularon las cargas de los encabezados IPHC y UDP, para considerar la sobrecarga de los encabezados de las capas superiores a la capa MAC IEEE 802.15.4.

A excepción del nodo resumidero, todos los nodos ECG ejecutan el diagrama a bloques que se muestra en la Figura 37 (a). Como se puede observar en los primeros procesos que se ejecutan es la asignación de direcciones y el asociamiento entre los nodos que componen la red. Este proceso de asociamiento consta del intercambio de solicitudes (“request”) y confirmaciones (“responses”) entre los nodos ECG y el nodo central. Una vez que los nodos se encuentran asociados al nodo central, se envía un paquete IEEE 802.15.4 (Ver Figura 34), si éste está disponible. Enseguida se espera un tiempo correspondiente a $\frac{1}{f_{RDC}}$ ms, para enviar el siguiente paquete.

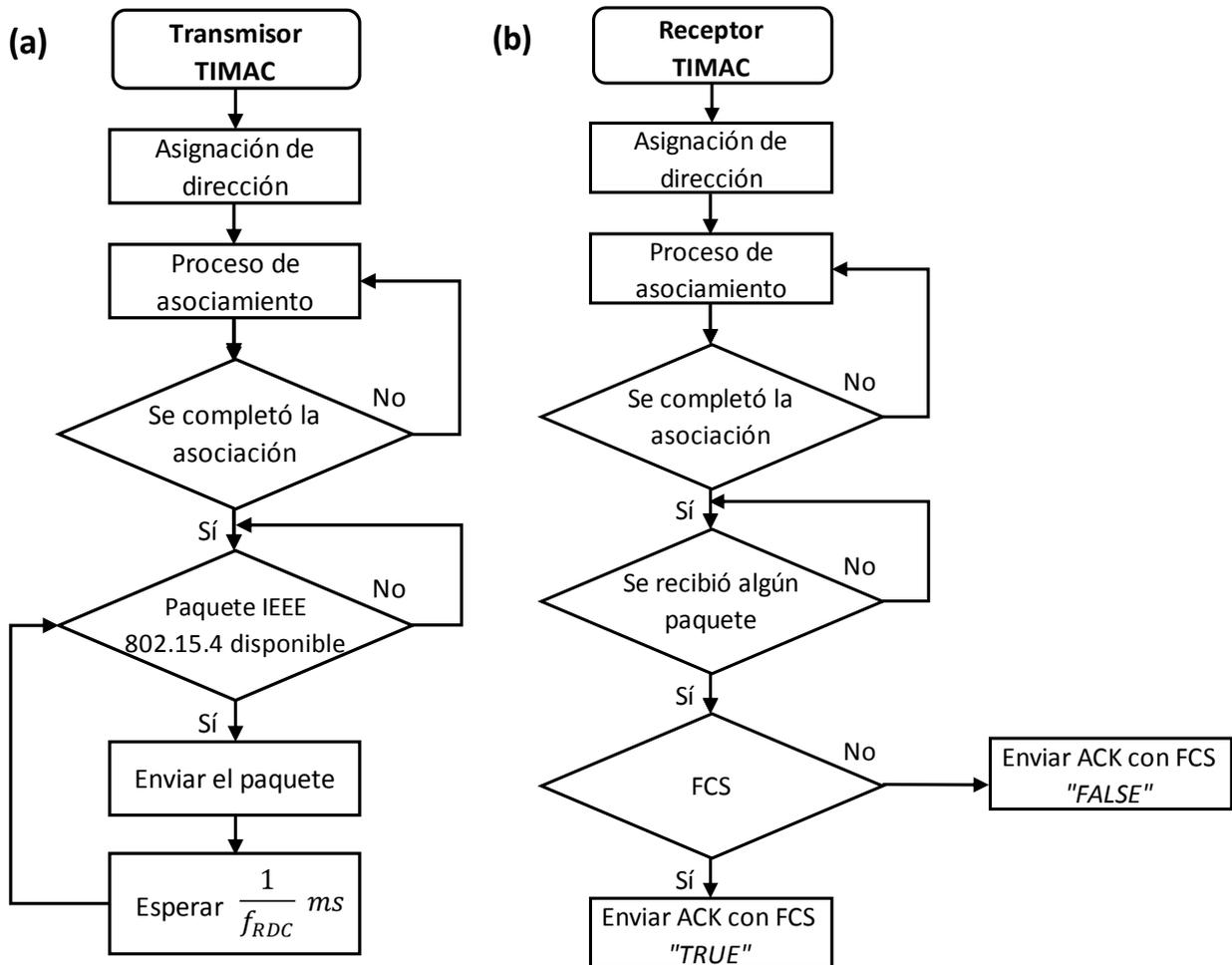


Figura 37. Diagrama de flujo de transmisión y recepción en TIMAC. (a) Diagrama de flujo de los nodos ECG en TIMAC. (b) Diagrama de flujo del nodo central en TIMAC.

Por otro lado en la Figura 37 (b), se ilustra el diagrama correspondiente al nodo central. En primera instancia, se ejecutan los procesos de asignación de dirección y asociamiento. Es importante mencionar que estos procesos se ejecutan conjuntamente al inicializar los nodos ECG y el nodo central. Enseguida, el nodo central, se encuentra listo para la recepción de paquetes provenientes de los nodos ECG. Al recibir un paquete, primero efectúa un mecanismo de FCS, para revisar que la información recibida coincida con

la información que fue transmitida, de ser esto cierto, entonces, se envía un paquete de confirmación ACK con un FCS "TRUE"; caso contrario se devuelve un ACK con un FCS "FALSE", indicando al nodo ECG destinatario que el paquete con el paquete debe de ser retransmitido.

Con los diagramas de flujo que se explicaron anteriormente, se puede inferir que el desempeño de la red será muy diferente al que se tendrá con Contiki OS. Debido a que utilizando TIMAC, no se tiene la implementación de una capa RDC que controla el encendido del radio y que a su vez, controla el acceso al medio de los nodos ECG. De tal manera, que la evaluación de la implementación del estándar 6LoWPAN utilizando TIMAC, servirá como apoyo y validación de los resultados que se obtengan de la implementación utilizando Contiki OS.

A continuación en el siguiente capítulo se mostrarán los resultados obtenidos en este trabajo de tesis, correspondientes, a la simulación e implementación de la red 6LoWPAN propuesta. Así como también se ilustrará los resultados que pueden alcanzarse con la implementación de la misma red utilizando TIMAC, como pila de protocolos.

Capítulo 5. Resultados de simulación y experimentación

5.1 Introducción

En esta sección se mostrarán la prueba de funcionamiento y los resultados obtenidos en la simulación e implementación del estándar 6LoWPAN de acuerdo a las arquitecturas propuestas en la Figura 27 y la Figura 30, respectivamente. Con el fin de evaluar el desempeño de la red para un escenario en el que se cuenta con un nodo central y de 1 a 4 nodos sensores que monitorean la señal ECG de un paciente los nodos “cc2420_driver” (en el simulador Cooja) y dispositivos CC2650STK (en la implementación física) emulan el tráfico de esta variable.

Como ya se ha mencionado en los capítulos anteriores, la capa RDC es una capa muy relevante en la evaluación de la implementación del estándar 6LoWPAN, debido a que es una capa que añade específicamente el sistema operativo de Contiki OS. Es por esta razón, que a lo largo del capítulo se mostrarán los resultados que se obtuvieron considerando diferentes valores de la f_{RDC} para así poder evaluar holísticamente tanto la simulación como implementación utilizando Contiki OS. De tal manera que se pueda elegir que tasa de encendido/apagado del radio es la adecuada para una aplicación de telemonitoreo médico o en su debido caso si la presencia de la misma provoca comportamientos desfavorables para los fines.

El desempeño de la implementación de dicha red 6LoWPAN será evaluado con base a métricas de pérdida de paquetes y retransmisiones realizadas. A continuación se describen brevemente las métricas que serán utilizadas para realizar la evaluación de la misma.

5.2 Definición de métricas a evaluar.

Para evaluar el desempeño de la implementación del estándar 6LoWPAN se consideraron algunas métricas, las cuales fueron calculadas y cuantificadas en el nodo receptor de la red (nodo central). A continuación se describen brevemente:

- **Paquetes transmitidos (P_{TX}).** Se considera a un paquete transmitido cuando un paquete con una longitud de 133 bytes ha sido mandado al canal radio, con direccionamiento hacia el nodo central. Es decir, este tipo de paquetes cuentan con los encabezados correspondientes a los protocolos IEEE 802.15.4 PHY, 802.15.4 MAC, IPv6 y UDP, además de la carga útil de 84 bytes. Está

compuesto por la suma de paquetes transmitidos con éxito y sin éxito, calculado con la ecuación (3).

$$P_{TX} = P_{success} + P_{unsuccess} \quad (3)$$

- **Paquetes transmitidos con éxito ($P_{success}$).** En este trabajo de tesis, este tipo de paquetes son aquellos que son recibidos por el nodo central, es decir, el nodo central envía un acuse de recibo al nodo ECG remitente, dentro de un intervalo de tiempo menor o igual a t_{RDC} , definido por la ecuación (1). En caso, de que el nodo ECG remitente, no escuche el paquete ACK que le envió el nodo central, éste paquete pasa a ser procesado por la capa MAC para reprogramar su retransmisión, hasta exceder el número de retransmisiones definidas por la capa MAC, ocurriendo esto, entonces el paquete se considera sin éxito.
- **Paquetes transmitidos sin éxito ($P_{unsuccess}$).** En esta investigación, se considera un paquete transmitido sin éxito, cuando éste es descartado, debido a que el nodo remitente no recibe un paquete ACK, con base a lo mencionado en la sección 2.5.1.2. En otras palabras, aunque el nodo central envíe un paquete ACK de confirmación, sí el nodo ECG receptor no lo escucha, entonces pasa a la capa MAC para reprogramar su retransmisión, hasta exceder el número de retransmisiones definidas por ésta capa. Sí esto último se cumple, entonces se considera una transmisión sin éxito.
- **Retransmisiones realizadas (R_{TX}).** El número de retransmisiones realizadas se refiere a la cantidad de retransmisiones necesarias (por parte de los nodos ECG) para que un paquete sea recibido por el nodo central. Tomando en cuenta que el nodo ECG remitente reciba también el paquete ACK. Por lo tanto, ésta métrica está basada en las retransmisiones realizadas dentro del intervalo de tiempo t_{RDC} , es decir, únicamente se cuantifican las retransmisiones efectuadas por la capa RDC y no por la capa MAC.
- **Retransmisiones promedio realizadas ($\overline{R_{TX}}$).** Está basada en las retransmisiones realizadas y muestra el promedio de retransmisiones efectuadas por el conjunto de nodos que componen la red 6LoWPAN para transmitir un paquete, es decir, es el promedio de retransmisiones a nivel de capa RDC por paquete transmitido. Ya que en caso de que no se envíe un ACK dentro del intervalo t_{RDC} hacia el nodo remitente o que éste no sea escuchado, el paquete es procesado por la capa MAC para reprogramar su próxima retransmisión.

- **Retransmisiones promedio por nodo.** Esta métrica indica el promedio de retransmisiones que realiza cada nodo ECG (independientemente) hasta nivel de capa RDC para que un paquete 6LoWPAN sea recibido por el nodo central. En caso de que no se envíe un ACK dentro del intervalo t_{RDC} hacia el nodo remitente o que éste no sea escuchado, el paquete es procesado por la capa MAC para reprogramar su próxima retransmisión.
- **Caudal eficaz (“Throughput”).** Es una medida que indica la cantidad de tráfico que es recibido exitosamente por unidad de tiempo (bps). Con base en esta métrica es posible determinar la eficiencia de la red durante un determinado periodo. Cabe mencionar que para calcular el caudal eficaz se considera el total de tráfico recibido por el nodo central proveniente de la red entera, como se observa en la ecuación (4), donde N_{bpp} corresponde a los bits de carga útil por paquete.

$$Throughput = \frac{P_{TX} \cdot N_{bpp}}{\text{tiempo de simulación}} \quad (4)$$

Estas métricas, ayudarán a valorar el comportamiento que tiene el “*netstack*” de Contiki OS, específicamente de las características que agrega la capa RDC, con base a la variación de la f_{RDC} que se defina para cada uno de los escenarios.

Por otro lado, los valores que se analizaron para la f_{RDC} en esta investigación fueron 8, 16, 32 y 64 Hz, debido a que este parámetro está establecido en múltiplos de potencias de 2. La razón por la cual se decidió utilizar la frecuencia de 8 Hz como f_{RDC} mínima se debe a que para valores de frecuencia menores a esta cota, los radios se ven en la necesidad de realizar un alto número de retransmisiones para enviar un paquete y que este sea recibido, por lo cual reducir la frecuencia a valores por debajo de 8 Hz causaría que el número de retransmisiones aumente gradualmente. La Figura 38 y la Figura 39 muestran gráficamente el comportamiento que tiene un nodo con una frecuencia de 8 Hz, durante la simulación e implementación física, respectivamente. Como se puede observar el nodo 2 (nodo ECG) tiene que realizar 19 retransmisiones (para ambos casos) para que el nodo 1 (nodo central) reciba el paquete y devuelva un ACK como acuse de recibido.

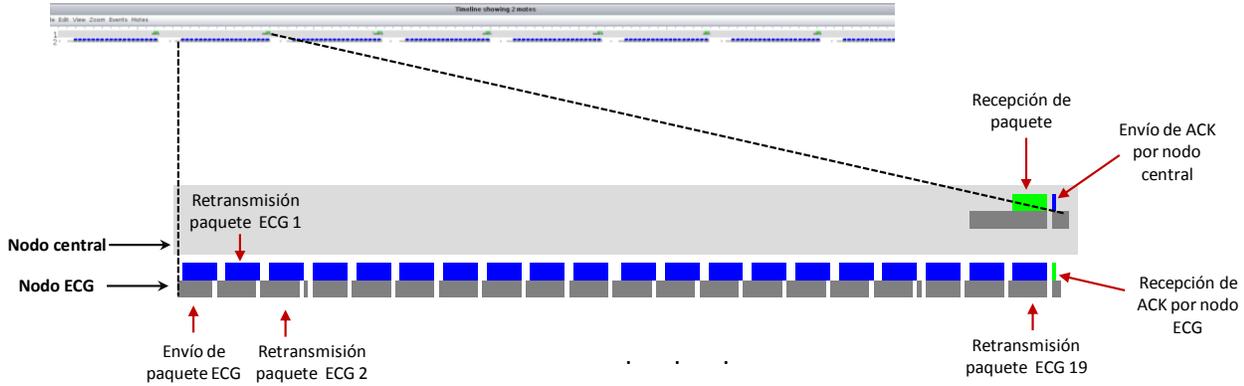


Figura 38. Línea de tiempo en Cooja para nodos con $f_{RDC} = 8 \text{ Hz}$.

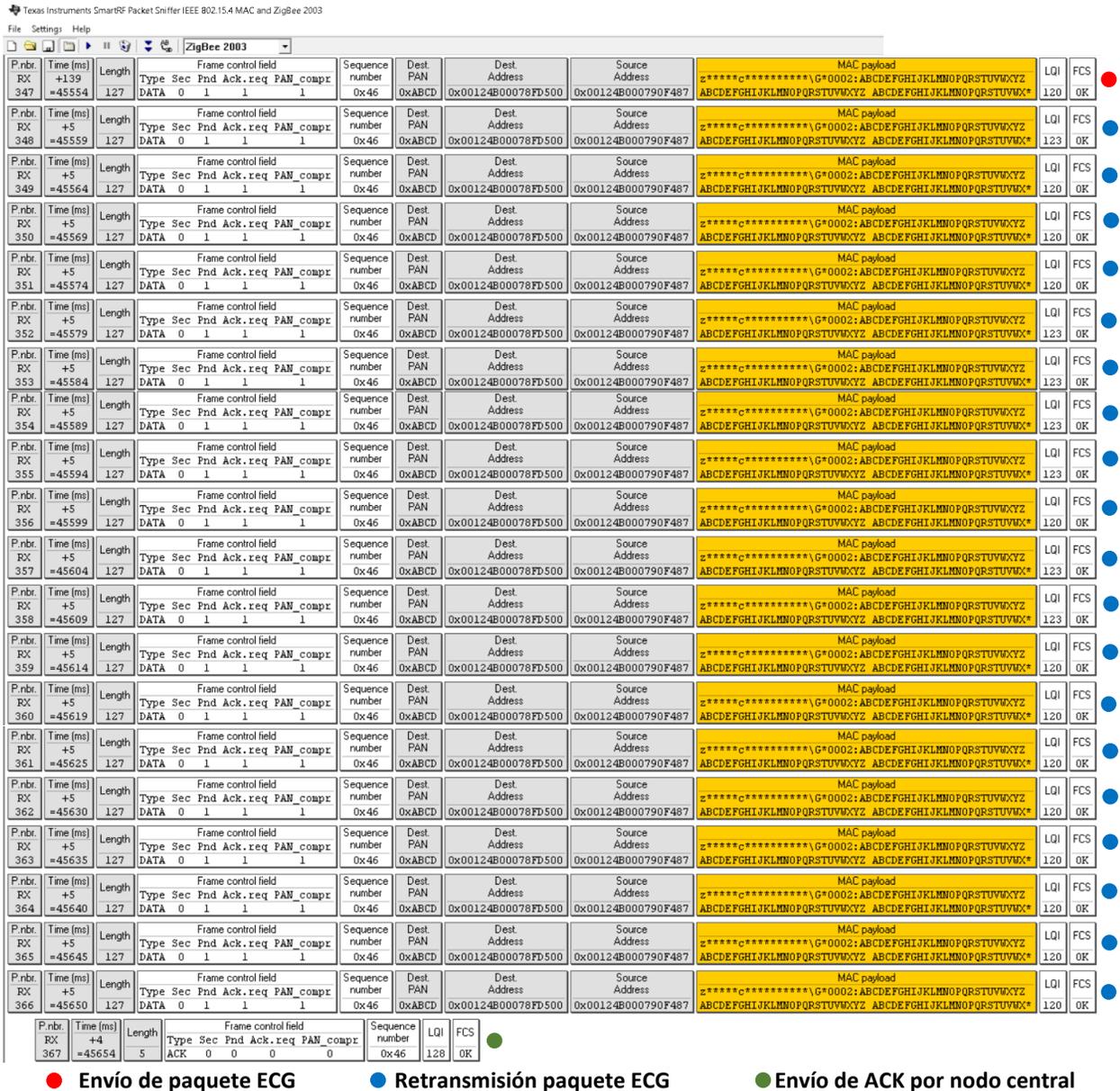


Figura 39. Captura sniffer 6LoWPAN para una $f_{RDC} = 8 \text{ Hz}$.

Por otro lado $f_{RDC} = 64 \text{ Hz}$ se estableció como frecuencia máxima a causa de que si se aumentaba a frecuencias mayores los nodos ECG estarían pasando de estado activo a inactivo en un lapso de tiempo muy corto, esto ocasiona que cuando los nodos se inician no se de oportunidad para que se realice el proceso de asociamiento y se establezca la comunicación UDP, a causa de esta transición inmediata de un estado a otro. Adicionalmente con el uso de esta frecuencia RDC se logró tener el menor número de retransmisiones necesarias para la recepción de un paquete, tal y como se muestra en la Figura 40 y Figura 41, donde se tiene un nodo ECG y un nodo central con una frecuencia de 64 Hz y se realizan 3 retransmisiones para que el nodo central detecte que se le está enviando un paquete, lo procese y envíe el ACK de recibido.

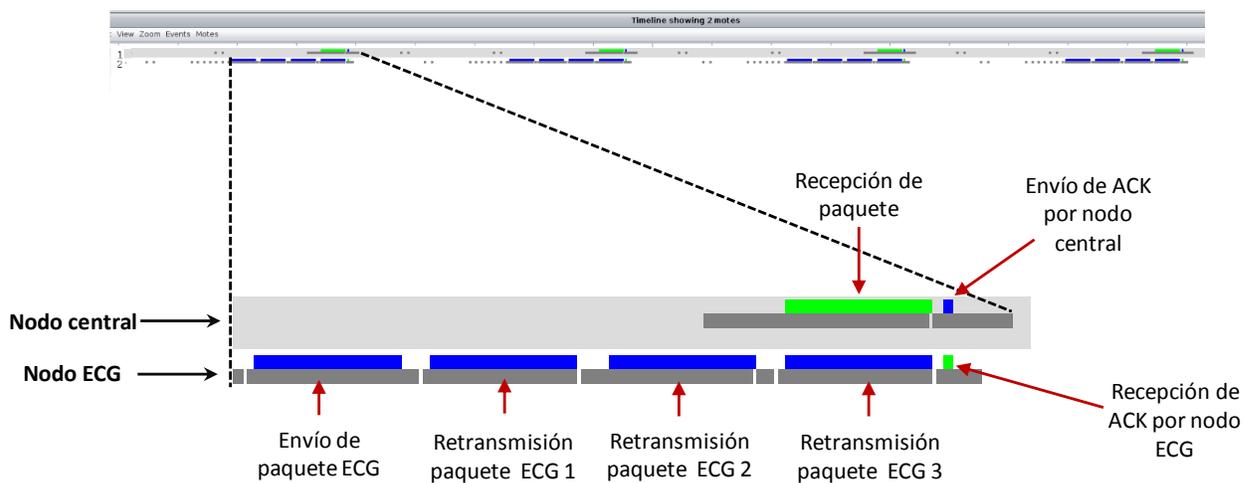


Figura 40. Línea de tiempo en Cooja para nodos con $f_{RDC} = 64 \text{ Hz}$.

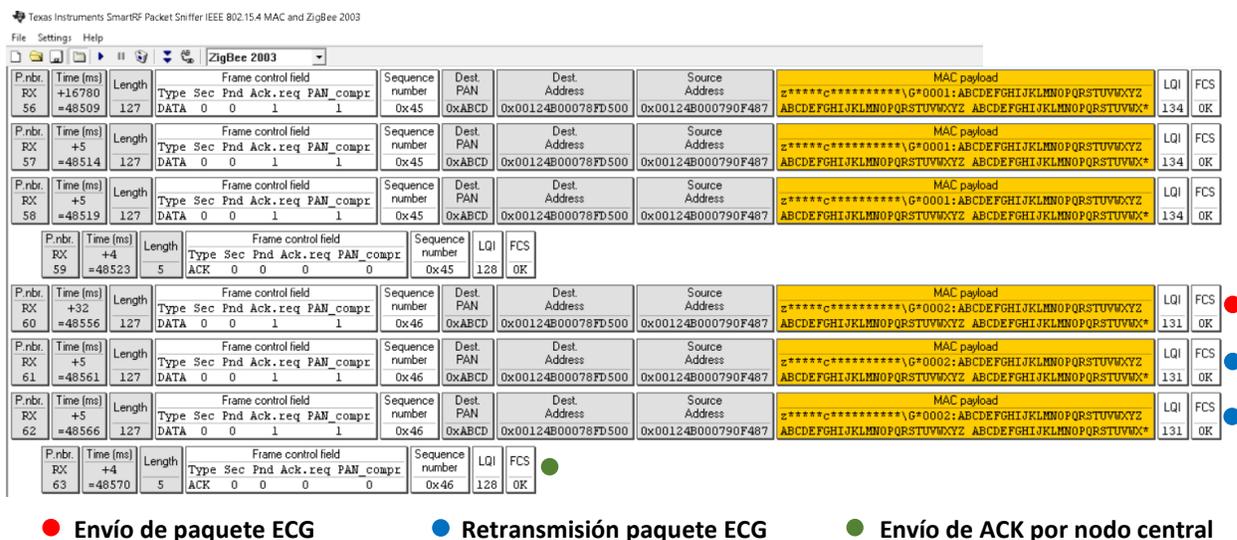


Figura 41. Captura sniffer 6LoWPAN para una $f_{RDC} = 64 \text{ Hz}$.

5.3 Validación de la implementación

Antes de proceder a la evaluación de los datos obtenidos durante el proceso de simulación e implementación del estándar 6LoWPAN utilizando Contiki OS, se realizó una prueba de funcionamiento para garantizar que la red y el “*netstack*” que se propone evaluar estuvieran ejecutándose de manera correcta. Esto consistió en utilizar las herramientas Wireshark y Sniffer 6LoWPAN para analizar mediante capturas el tráfico de paquetes intercambiados entre el nodo central y los nodos ECG.

5.3.1 Pila de protocolos

La Figura 42 muestra el formato del paquete 6LoWPAN enviado desde un nodo ECG hacia el nodo central definido anteriormente (ver Figura 16). Se puede comprobar que efectivamente se tienen presentes los encabezados definidos por el estándar 6LoWPAN hasta la capa de transporte UDP proporcionando una carga útil de 84 bytes en donde se emula los bytes que proporcionan las 14 muestras de cada canal de ECG.

No.	Time	Source	Destination	Protocol	Length	Info
238	1.033000	::212:7402:2:202	::212:7401:1:101	UDP	127	1234 → 1234 Len=84
239	1.033000	::212:7402:2:202	::212:7401:1:101	UDP	127	1234 → 1234 Len=84
240	1.034000	::212:7402:2:202	::212:7401:1:101	UDP	127	1234 → 1234 Len=84

```

> Frame 238: 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits)
  > IEEE 802.15.4 Data, Dst: NitLab_01:00:01:01:01, Src: NitLab_02:00:02:02:02
    > Frame Control Field: 0xdc61, Frame Type: Data, Acknowledge Request, Intra-PAN, Destination Addressing
      Sequence Number: 128
      Destination PAN: 0xabcd
      Destination: NitLab_01:00:01:01:01 (00:12:74:01:00:01:01:01)
      Extended Source: NitLab_02:00:02:02:02 (00:12:74:02:00:02:02:02)
    > Frame Check Sequence (TI CC24xx format): FCS OK
  > 6LoWPAN
    > IPHC Header
      Next header: IPv6 Hop-by-Hop Option (0x00)
      Source: ::212:7402:2:202
      Destination: ::212:7401:1:101
    > User Datagram Protocol, Src Port: 1234 (1234), Dst Port: 1234 (1234)
    > Data (84 bytes)
  <
    61 dc 80 cd ab 01 01 01 00 01 74 12 00 02 02 02 a.....t.....
    00 02 74 12 00 7a f7 00 00 11 00 63 04 00 1e 02 ..t..z..c....
    00 04 d2 04 d2 00 5c cb 65 30 30 30 32 3a 41 42 .....e0002:AB
    43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 CDEFGHIJ KLMNOPQR
    53 54 55 56 57 58 59 5a 20 41 42 43 44 45 46 47 STUVWXYZ ABCDEFG
    48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 HIJKLMNO PQRSTUVM
    58 59 5a 20 41 42 43 44 45 46 47 48 49 4a 4b 4c XYZ ABCD EFGHIJKL
    4d 4e 4f 50 51 52 53 54 55 56 57 58 00 30 b0 MNOPQRST UVWX.0.
  >
  
```

Figura 42. Captura de paquete 6LoWPAN en Wireshark.

Además debido a que el simulador Cooja no cuenta con el soporte para la plataforma CC2650STK que se utilizará en la implementación, se utilizó el nodo “*cc2420_driver*”; debido a que posteriormente se compararán los resultados obtenidos durante la simulación e implementación del escenario de telemonitoreo propuesto utilizando Contiki OS, también se evaluó el mote Zolertia “*zolertia_driver*”, con la finalidad de verificar que el comportamiento es aproximadamente similar entre una plataforma u otra dentro de la simulación en Cooja, y por consecuencia, debido a que los códigos que se cargan en los nodos

de la simulación son iguales a los que se cargan en los nodos de la implementación, no afectará el uso de una plataforma u otra.

Como se puede observar en la Figura 43, se tiene una comparación entre estos dos motes en relación al promedio de retransmisiones realizadas durante la simulación. Esta Figura ilustra que ambos motes tienen la misma tendencia, conforme se decrementa la frecuencia de encendido/apagado del radio controlada por la capa RDC ContikiMAC e incrementa el número de nodos, el número de retransmisiones realizadas aumenta. Esto se debe a que el periodo de transmisión de paquetes ECG no coincide con el periodo de actividad del radio, ocasionando que el nodo ECG realice un mayor número de transmisiones debido a la falta de acuse de recibo (paquete ACK) por parte del nodo central.

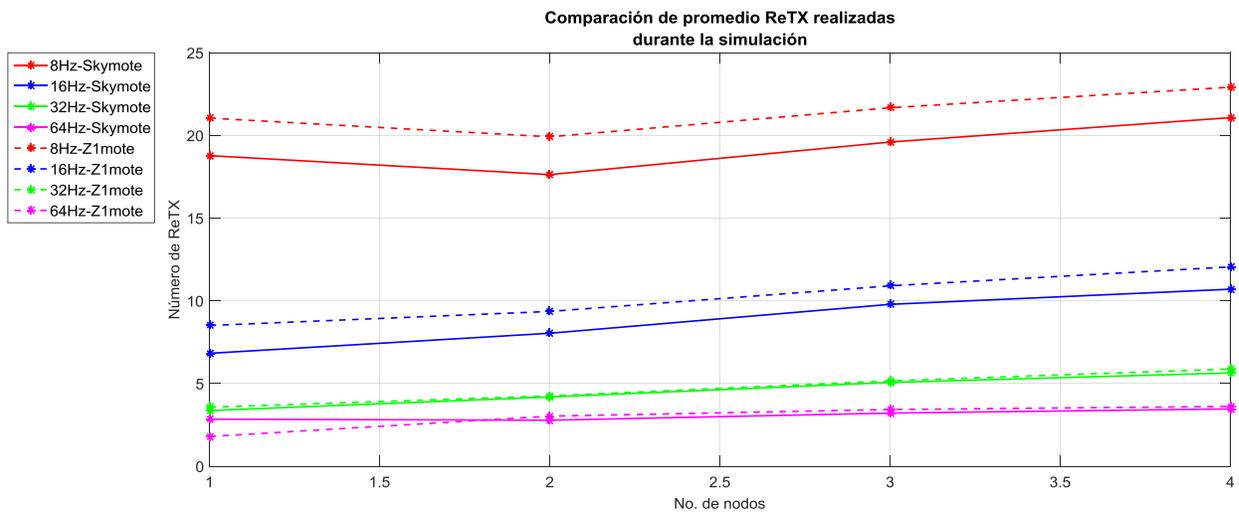


Figura 43. Comparación del promedio de retransmisiones realizadas del mote Sky y mote Zolertia en el simulador Cooja.

De acuerdo a lo mencionado anteriormente se pudo concluir que los nodos al utilizar una plataforma diferente no afecta el comportamiento de los protocolos del “*netstack*” de Contiki OS, ya que se tiene un comportamiento muy similar, esto permite que posteriormente se puedan comparar los resultados obtenidos mediante la simulación con el mote cc2420 y la implementación con el mote CC2650STK.

5.3.2 Emulación de ECG

Adicionalmente, también se validó la metodología de emulación del tráfico ECG. Esto se realizó verificando que el tráfico generado por la implementación y simulación correspondiera con el siguiente análisis: el total de paquetes generados por nodo corresponde a la cantidad de paquetes que se producen durante el

tiempo de simulación 300,000 ms, considerando que cada paquete se genera cada 46.667 ms. Tal y como se muestra en la ecuación (5)

$$N_T = \frac{\text{tiempo de duración del experimento}}{\text{tiempo de generación del paquete}} \quad (5)$$

$$N_T = \frac{300\,000\text{ ms}}{46.667\text{ ms}} = 6428.57 \approx 6428 \text{ paquetes}$$

Con el fin de ejemplificar la generación de paquetes con base a los nodos ECG que se tienen presentes en la red, en la Tabla 3 se muestra el total de paquetes generados cuando se cuenta con uno o más nodos ECG.

Tabla 3. Total de paquetes generados en la red de acuerdo a la cantidad de nodos ECG.

Cantidad de nodos ECG	Total de paquetes generados en la red
1	6,428
2	12,856
3	19,284
4	25,712

Cabe mencionar que este tipo de paquetes no incluye los paquetes de control, únicamente incluye paquetes de datos, que portan las 42 muestras de ECG y sus respectivos encabezados.

Con el objetivo de obtener las métricas correspondientes al estado estable tanto del experimento como de la simulación, los resultados que se muestran a continuación se encuentran validados por medio de la ejecución de varias simulaciones (para los resultados de la simulación), y varios experimentos físicos (para el caso de la implementación física). Cada simulación/experimento, fue de una duración de 300,000 ms cada uno. Conforme a ello la generación total de paquetes ECG por cada nodo (N_T), está definido por la ecuación (5).

Con base en ésta ecuación, durante los 300,000 ms que se ejecuta un experimento cada nodo ECG genera un total de 6428 paquetes (ver Figura 44). Por tal motivo el total de paquetes generados aumentará linealmente conforme se incremente el número de nodos ECG que conformen la red.

El número de paquetes generados son cuantificados dentro del código fuente, tanto de la simulación como de la implementación en la capa PHY. Ya que el total de estos paquetes está compuesto por los paquetes que se transmitieron (los que fueron enviados al canal radio) y los que se quedaron encolados.

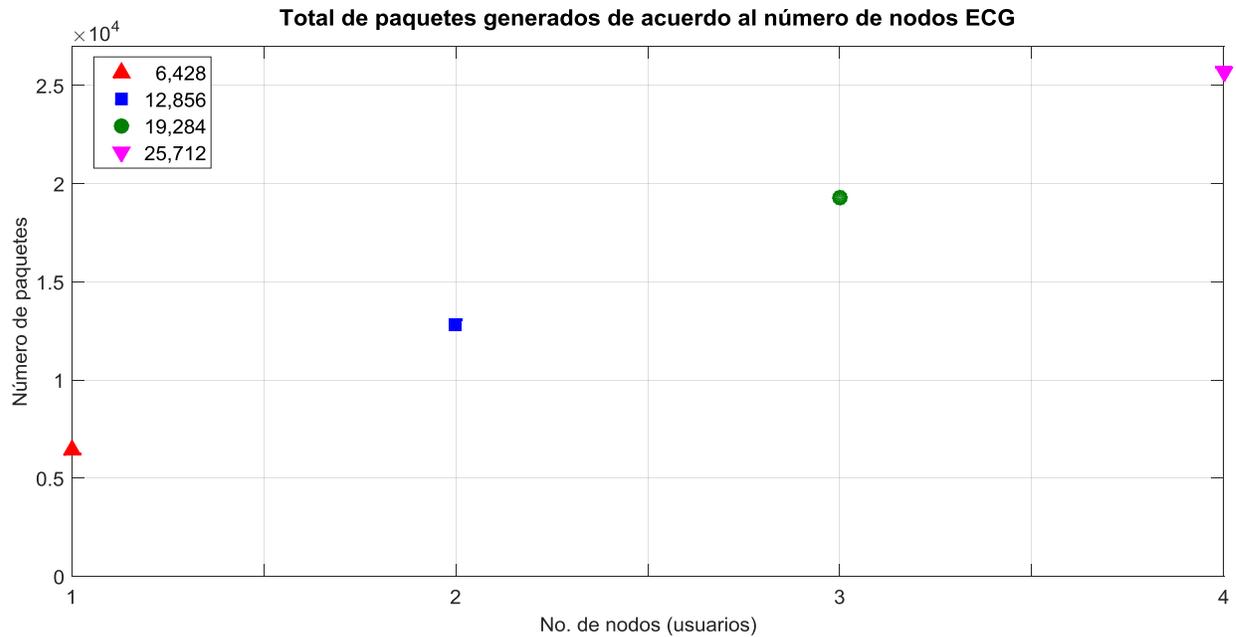


Figura 44. Total de paquetes generados de acuerdo al número de nodos ECG en la red 6LoWPAN.

Cada paquete que se genera está compuesto por los encabezados de los protocolos UDP, IPv6, IEEE 802.15.4 MAC. Posteriormente estos paquetes pasan a la capa MAC que ejecutará el mecanismo de acceso al medio CSMA encargado de indicar el momento en que cada nodo ECG puede transmitir o no un paquete hacia el nodo central. No obstante como se ha venido mencionando en Contiki OS CSMA es controlado directamente por ContikiMAC, esto ocasiona que al transmitir un paquete no se tenga certeza si el nodo central está activo para recibir el paquete que ha sido enviado por cualquiera de los nodos ECG que componen la red, ocasionando que los nodos ECG retransmitan el paquete hasta recibir un ACK por parte del nodo central para confirmar la recepción del paquete ECG.

5.4 Evaluación del desempeño de la red 6LoWPAN

El desempeño de la implementación de dicha red será evaluado con base a métricas de retransmisiones realizadas, paquetes recibidos y caudal eficaz. Es importante mencionar que los resultados obtenidos fueron validados por la ejecución de varios experimentos como se mencionó en la sección 5.3.2.

En este sentido se realizó un barrido para las frecuencias 8, 16, 32 y 64 Hz con las topologías de las Figuras 27 y 30 (simulación e implementación física), variando la cantidad de nodos ECG que forman la red 6LoWPAN. A continuación se muestra el total de retransmisiones realizadas durante la simulación y la implementación física, Figura 45 (a) y Figura 45 (b), respectivamente.

Como se puede notar en la Figura 45 (a) conforme el número de nodos incrementa el número de retransmisiones aumenta, independientemente de la f_{RDC} . Esto se debe principalmente a que se tienen un aumento en la cantidad de tráfico generado, ocasionado por la cantidad de usuarios que componen la red 6LoWAN. Esto quiere decir, que el incremento de tráfico genera mayor contención por el medio.

Por otra parte, el factor de la frecuencia f_{RDC} también influye notoriamente en el número de retransmisiones realizadas. Específicamente, conforme aumenta la f_{RDC} , el número de retransmisiones es menor, independientemente de la cantidad de nodos ECG que conforman la red. Con una f_{RDC} mayor el radio pasa de un estado activo a inactivo en menor tiempo. Por ejemplo, una $f_{RDC} = 64\text{Hz}$ el radio está activo cada 15.625ms , lo cual permite que durante estos intervalos de tiempo se coincida más frecuentemente con el periodo de transmisión de un paquete ECG 46.667ms y con ello se tenga un menor número de retransmisiones. Evidentemente conforme la frecuencia disminuye el periodo de transición entre estados aumenta. Tal es el caso de $f_{RDC} = 8\text{Hz}$, donde se tiene un periodo de 125ms , de forma que los periodos de actividad del radio ocurren con menor frecuencia durante el tiempo de duración de experimento, haciendo que los nodos ECG se vean en la necesidad de retransmitir el paquete un mayor número de veces para coincidir con un periodo en el que el nodo central se encuentre activo y pueda recibir un paquete.

Como se puede observar en la Figura 45 (b) los resultados de la implementación física difieren notoriamente con respecto a los resultados obtenidos en la simulación. Esto se debe a que durante la implementación cuando se tiene más de un nodo ECG conteniendo por el medio, éstos entran en conflicto. Específicamente, este conflicto se presenta en un escenario multiusuario y corresponde a una dificultad por parte de los nodos ECG a acceder al medio. Esto quiere decir, que después de que cada nodo ECG se encuentra asociado al nodo central, comienza con la transmisión de sus respectivos paquetes de datos. No obstante, debido a que es un escenario multiusuario, aumenta el tráfico en la red y por ende se genera mayor contención. Esto ocasiona que los nodos ECG entren en conflicto por no poder acceder al medio, y se lleve a cabo un mecanismo conocido como “*whatchdog*”¹⁰, implementado por Contiki OS, con la finalidad de suspender las actividades de transmisión del nodo ECG. Esto conlleva a que los nodos no completen su proceso de transmisión de paquetes de datos, de tal manera que el total de retransmisiones cuando se tienen 2 o más nodos ECG tiene un comportamiento similar a cuando únicamente se tiene uno.

¹⁰ El mecanismo *whatchdog*, es utilizado en sistemas embebidos que no pueden ser constantemente vigilados, de tal manera que su función principal es detectar automáticamente las anomalías de software, suspender y reiniciar el procesador.

La discusión anterior, se comprueba en la Figura 46, donde se tienen cuatro escenarios diferentes, en los cuales se puede observar la cantidad de paquetes transmitidos por nodo. En la Figura 46 (d), se muestra un escenario que cuenta con cuatro nodos ECG, de tal manera que para una $f_{RDC} = 64 \text{ Hz}$, ninguno de los nodos logra transmitir la cantidad de paquetes generados. De tal manera que el mejor de los escenarios se tiene cuando únicamente se cuenta con un nodo ECG, como se muestra en la Figura 46 (a).

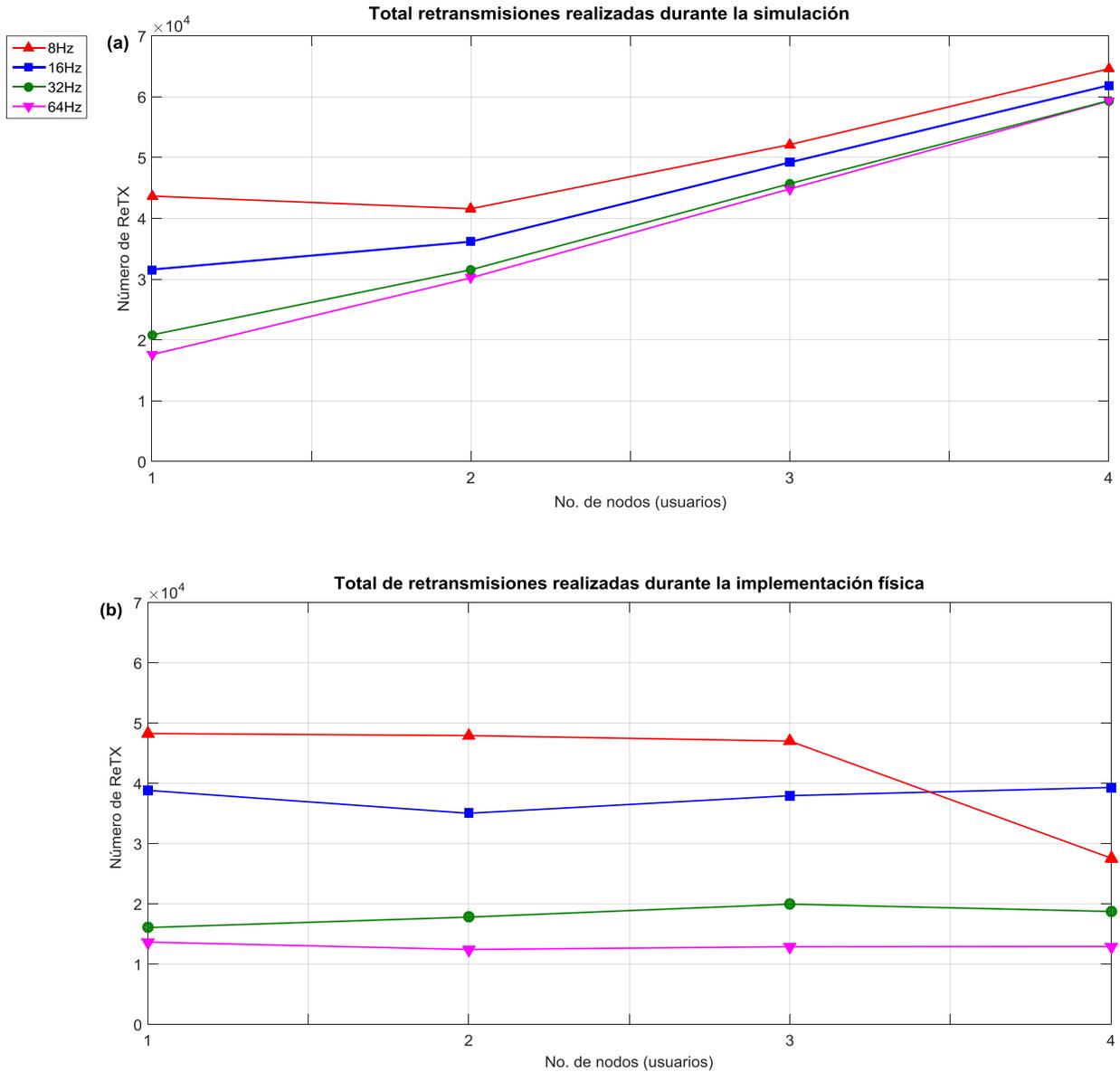


Figura 45. Total de retransmisiones realizadas conforme se incrementa el número de usuarios considerando diferentes valores f_{RDC} para (a) simulación. (b) cama de pruebas experimental.

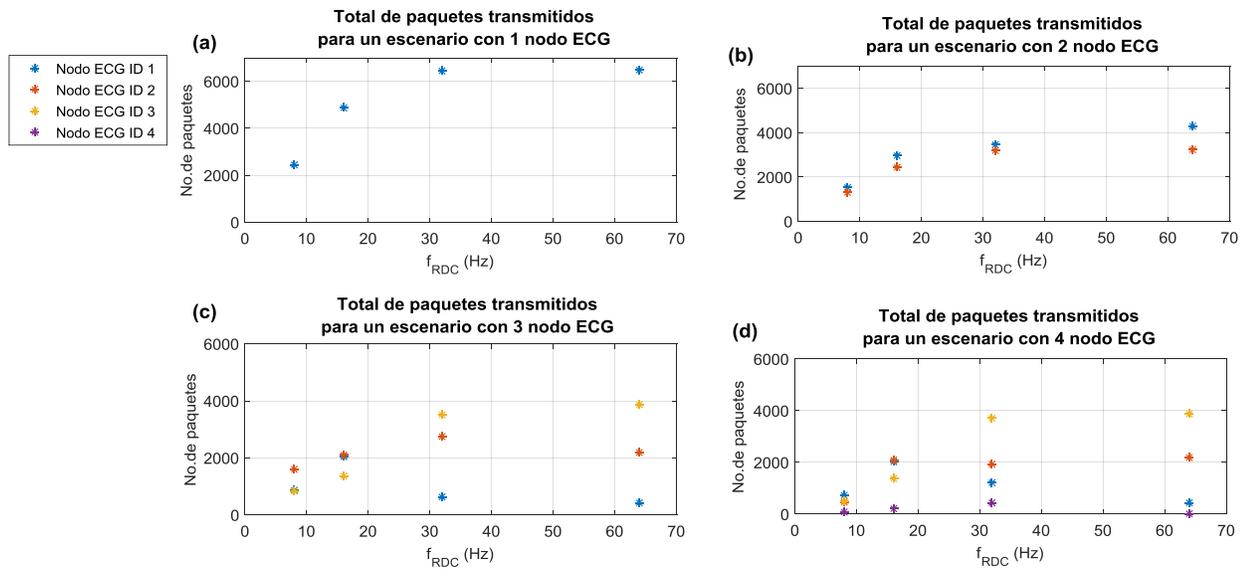


Figura 46. Total de paquetes transmitidos por nodo para un escenario con (a) un nodo ECG; (b) dos nodos ECG; (c) tres nodos ECG; (d) cuatro nodos ECG, considerando diferentes valores f_{RDC} .

De acuerdo a los resultados obtenidos en la Figura 47, el mejor de los casos se da para una $f_{RDC} = 64 \text{ Hz}$, en donde se tiene un promedio de 3.471 y 2.06 retransmisiones en el caso de la simulación e implementación, respectivamente. En la Figura 47 (a) se puede observar que, conforme la frecuencia disminuye el número de retransmisiones incrementa, teniendo un promedio de 4.5, 8.44 y 19.2725 retransmisiones, para las frecuencias de 32 Hz, 16 Hz y 8 Hz, respectivamente.

Por otro lado, de acuerdo a los resultados obtenidos en la Figura 47 (b), se observa que al igual que en la simulación el valor de f_{RDC} afecta en gran medida el número de retransmisiones realizadas, ya que independientemente de la cantidad de nodos ECG que componen la red 6LoWPAN. Esto es, que cuando f_{RDC} disminuye el número de retransmisiones incrementa debido a que los periodos de transición ocurren con menos frecuencia, haciendo que los nodos ECG se vean en la necesidad de retransmitir el paquete un mayor número de veces para coincidir con un periodo en el que el nodo central se encuentre activo y libre para recibir el paquete. Evidentemente el menor número de retransmisiones se tiene cuando el control del estado de actividad de radio corresponde a una $f_{RDC} = 64 \text{ Hz}$, equivalente a lo obtenido en la simulación. Por lo cual, el promedio para transmitir un paquete con esta frecuencia corresponde a 2.06 retransmisiones.

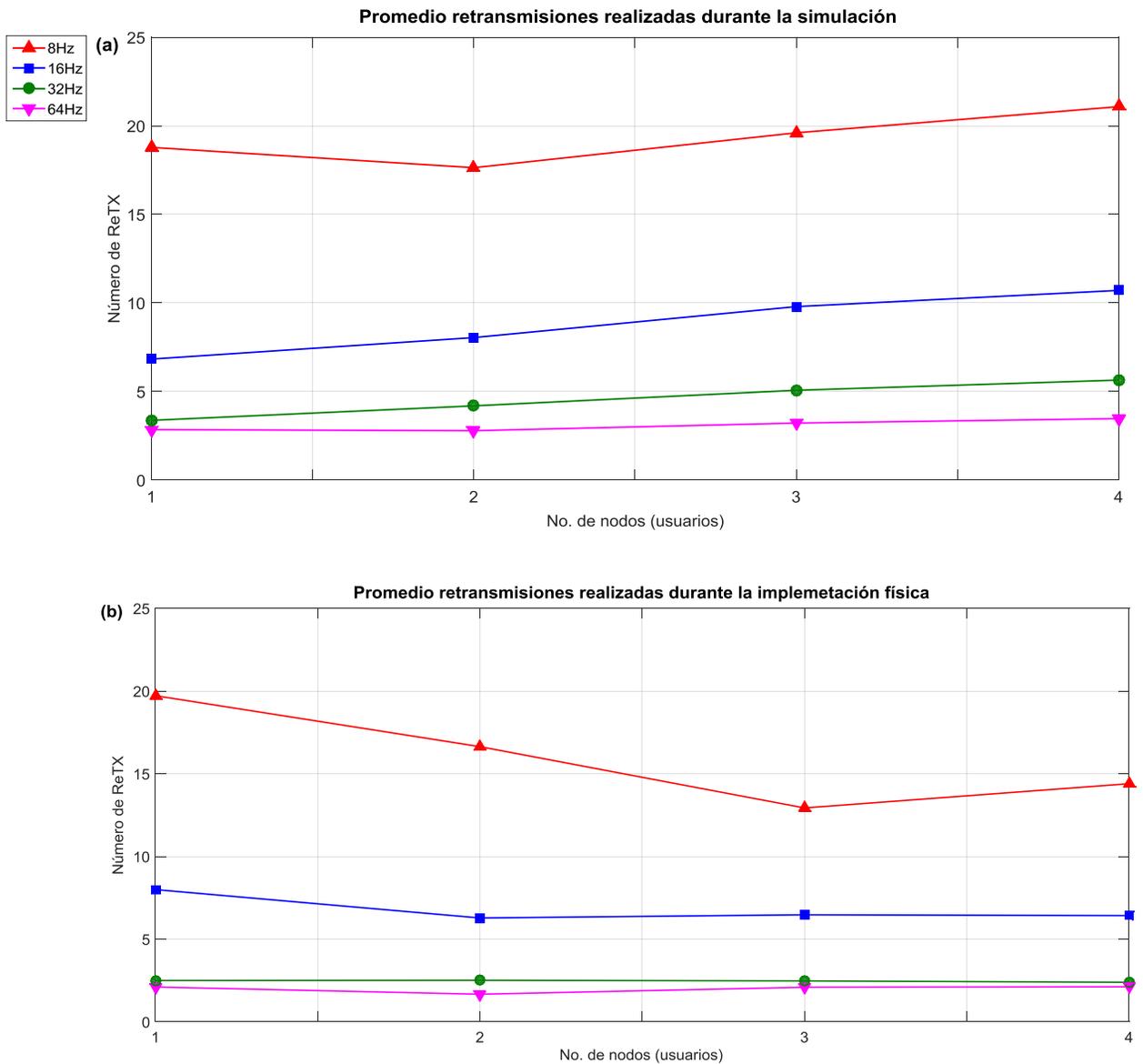


Figura 47. Promedio de retransmisiones realizadas conforme se incrementa el número de usuarios considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.

Como se observa en las gráficas de las Figuras 48 y 49 (resultados de la simulación e implementación física, respectivamente), se tienen cuatro escenarios diferentes donde varía el número de nodos ECG que componen la red 6LoWPAN. Haciendo el análisis para una frecuencia RDC, se esperaría que al incrementar el número de nodos que se encuentran conteniendo por el medio, el promedio de retransmisiones de cada nodo puede variar, debido a los mecanismos de detección de canal libre o de la ejecución del algoritmo de “backoff” que se efectúan por las capas RDC y MAC. No obstante, la cantidad de retransmisiones tiende a un solo valor, es decir, se tiene una respuesta plana. Esto indica que el “netstack” de Contiki OS añade un comportamiento equitativo (similar) en el promedio de retransmisiones que realiza para la transmisión de un paquete en cada nodo, en otras palabras, aunque se incremente el número de

nodos ECG se tendrán en promedio el mismo número de retransmisiones para transmitir un paquete ECG. De esta manera se puede comprobar que efectivamente el control de acceso al medio no es realizado únicamente por la capa MAC CSMA, sino por su trabajo en conjunto con la capa RDC ContikiMAC, lo cual ocasiona que el canal se reparta de manera uniforme para los nodos ECG que conforman la red 6LoWPAN.

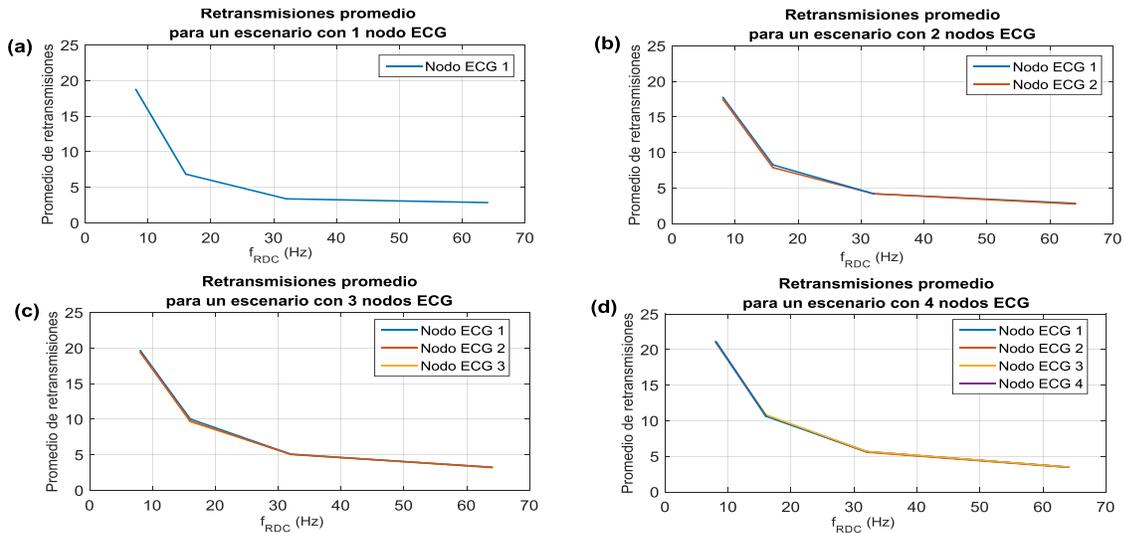


Figura 48. Respuesta del promedio de retransmisiones durante la simulación. (a) Promedio de retransmisiones escenario con 1 nodo ECG. (b) Promedio de retransmisiones escenario con 2 nodos ECG. (c) Promedio de retransmisiones escenario con 3 nodos ECG. (d) Promedio de retransmisiones escenario con 4 nodos ECG.

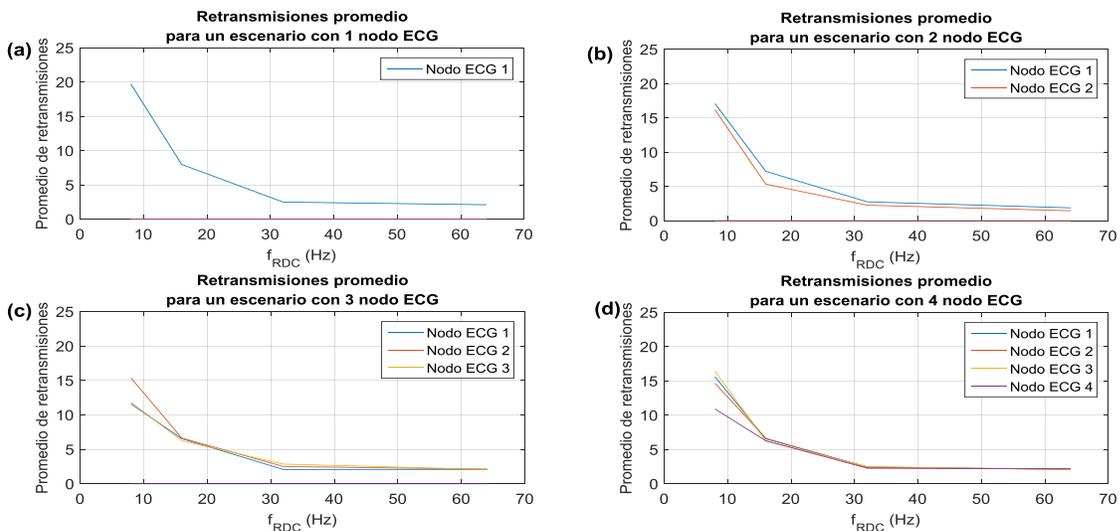


Figura 49. Respuesta del promedio de retransmisiones durante la implementación física. (a) Promedio de retransmisiones escenario con 1 nodo ECG. (b) Promedio de retransmisiones escenario con 2 nodos ECG. (c) Promedio de retransmisiones escenario con 3 nodos ECG. (d) Promedio de retransmisiones escenario con 4 nodos ECG.

Tomando en cuenta estos resultados es necesario analizar el desempeño de la red de tal manera que con base en el total de paquetes ECG generados durante la simulación se cuantifique la cantidad de los

recibidos por el nodo central, los que se pierden durante la transmisión y los que no son transmitidos debido a que se quedan en cola. A partir de ello se tendrá un acercamiento para poder afirmar que efectivamente utilizando el sistema operativo de Contiki OS es posible el desarrollo de una aplicación de telemonitoreo de la variable fisiológica ECG atendiendo a múltiples usuarios y que la implementación de la capa RDC no provoca comportamientos desfavorables en el desempeño de la red.

Con base al análisis que se realizó acerca del comportamiento que presenta la red 6LoWPAN (entre la relación que existe con respecto al incremento de tráfico y el aumento en la contención del medio), considerando diferentes valores de f_{RDC} . En la Figura 50, se muestran el total de paquetes transmitidos, en donde se puede observar, que no todos los paquetes que se generan durante la implementación, son enviados al aire. Esto quiere decir, que parte de los paquetes generados se quedan en la cola del búfer para ser transmitidos, debido al mecanismo de la capa RDC ContikiMAC.

En la Tabla 4 se muestra el porcentaje de paquetes que fueron transmitidos con relación a los que fueron generados para el caso de la simulación. Como se puede observar para los porcentajes de paquetes transmitidos para cualquiera de los escenarios planteados (excepto $f_{RDC} = 16 \text{ Hz}$ con un nodo ECG) se encuentran por debajo del 37 %, haciendo inviable proponer un escenario multiusuario para el monitoreo de la variable fisiológica ECG¹¹. Cuando se tiene una $f_{RDC} = 32 \text{ Hz}$, el porcentaje de paquetes transmitidos aumenta, no obstante con base a los porcentajes obtenidos para un escenario mayor a dos usuarios ECG, el porcentaje de los paquetes que no fueron transmitidos (paquetes que se quedaron almacenados en buffer) es más de la mitad de los paquetes que fueron generados. Por último para una $f_{RDC} = 64 \text{ Hz}$ el total de paquetes transmitidos mínimo es de 66.47% cuando se tienen 4 nodos ECG y hasta 96.03% con un solo nodo ECG.

De igual manera en Tabla 4 se resumen los porcentajes de paquetes que fueron transmitidos en relación con los que fueron generados para el caso de la implementación física (ver Figura 50 (b)), y como se puede observar para las frecuencias $f_{RDC} = 32 \text{ Hz}$ y $f_{RDC} = 64 \text{ Hz}$ con un nodo ECG se consiguen transmitir casi en su totalidad los paquetes generados. Sin embargo para los otros escenarios el porcentaje de paquetes transmitidos se encuentra por debajo del 50 %, haciendo inviable tener un escenario con más de un nodo ECG. A demás en esta tabla también se puede observar que entre los nodos que conforman la red, durante el tiempo de ejecución de los experimentos únicamente se transmiten los paquetes equivalentes a cuando

¹¹ Tomando como referencia el trabajo de (Casillas, 2012), hace referencia que un porcentaje de paquetes perdidos mayor a 25% es inviable para cualquier aplicación.

se tiene un solo nodo ECG como usuario, esto a causa del conflicto que presentan los nodos al contender por el medio.

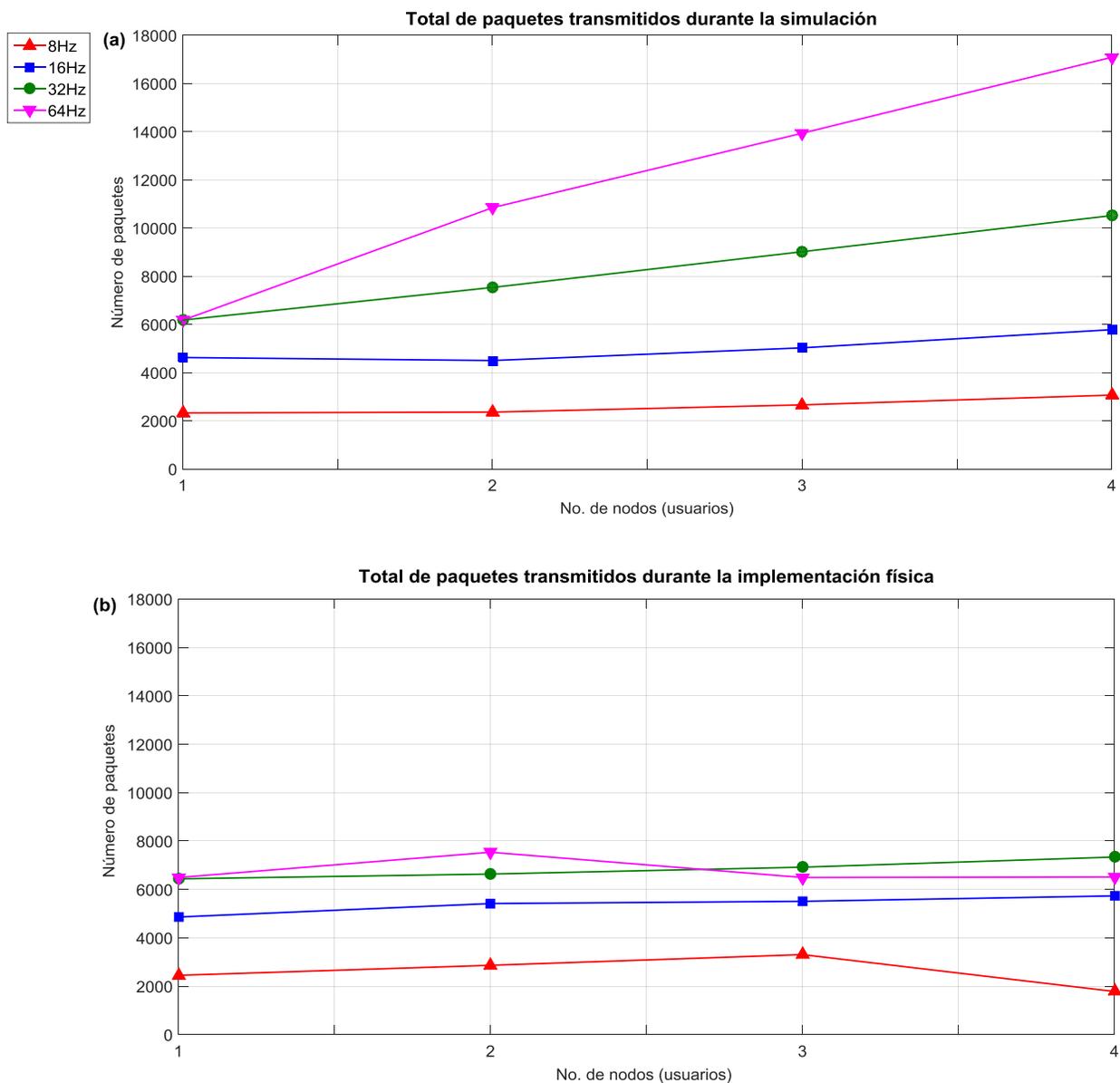


Figura 50. Total de paquetes transmitidos conforme se incrementa el número de nodos considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.

Tabla 4. Porcentaje de paquetes transmitidos durante la simulación y cama de pruebas experimental

No. de nodos ECG	Total de paquetes generados	Porcentaje de paquetes transmitidos (%)	
		SIMULACIÓN	IMPLEMENTACIÓN FÍSICA
1	6,428	36.17	38.10
2	12,856	18.36	22.28
3	19,284	13.79	17.14
4	25,712	11.92	25.25
1	6,428	71.95	75.61
2	12,856	35.00	42.13
3	19,284	26.06	28.57
4	25,712	22.48	29.32
1	6,428	96.11	98.46
2	12,856	58.60	51.62
3	19,284	46.76	35.89
4	25,712	40.91	25.26
1	6,428	96.03	98.97
2	12,856	84.32	58.63
3	19,284	72.24	33.68
4	25,712	66.47	25.33

Con base en estos resultados, se puede establecer una cota de tolerancia que defina el porcentaje de paquetes que deben ser transmitidos del total de paquetes que se generaron. De tal manera que sí en un escenario evaluado se quisiese un porcentaje de paquetes transmitidos mayor al 90% entonces se podrían tener únicamente un usuario ECG y un nodo central para una frecuencia RDC igual 32 y 64 Hz, tanto para el caso de la simulación como para la implementación física.

En resumen se puede notar que debido al mecanismo que agrega ContikiOS de la capa RDC afecta en gran medida el control de acceso al medio y en consecuencia hace que gran parte de los paquetes generados se queden encolados y no puedan ser transmitidos a causa de las retransmisiones que tienen que ser realizadas para que el nodo central pueda recibir un paquete. Por lo tanto para el monitoreo de una variable fisiológica como el ECG, es de gran importancia considerar la situación médica que requiere un paciente y a partir de ello definir una cota de tolerancia, esto es, en una situación de emergencia, es probable que se pueda tolerar cierta pérdida de paquetes, si a cambio se reduce el retardo, con tal de obtener la mayor cantidad de información ECG. Sin embargo, en el caso de seguimiento de pacientes con enfermedades crónicas o pacientes a los cuales se les está realizando un diagnóstico en una cita normal, entonces se podría tolerar cierto retardo, si a cambio se obtiene información más precisa y sin pérdidas, de tal manera que se pueda realizar un diagnóstico más certero.

Por otro lado, a partir del análisis anterior, enseguida se cuantificarán la cantidad de paquetes que efectivamente fueron recibidos por el nodo central (paquetes transmitidos con éxito) y la cantidad de paquetes que se perdieron a causa de colisiones o debido a que no se recibió ACK dentro del intervalo de tiempo t_{RDC} (paquetes transmitidos sin éxito).

Como se puede observar en la Figura 51 (a), el porcentaje de paquetes que son recibidos por el nodo central, es mayor únicamente cuando se tienen de uno a dos nodos ECG con respecto a los paquetes que se transmitieron sin éxito. Para una $f_{RDC} = 64 \text{ Hz}$, con un nodo ECG, se logra de un total de 6428 paquetes generados transmitir 6173 y de estos últimos recibir 6165, es decir, un 99.87% (ver Tabla 5). No obstante, al aumentar el número de nodos el porcentaje de recibir paquetes con éxito decae por debajo del 50%, lo cual es crucial en un escenario de telemonitoreo, ya sea en una situación de emergencia o en un diagnóstico en una cita normal, como se comentó anteriormente.

Por otra parte, para el caso de la implementación, en la Figura 51 (b) y la Tabla 5 se puede observar que el porcentaje del total de paquetes recibidos es casi del 100% para todos los escenarios. En primera instancia pudiera parecer que los resultados de la implementación física son mejores que los obtenidos en la simulación, dado que se recibe una mayor cantidad de paquetes del total de los paquetes transmitidos en los diferentes escenarios. Sin embargo, el porcentaje del total de paquetes recibidos para cada uno de los escenarios en la implementación física corresponde a los paquetes transmitidos por un solo nodo ECG. Es decir, en los escenarios en los que existe más de un nodo se tiene el problema que ya se ha venido mencionando, en donde los nodos entran en conflicto al estar conteniendo por el medio, ocasionando que algunos de ellos ejecuten el mecanismo “*whatchdog*” de tal manera que éstos no puedan transmitir en su totalidad los paquetes generados.

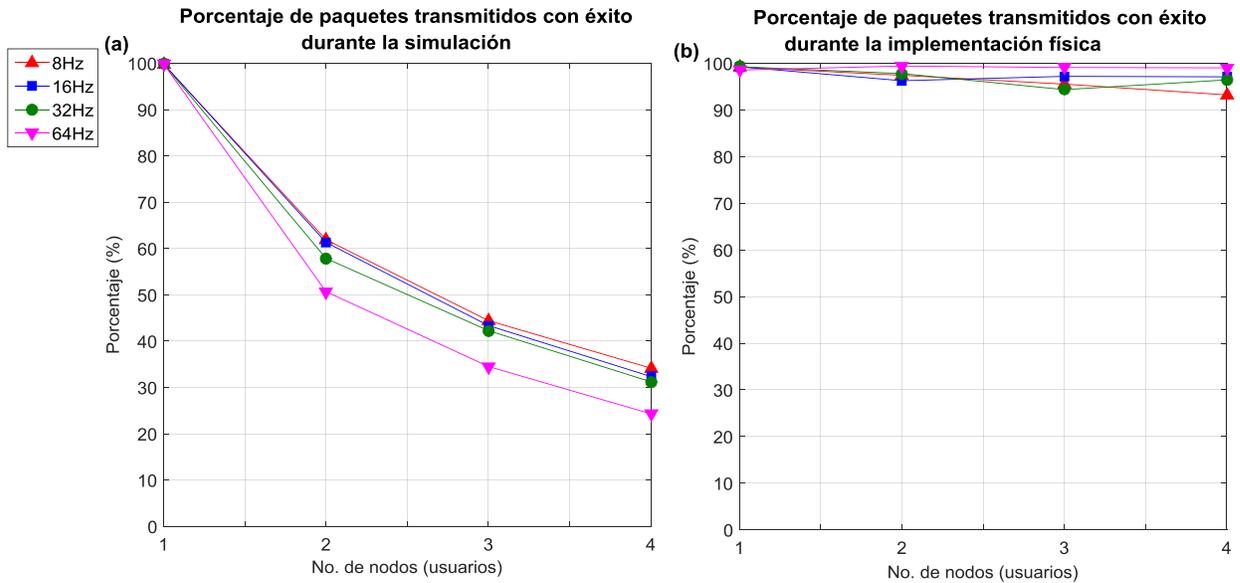


Figura 51. Porcentaje de paquetes transmitidos con y sin éxito conforme se incrementa el número de nodos considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.

Tabla 5. Resumen de resultados de paquetes transmitidos con y sin éxito durante la simulación y la cama de pruebas experimental.

No. de nodos ECG	Total de paquetes generados	SIMULACIÓN			IMPLEMENTACIÓN FÍSICA		
		Total de paquetes transmitidos	Porcentaje de paquetes transmitidos (%)		Total de paquetes transmitidos	Porcentaje de paquetes transmitidos (%)	
			Con éxito	Sin éxito		Con éxito	Sin éxito
1	6,428	2,325	99.74	0.27	2,449	99.18	0.82
2	12,856	2,360	61.91	38.07	2,864	97.43	2.57
3	19,284	2,659	44.45	55.58	3,306	95.54	4.46
4	25,712	3,066	34.18	65.82	6,493	98.56	1.44
1	6,428	4,625	99.87	0.12	4,860	99.26	0.74
2	12,856	4,500	61.33	38.64	5,417	96.27	3.73
3	19,284	5,026	43.39	56.61	5,509	97.20	2.80
4	25,712	5,780	32.37	67.63	7,538	99.41	0.59
1	6,428	6,178	99.85	0.15	6,329	99.26	0.74
2	12,856	7,534	57.91	42.09	6,636	97.80	2.20
3	19,284	9,017	42.29	57.71	6,920	94.36	5.64
4	25,712	10,520	31.23	68.75	6,494	99.10	0.90
1	6,428	6,173	99.87	0.13	6,362	98.52	1.47
2	12,856	10,840	50.71	49.33	7,538	99.41	0.59
3	19,284	13,930	34.54	65.50	6,494	99.10	0.90
4	25,712	17,090	24.32	75.66	6,514	99.02	0.98

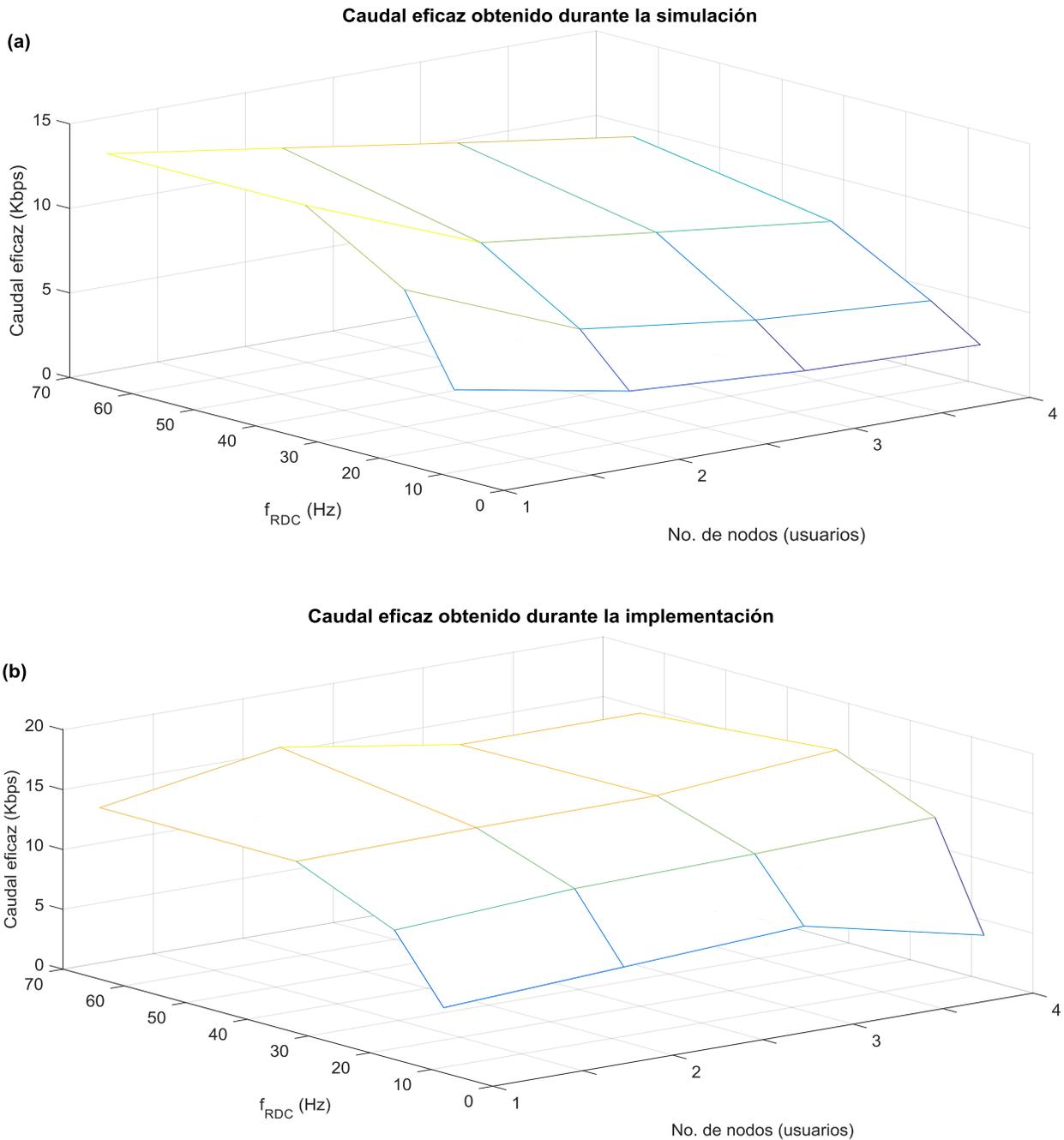


Figura 52. Caudal eficaz conforme se incrementa el número de nodos considerando diferentes valores de f_{RDC} para (a) simulación; (b) cama de pruebas experimental.

A partir de los resultados obtenidos hasta el momento, se obtuvo una gráfica en términos del caudal eficaz, aumentando el tráfico en la red (el número de usuarios concurrentes) y variando el valor de la f_{RDC} . En la gráfica de la Figura 52, tanto en la simulación como en la implementación física, se observa que, el caudal eficaz decrece a medida que la f_{RDC} disminuye. Por ejemplo, en la Figura 52 (a), para una $f_{RDC} = 64 \text{ Hz}$, se tiene un caudal eficaz de 13.81 Kbps, cuando únicamente se tiene un usuario, es decir, no se tiene contención por el medio con otros usuarios; a manera que se incrementa la contención por el medio,

el caudal eficaz disminuye notoriamente, hasta llegar a los 9.309 Kbps. Sin embargo, cuando se tiene una $f_{RDC} = 8 \text{ Hz}$, debido a que los periodos de encendido/apagado del radio son menos frecuentes, el caudal eficaz es mucho menor comparado a los otros valores de f_{RDC} .

Este comportamiento, era el esperado, ya que de acuerdo a los resultados que se han obtenido en la implementación del estándar 6LoWPAN utilizando Contiki OS la inclusión de una capa RDC, con el objetivo principal de ahorrar de energía, introduce comportamientos alejados de los paquetes que se esperan recibir por parte de los nodos ECG que componen la red. Esto condiciona a llevar a Contiki OS a una aplicación real de telemonitoreo multiusuario.

En este sentido, se decidió apoyar los resultados obtenidos utilizando la pila de protocolos TIMAC, mediante un escenario, en el que se tienen: un nodo central y de 1 a 4 nodos sensores que monitorean la señal ECG de un paciente, cabe mencionar que el tráfico de la variable fisiológica ECG, es emulado.

En la Figura 53 se muestran una captura realizada por el sniffer 6LoWPAN, cuando únicamente se cuenta con un nodo ECG. Como se puede observar, como solo se tiene un nodo ECG, y a diferencia del “netstack” que proporciona Contiki OS, TIMAC cuenta con la implementación de una capa MAC que no está controlada por una capa RDC para efectuar sus tareas, es decir, se tiene una capa MAC más pura. De esta manera para enviar un paquete ECG únicamente es necesario transmitirlo una única vez, y enseguida se envía un acuse de recibido (con el mismo formato, del utilizado en Contiki OS) por parte del nodo central. Comparando con los resultados de la implementación en Contiki OS, cuando se tiene el mejor de los casos con una $f_{RDC} = 64 \text{ Hz}$ y un nodo ECG, en promedio es necesario retransmitir el paquete 2.848 en la simulación y 2.105 en la implementación (ver las **Figura 47 (a)** y **Figura 47 (b)**, respectivamente).



Figura 53. Captura sniffer 6LoWPAN de la transmisión de paquetes ECG, para una red 6LoWPAN con un nodo ECG.

De acuerdo, a lo mencionado anteriormente, se puede concluir, que utilizando la pila de protocolos que proporciona TIMAC, se pueden tener mejores resultados, que los que se obtuvieron con el sistema operativo Contiki. En otras palabras, si no se consideran las capas de RDC y trama que agrega el “*netstack*” de Contiki OS los resultados serán significativamente mejores en términos de paquetes retransmitidos con éxito, gracias a que los nodos no se verán en la necesidad de retransmitir el paquete con base a la recepción de un ACK durante un intervalo de tiempo, sino que se basará únicamente en el método de acceso al medio CSMA que define el estándar de 6LoWPAN.

Cabe mencionar, que para una implementación más completa utilizando TIMAC, se requiere de realizar el conteo del mismo modo que como se realizó en Contiki OS. Sin embargo, con el trabajo realizado hasta el momento en esta investigación, se tiene un acercamiento a la consolidación de una implantación del estándar 6LoWPAN en un escenario de telemonitoreo multiusuario utilizando a TIMAC como pila de protocolos.

5.5 Conclusiones sobre la simulación y desarrollo de la red inalámbrica de sensores utilizando Contiki OS y TIMAC.

Los resultados de la simulación de la red 6LoWPAN utilizando Contiki OS muestran que para todas las evaluaciones realizadas, es decir, contemplando los diferentes escenarios (variando el número de nodos que conforman la red y el valor de la f_{RDC}) el mejor de los desempeños se obtiene una $f_{RDC} = 64 \text{ Hz}$, debido a que con esta frecuencia, los nodos ECG pasan de un estado activo a inactivo en un lapso de tiempo más corto que los otros valores de f_{RDC} . De esta manera cuando un paquete es enviado por un nodo ECG hacia el nodo central, se requiere un número menor de retransmisiones para que dicho paquete sea recibido.

Por otro lado, tomando como referencia el valor $f_{RDC} = 64 \text{ Hz}$, los resultados obtenidos con base al incremento de nodos ECG que conforman la red 6LoWPAN, permiten observar que únicamente es posible atender a un nodo ECG, ya que de un total de 6,428 paquetes generados, solamente son transmitidos 6,173, es decir, el 96.03% del total de los paquetes generados. Además del total de paquetes transmitidos solo el 99.87% es recibido por el nodo central.

Contemplando lo anteriormente mencionado, si se incrementa el tráfico en la red (número de usuarios), la cantidad de paquetes recibidos decrementa considerablemente por debajo de 50%. Esto se debe principalmente a que los picos de tráfico aumentan el riesgo de colisiones de radio en la red, ya que se producen colisiones de datos cuando llegan múltiples transmisiones a un receptor al mismo tiempo, causando la pérdida de datos y las retransmisiones. El riesgo de colisiones de datos se agrava en la red, ya

que el nodo central está despierto menos tiempo, y por lo tanto tienen menos oportunidad para recibir datos.

Por otro lado, los resultados de la implementación de la red 6LoWPAN utilizando Contiki OS, mostró un comportamiento similar a los que se obtuvo en la simulación. Sin embargo el sistema operativo de Contiki OS tiene un efecto de control en los radios físicos, ya que cuando se tiene más de un nodo ECG conteniendo por el medio, éstos entran en conflicto, ocasionando que se lleve a cabo un mecanismo de “*whatchdog*”, con la finalidad de suspender la transmisión de paquetes por parte de los nodos involucrados. Esto conlleva a que también en la implementación solamente se puedan recibir los paquetes correspondientes a un nodo ECG, utilizando la $f_{RDC} = 64 \text{ Hz}$, ya que de un total de 6362 paquetes transmitidos, se recibe el 98.52%. En cambio cuando se incrementa el número de nodos la cantidad de paquetes recibidos corresponde a un escenario en donde solamente se tiene un nodo ECG.

Aunado a ello, el caudal eficaz tanto en la simulación como en la implementación, se ve afectado por el valor de la f_{RDC} , donde si ésta disminuye el caudal eficaz también desciende. Debido a que los periodos de transición entre el encendido y apagado del radio son más largos, por lo cual la probabilidad de que un paquete sea recibido es menor, ocasionando que el total de paquetes que se transmiten exitosamente sean menores cuando la f_{RDC} se reduce.

Esto permite reflexionar acerca de la importancia que se tiene o no de perder un paquete desde el punto de vista médico, es decir, qué tan buena es la calidad de la señal recibida por parte de los nodos ECG para que un médico realice un diagnóstico a pesar de que se haya tenido pérdida de paquetes (muestras de la señal ECG). Esto dependerá específicamente de la situación del paciente en cuestión, es decir, que tan crítico es el estado del paciente. Para ello, se requiere de un estudio en técnicas de interpolación para reconstrucción de señales que permitan mejorar la calidad de las señales ECG recibidas. Sin embargo, se encuentra del alcance de investigación.

En general, la evaluación del estándar 6LoWPAN utilizando Contiki OS tanto en la simulación como en la implementación, muestra que el sistema operativo agrega características que controlan de cierto modo el comportamiento de los radios (virtuales y físicos), tales como:

- Comportamiento equitativo, en las retransmisiones realizadas para la entrega de cada paquete, independientemente del número de nodos y la f_{RDC} que se esté empleando.
- El control de acceso al medio, realizado en conjunto por las capas RDC y MAC del “*netstack*” de Contiki OS. Ocasionando que cuando se envíe un paquete, la recepción del mismo dependa del

ciclo de trabajo de nodo central, lo cual disminuye el desempeño en cuanto el número de paquetes transmitidos y los que sí fueron recibidos por el nodo central.

- El mecanismo de “*whatchdog*” implementado por Contiki OS, causando que las actividades de los nodos involucrados sean suspendidas, de tal manera que no puedan continuar con su transmisión de paquetes. En otras palabras, Contiki OS controla el comportamiento de los nodos, de modo que no es factible tener la implementación de un escenarios multiusuario para el monitoreo de la variable fisiológica ECG.

De acuerdo a lo anterior, se puede percibir que no se cumplen las especificaciones que dicta el estándar 6LoWPAN en Contiki OS. Si bien Contiki cuenta con algunos de los protocolos que conforman 6LoWPAN, también agrega una capa RDC para mejorar la eficiencia en el ciclo de trabajo del radio. Esto ocasiona que la implementación de la capa MAC no sea pura. De tal manera que es conveniente indagar en otras soluciones que brinden una mejor pila de protocolos para la implementación de una red 6LoWPAN, tal como TIMAC.

Por esta razón, se realizó una implementación sencilla utilizando TIMAC, en donde se pudo observar por medio de una captura del sniffer 6LoWPAN, que para un escenario con un nodo ECG, no se presentan retransmisiones para el envío de los paquetes ECG. A diferencia de los resultados obtenidos tanto en la simulación e implementación de éste mismo escenario en Contiki OS, donde el número de retransmisiones es de 2.848 y 2.105, respectivamente.

No obstante, al utilizar TIMAC, existe un compromiso entre utilizar una pila de protocolos que ofrece una capa MAC que permite mejorar la entrega de paquetes exitosos pero que en consecuencia mantiene encendido el radio y ocasionar que el consumo de energía sea mayor.

Capítulo 6. Conclusiones y trabajo futuro

6.1 Introducción

En el presente trabajo de investigación se evaluó el desempeño del estándar 6LoWPAN en una red orientada a la transmisión de señales biomédicas, es decir, encaminada a una implementación de una aplicación de monitoreo de las mismas en un escenario multiusuario, mediante simulaciones y la implementación física de la misma. Durante el proceso surgieron inconvenientes a los cuales se les trató de dar la mejor solución posible, adquiriendo con ellos nuevos conocimientos y herramientas. A lo largo de este capítulo se presentan propuestas de trabajo a futuro que permitirían llevar a cabo una serie de mejoras al trabajo de investigación aquí presentado.

La aportación principal de esta tesis radicó en realizar una evaluación de la implementación del estándar 6LoWPAN en un escenario de telemonitoreo multiusuario, en la que se consideren las características de la pila de protocolos del sistema operativo en el cual será implementado así como el perfil de tráfico de la variable fisiológica que será transmitida. De tal manera, que realizando esta evaluación se pueda verificar la factibilidad de utilizar este estándar para escenarios de telemonitoreo. Para lograr esto se realizó una evaluación de una red 6LoWPAN utilizando el “*netstack*” que ofrece ContikiOS mediante simulaciones y una implementación física. Por lo tanto, con esto se logra cumplir plenamente con el objetivo principal de esta tesis. Además de las aportaciones mencionadas, se produjo otra que no fue planteada en un inicio, como la implementación del estándar 6LoWPAN utilizando la pila de protocolos que ofrece TIMAC, brindando solución a la estructura y características que se definen en el sistema operativo de Contiki OS.

6.2 Conclusiones

A continuación se muestra la discusión de las conclusiones y de las aportaciones desarrolladas durante este proyecto de investigación. Se presentan cada una de ellas divididas en la simulación de la red inalámbrica de sensores en Cooja, implementación de la red inalámbrica de sensores utilizando Contiki OS e implementación de la red inalámbrica de sensores utilizando TIMAC.

6.2.1 Sobre la simulación de la red inalámbrica de sensores en Cooja.

Se realizó una revisión bibliográfica de las características y propiedades más importantes del estándar 6LoWPAN. Además se investigaron en la literatura las implementaciones realizadas del estándar 6LoWPAN cuya aplicación estuviera relacionada con aplicaciones de telemonitoreo de variables fisiológicas. A partir de este estudio se propuso un escenario de una red 6LoWPAN, compuesta de nodos sensores que

monitorean la variable fisiológica ECG. Con el objetivo de evaluar el desempeño de la red propuesta, se implementó éste escenario de simulación mediante el simulador Cooja de Contiki OS. A continuación se presentan las conclusiones más relevantes obtenidas para esta etapa.

- Cooja es una herramienta de simulación de código abierto basado en Java con una potente interfaz gráfica que ha ganado gran popularidad dentro de la comunidad científica, ofreciendo un entorno útil y versátil que permite evaluar al estándar 6LoWPAN sobre varias plataformas, con la ventaja de que también pueda ser implementado físicamente en ellas. Sin embargo, la curva de aprendizaje que se requiere tanto en términos de programación como de la tecnología a evaluar debe considerarse.
- De acuerdo al estudio de literatura realizado de las implementaciones realizadas del estándar 6LoWPAN y el enfoque que se planteó en este trabajo de tesis en la cual se considera el manejo de esta tecnología para el transporte de variables fisiológicas, se propuso un escenario multiusuario, compuesto por 4 nodos y un nodo central que cuentan con la pila de protocolos del estándar hasta la capa de transporte. Gracias a este escenario fue posible tener una evaluación más completa del desempeño de una red 6LoWPAN en este tipo de aplicaciones, considerando las características que agrega el sistema operativo Contiki OS.
- El “*netstack*” que ofrece Contiki OS añade una capa RDC, encargada de controlar el encendido y apagado del radio. De tal manera que la capa MAC se ve afectada al realizar el control de acceso al medio, ya que depende de la capa RDC para poder enviar o recibir un paquete. Es por ello que desde este punto el “*netstack*” de 6LoWPAN que provee Contiki difiere con el estándar 6LoWPAN definido por la IETF. Esta capa agrega un comportamiento no deseado en cuanto el número de paquetes que son transmitidos con éxito.
- Los resultados obtenidos al evaluar el funcionamiento de la red 6LoWPAN, permitieron demostrar que la capa RDC que agrega Contiki OS, repercute en el número de retransmisiones que se realizan para que un paquete sea recibido por el nodo central. Al realizar el barrido de frecuencias f_{RDC} , se pudo observar que cuando se tiene una f_{RDC} mayor, el intervalo de cuando un nodo ECG pasa de su estado activo a inactivo es más corto. De esta manera cuando un paquete es enviado por un nodo ECG hacia el nodo central, se requiere un número menor de retransmisiones para que dicho paquete sea recibido, provocando que el caudal eficaz aumente.

- Respecto al número de usuarios ECG que pueden ser atendidos por el nodo central, utilizando 6LoWPAN de Contiki OS y tomando en cuenta lo anteriormente mencionado, los resultados obtenidos mostraron que únicamente es posible atender a un usuario ECG, debido a que se reciben con éxito en su totalidad la cantidad de paquetes transmitidos. La causa de ello, se debe a que al incrementar el número de nodos, la cantidad de paquetes transmitidos incrementa, y en consecuencia se tiene un canal saturado. Aunado a ello, el mecanismo de la capa RDC afecta la recepción de paquetes en el nodo central. Esto se debe a que éste se encuentra activo cada $\frac{1}{f_{RDC}}$ y cuando la f_{RDC} se incrementa, éste periodo es inversamente proporcional, por lo tanto disminuye, dando lugar a que el nodo central no tenga la oportunidad para recibir dichos paquetes.

6.2.2 Sobre el desarrollo de la red inalámbrica de sensores utilizando Contiki.

Contemplando la revisión bibliográfica que se realizó para la simulación de un escenario de telemonitoreo utilizando el estándar 6LoWPAN, se procedió a llevar a cabo una implementación física del mismo. Con la finalidad de evaluar el desempeño de la red propuesta, utilizando de igual manera el “*netstack*” de Contiki OS. Enseguida se presentan las principales conclusiones que se generaron del trabajo en esta etapa.

- Realizar una implementación física no es una tarea trivial. Debido a que se requiere del conocimiento de la plataforma en la cual se realizará la implementación. Además de crear el soporte para la misma, realizando las configuraciones dentro de Contiki OS para que el “*netstack*” de 6LoWPAN pueda ser cargado correctamente.
- Los resultados que se obtuvieron a partir de la implementación física del escenario propuesto, mostraron un comportamiento similar a los que se obtuvieron en la simulación. En donde, se pudo observar, el efecto de la capa RDC en cada uno de los nodos que componen la red. Ya que ésta influye tanto en el desempeño de los nodos ECG, limitando a que se realicen el envío de paquetes hasta no recibir un acuse de recibido por parte del nodo central; así como en el nodo central, donde si éste no se encuentra activo no podrá recibir o enviar paquetes.
- En relación al número de nodos ECG concurridos que pueden ser atendidos por el nodo central, en el escenario propuesto, se pudo concluir que únicamente es posible atender a un nodo ECG. Ya que cuando se tiene más de un nodo ECG contendiendo por el medio, éstos entran en conflicto, causando que se lleve un mecanismo de “*whatchdog*”, implementado por el mismo sistema

operativo de Contiki OS. Esto con la finalidad de suspender la transmisión de paquetes por parte de los nodos involucrados. Por lo tanto, lo anteriormente mencionado, muestra que Contiki OS es un sistema que ejerce cierto control a los radios físicos que para la aplicación en cuestión no es factible su uso.

- El mecanismo de “*whatchdog*”, que se ejecuta en la implementación física, ocasiona que al incrementar el número de nodos, la recepción de paquetes corresponda únicamente a como si solo se tiene un nodo ECG y el nodo central, sin importar el número de la f_{RDC} que se esté utilizando.
- En general, con las conclusiones hasta el momento mencionadas, se puede inferir que Contiki OS es un sistema operativo orientado aplicaciones de bajo consumo, es decir, aplicaciones en las cuales no se requiera que los radios se encuentren activos la mayor parte del tiempo, ya sea para recibir o transmitir datos. Por lo tanto, en un escenario como el que se propuso en este trabajo de tesis, donde se monitorean variables fisiológicas y donde es importante que todos los paquetes sean recibidos y lleguen hacia su destino, no es factible utilizar a Contiki OS como una solución.

6.2.3 Sobre el desarrollo de la red inalámbrica de sensores utilizando TIMAC.

De acuerdo a los resultados que se obtuvieron tanto en la simulación como en la implementación del escenario de telemonitoreo propuesto, se planteó buscar otra solución en la cual se pueda implementar el estándar 6LoWPAN que no cuente con una capa RDC y que proponga una capa MAC más pura, tal como TIMAC. Al respecto se pueden mencionar las siguientes conclusiones como las más importantes:

- TIMAC, es un kit de desarrollo en software que proporciona la pila de protocolos correspondientes a la capa PHY y MAC del estándar IEEE 802.15.4. Así mismo esta herramienta es soportada por IAR, que proporciona librerías en lenguaje C, lo cual permite que se modifiquen y agreguen líneas de código. Gracias a ello se logró realizar un escenario similar al que se implementó utilizando Contiki OS.
- Los resultados que se obtuvieron de la implementación del escenario de telemonitoreo utilizando TIMAC, permitieron demostrar que efectivamente al utilizar una capa MAC más pura y la omisión de una capa RDC que controla el estado de actividad del radio, no es necesario realizar un número elevado de retransmisiones para que el paquete sea recibido por el nodo central. De esta manera, se puede concluir que empleado a TIMAC como pila de protocolos se podrían tener mejores

resultados, y que en su caso sería posible tener una implementación de un escenario de telemonitoreo.

6.3 Trabajo a futuro

Aunque se cumplieron con los objetivos de este proyecto de investigación, el trabajo realizado representa un acercamiento a lo que puede explotarse para el despliegue de una implementación del estándar 6LoWPAN en un escenario de telemonitoreo. En esta sección se muestran las recomendaciones para el trabajo a futuro que pueden extenderse a partir de este trabajo de tesis.

- Evaluar el desempeño de red 6LoWPAN propuesta, considerando otro perfil de tráfico, de tal manera que se pueda verificar para que escenarios de telemonitoreo es posible utilizar a Contiki OS como solución.
- Analizar el efecto que tiene el empaquetar o no las muestras de las variables fisiológicas contemplando la sobrecarga de la pila de protocolos completa del estándar 6LoWPAN, utilizando el escenario de telemonitoreo propuesto y las métricas correspondientes para la evaluación.
- Añadir la conexión hacia un servidor externo, de tal manera que los datos que se están monitoreando puedan ser recibidos en una PC remota y de igual manera evaluar el desempeño de la red.
- Evaluar el desempeño de la red utilizando la propuesta de TIMAC, en términos del número de paquetes recibidos/perdidos y de usuarios que pueden ser atendidos.
- Realizar un estudio completo del consumo de potencia en el escenario de monitoreo propuesto utilizando TIMAC, de tal manera que se pueda conocer si es conveniente realizar su implementación.
- Implementar los protocolos de comunicación de las capas superiores dentro de TIMAC, con la finalidad de armar la pila de protocolos completa del estándar 6LoWPAN. De tal manera que no sea necesario emular la carga de los mismos, como se realizó en este trabajo de investigación. A partir de ello, realizar una comparación con lo que se obtuvo al utilizar Contiki OS.

Literatura citada

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials*, 17(4), 2347–2376. <https://doi.org/10.1109/COMST.2015.2444095>
- Alinejad, A., Philip, N., & Istepanian, R. S. H. (2011). Mapping of multiple parameter m-health scenarios to mobile WiMAX QoS variables. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 1532–1535. <https://doi.org/10.1109/IEMBS.2011.6090447>
- Babu, H. S. R., & Dey, U. (2014). Routing Protocols in IPv6 enabled LoWPAN : A Survey, 4(2), 2–7.
- Bertocco, M., Gamba, G., & Sona, A. (2008). Is CSMA/CA really efficient against interference in a wireless control system? an experimental answer. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA* (pp. 885–892). <https://doi.org/10.1109/ETFA.2008.4638501>
- Cama Pinto, A., De la Hoz Franco, E., & Cama Pinto, D. (2012). Las redes de sensores inalámbricos y el internet de las cosas. *Inge Cuc*, 8(1), 163–172. Retrieved from <http://dialnet.unirioja.es/servlet/articulo?codigo=4869014&info=resumen&idioma=SPA>
- Casillas, M. (2012). *Diseño de una red híbrida de dos saltos para aplicaciones de redes de sensores en telemonitoreo y telemedicina*.
- Contiki: The open source operating system for the internet of things. (2015). Retrieved from <http://www.contiki-os.org/index.html>
- Dodd, A. (2015). CCSnifer. Retrieved June 5, 2016, from <https://github.com/andrewdodd/ccsniffpiper>
- Dron, W., Duquennoy, S., Voigt, T., Hachicha, K., & Garda, P. (2014). An emulation-based method for lifetime estimation of wireless sensor networks. In *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2014* (pp. 241–248). <https://doi.org/10.1109/DCOSS.2014.10>
- Dunkels, A. (2011). *The ContikiMAC Radio Duty Cycling Protocol*. SICS Technical Report T2011:13 , ISSN 1100-3154. Retrieved from <http://dunkels.com/adam/dunkels11contikimac.pdf>
- Dunkels, A., Grönvall, B., & Voigt, T. (2004). Contiki - A lightweight and flexible operating system for tiny networked sensors. *Proceedings - Conference on Local Computer Networks, LCN*, 455–462. <https://doi.org/10.1109/LCN.2004.38>
- Felsche, M., Huhn, A., & Schwetlick, H. (2012). Routing protocols for 6LoWPAN. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering* (Vol. 82 LNICTS, pp. 71–83). https://doi.org/10.1007/978-3-642-32304-1_7
- Foster, A. (PrismTech). (2014). *Messaging Technologies for the Industrial Internet and the Internet of Things (white paper)*. Retrieved from <http://www.prismtech.com/sites/default/files/documents/MessagingComparsionMarch2014USROW-final.pdf>
- Hartmann, E. (2003). ECG front-end design is simplified with microconverter. *Analog Dialogue*, 1–5.

Retrieved from http://d1.amobbs.com/bbs_upload782111/files_46/ourdev_680028HHNGNM.pdf

- Ilyas, M., & Ahmed, S. (2011). *MAC Protocol for Wireless Network on Chip Master 's Thesis in Computer Systems Engineering. Electrical Engineering.*
- International Organization for Standardization (ISO). (1996). Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. *International Standard ISO/IEC 7498-1*. Retrieved from [http://standards.iso.org/ittf/PubliclyAvailableStandards/s025022_ISO_IEC_7498-3_1997\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/s025022_ISO_IEC_7498-3_1997(E).zip)
- Ismail, N. H. A., Hassan, R., & Ghazali, K. W. M. (2012). A study on protocol stack in 6lowpan model. *Journal of Theoretical and Applied Information Technology*, 41(2), 220–229.
- Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., & Alonso-Zarate, J. (2015). A Survey on Application Layer Protocols for the Internet of Things. In *Transaction on IoT and Cloud Computing* (Vol. 3, pp. 11–17). <https://doi.org/10.5281/ZENODO.51613>
- Kermajani, H. R., & Gomez, C. (2011). Route change latency in low-power and lossy wireless networks using RPL and 6LoWPAN Neighbor Discovery - Kermajani, Gomez.pdf, 937–942.
- Laine, T. H., Lee, C., & Suk, H. (2014). Mobile Gateway for Ubiquitous Health Care System Using ZigBee and Bluetooth. In *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (pp. 139–145). <https://doi.org/10.1109/IMIS.2014.17>
- Malindi, P. (2011). QoS in Telemedicine. In *Telemedicine Techniques and Applications*. <https://doi.org/10.5772/47751.hancement>
- Matús, H. (2007). *Connecting Wireless Sensor Networks to the Internet – a 6lowpan Implementation for TinyOS 2.0*. Jacobs University Bremen.
- Mehmood, W., Hassan, A., Tabish, R., Touati, F., Mnaouer, A. Ben, & Gaabab, B. (2014). Performance Evaluation of 6LoWPAN Based Networks for Ubiquitous Health Monitoring System.
- Michel, M., & Quoitin, B. (2016). *Technical Report : ContikiMAC vs X-MAC performance analysis. arXiv:1404.3589v6 [cs.N]*. Retrieved from <http://arxiv.org/abs/1404.3589>
- Misic, J., & Misic, V. B. (2009). Bridging between IEEE 802.15. 4 and IEEE 802.11 b networks for multiparameter healthcare sensing. *Selected Areas in Communications, IEEE*, 27(4), 435–449. <https://doi.org/10.1109/JSAC.2009.090508>
- Misic, J., & Misic, V. B. (2010). Bridge performance in a multitier wireless network for healthcare monitoring. *IEEE Wireless Communications*, 17(1), 90–95. <https://doi.org/10.1109/MWC.2010.5416355>
- Nedelcu, A., Sandu, F., Machedon-Pisu, M., Alexandru, M., & Ogrutan, P. (2009). Wireless-based remote monitoring and control of intelligent buildings. *2009 IEEE International Workshop on Robotic and Sensors Environments*, 47–52. <https://doi.org/10.1109/ROSE.2009.5355999>
- Olsson, J. (2014). *6LoWPAN demystified*. Retrieved from <http://www.ti.com/lit/wp/swry013/swry013.pdf>
- Österlind, F., Dunkels, A., Eriksson, J., Finne, N., & Voigt, T. (2006). Cross-level sensor network simulation with COOJA. *Proceedings - Conference on Local Computer Networks, LCN*, 641–648.

<https://doi.org/10.1109/LCN.2006.322172>

- Paksuniemi, M., Sorvoja, H., Alasaarela, E., & Myllyla, R. (2005). Wireless sensor and data transmission needs and technologies for patient monitoring in the operating room and intensive care unit. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference, 5*, 5182–5. <https://doi.org/10.1109/IEMBS.2005.1615645>
- Pešovi, U. M., Mohorko, J. J., Benki, K., & Žarko, F. Č. (2010). Single-hop vs . Multi-hop – Energy efficiency analysis in wireless sensor networks.
- Pham-Huu, D.-N., Nguyen, V.-H., Trinh, V.-A., Bui, V.-H., & Pham, H.-A. (2015). Towards an Open Framework for Home Automation Development. In *2015 International Conference on Advanced Computing and Applications (ACOMP)* (pp. 75–81). <https://doi.org/10.1109/ACOMP.2015.28>
- Pizzutti, G. P., Cifaldi, S., & Nolfi, G. (1985). Digital sampling rate and ECG analysis. *Journal of Biomedical Engineering, 7*(3), 247–250. Retrieved from <http://eprints.sics.se/2296>
- Shelby, Z., & Bormann, C. (2011). *6LoWPAN: the wireless embedded internet* (Vol. 43).
- Skorin-Kapov, L., & Matijasevic, M. (2010). Analysis of QoS requirements for e-Health services and mapping to evolved packet system QoS classes. *International Journal of Telemedicine and Applications, 2010*. <https://doi.org/10.1155/2010/628086>
- Stehlík, M. (2011). *Comparison of Simulators for Wireless Sensor Networks*.
- Touati, F., Erdene-Ochir, O., Mehmood, W., Hassan, A., Mnaouer, A. Ben, Gaabab, B., ... Khrijji, L. (2015). An experimental performance evaluation and compatibility study of the bluetooth low energy based platform for ECG monitoring in WBANs. *International Journal of Distributed Sensor Networks, 2015*. <https://doi.org/10.1155/2015/645781>
- Touati, F., Mnaouer, A. Ben, Erdene-Ochir, O., Mehmood, W., Hassan, A., & Gaabab, B. (2015). Feasibility and performance evaluation of a 6LoWPAN-enabled platform for ubiquitous healthcare monitoring. *Wireless Communications and Mobile Computing, (February 2015)*, 421–430. <https://doi.org/10.1002/wcm>
- Touati, F., Tabish, R., & Ben Mnaouer, A. (2013). Towards u-health: An indoor 6LoWPAN based platform for real-time healthcare monitoring. *Proceedings of 2013 6th Joint IFIP Wireless and Mobile Networking Conference, WMNC 2013*. <https://doi.org/10.1109/WMNC.2013.6548958>