



UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO



MEDICIÓN DEL IMPACTO DE LA CAPA RDC EN EL ENRUTAMIENTO SOBRE EL ESTÁNDAR 6LOWPAN

Tesis que presenta

GARCIA LOYA EDUARDO

como requisito para obtener el grado de

Ingeniero en Electrónica

Co-Director de tesis: **Dr. Gabriel Alejandro Galviz Mosqueda**

Co-Director de tesis: **Dr. Humberto Cervantes de Avila**

Ensenada Baja California, México, Agosto 2018

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

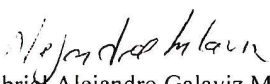
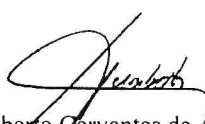
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

**MEDICIÓN DEL IMPACTO DE LA CAPA RDC EN EL ENRUTAMIENTO
SOBRE EL ESTÁNDAR 6LOWPAN****TESIS**

Que para obtener el grado de ingeniero en electrónica presenta:

Eduardo Garcia Loya

Aprobada por:

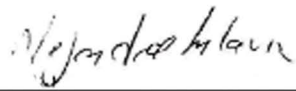

Dr. Gabriel Alejandro Galaviz Mosqueda
Co-Director de tesis
Dr. Humberto Cervantes de Ávila
Co-Director de tesis
Dr. Juan Iván Nieto Hipólito
Miembro del comité
M.C. Edwin Martínez Aragón
Miembro del comité
Dr. José Antonio Michel Macarty
Miembro del comité

Ensenada Baja California, México. Septiembre 2018

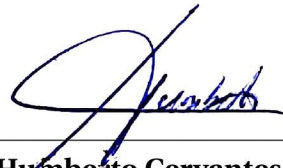
RESUMEN de la tesis de **Eduardo Garcia Loya**, presentada como requerimiento para, obtener el grado de Ingeniero en Electrónica, del programa de licenciatura de la Universidad Autónoma de Baja California, Ensenada, Baja California, México. Agosto 2018.

**MEDICIÓN DEL IMPACTO DE LA CAPA RDC EN EL ENRUTAMIENTO SOBRE EL
ESTÁNDAR 6LOWPAN**

Resumen Aprobado por:



Dr. Gabriel Alejandro Galviz Mosqueda



Dr. Humberto Cervantes de Avila

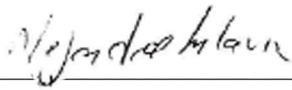
El Internet de las cosas (IoT por sus siglas en inglés) puede habilitar aplicaciones enfocadas a diversos rubros y potenciar su impacto[1]. Por ejemplo, en caso de un accidente o desastre natural, los cuerpos de emergencia podrían monitorizar a los heridos utilizando sensores de variables fisiológicas a través de una red inalámbrica multi-salto de sensores (WSN por sus siglas en inglés). La información recolectada de cada persona se podría enviar de manera continua hacia el equipo de emergencia o a una central de urgencias médicas. Directamente si los sensores están cerca del resumidero de información, o a través de otros sensores hasta llegar al resumidero. Se debe considerar que los dispositivos que recolectan las señales que se enviaran requieren ser móviles y vestibles, por lo que uno de los principales problemas es el consumo de energía necesario para realizar la comunicación inalámbrica. Debido a esto, es sumamente importante que los protocolos de comunicación tengan un desempeño eficiente que permita enviar la mayor cantidad de carga útil utilizando la menor cantidad de paquetes de control. En este sentido, el protocolo RPL ha mostrado ser eficiente para este tipo de redes, y es considerado como el estándar de facto. De manera adicional, el radio de comunicaciones es uno de los que componentes que mayor energía consumen. Por lo que, una de las estrategias más utilizadas para disminuir el consumo de energía de los nodos es introducir un ciclo de trabajo para el radio (RDC por sus siglas en inglés) de los nodos. Sin embargo, esta capa de RDC puede introducir retardos para el envío de paquetes o pérdida de los mismos debido a que la frecuencia del ciclo de trabajo del transmisor y el receptor pueden

no estar sincronizados. Lo que podría ocasionar una mayor sobrecarga debido a que dichos paquetes deben ser retransmitidos. Por lo tanto, si bien es cierto que la implementación de un mecanismo que se asegure la mayor parte del tiempo de mantener el radio de los dispositivos en un estado apagado o de bajo consumo reduciría en mayor parte el consumo energético total del mismo se tienen que tomar en cuenta todos los mecanismos superiores y como estos se desempeñan para asegurarse de que no exista un impacto negativo en el comportamiento deseado del dispositivo. En este trabajo de tesis se mide de manera experimental el impacto que presenta incorporar una capa de RDC o un mecanismo que permite variar el ciclo de trabajo del radio de dispositivos en el algoritmo de enrutamiento prominente para el IoT, RPL. La cama de pruebas implementada se basa en el estándar 6LoWPAN, uno de los más utilizados en la industria para llevar a cabo implementaciones de IoT. Dicho estándar se implementó en el simulador de redes COOJA y el sistema operativo CONTIKI . El desempeño de RPL es medido utilizando métricas como PDR y número de retransmisiones. Los resultados demuestran que una capa de RDC debajo del algoritmo de enrutamiento RPL afecta en gran medida su desempeño, demostrándose en el aumento del número de retransmisiones y la pérdida de PDR para la mayoría de los casos analizados.

Abstract of the thesis presented by **Eduardo Garcia Loya**, as a requirement to obtain the bachelor degree in electronics engineering, of the program of Bachelor's degree of the Universidad Autonoma de Baja California, Ensenada Baja California, Mexico. Agosto 2018

**MEASUREMENT OF THE IMPACT OF THE RDC LAYER ON ROUTING IN THE 6LOWPAN
STANDARD**

Abstract approved by:



Dr. Gabriel Alejandro Galviz Mosqueda



Dr. Humberto Cervantes de Avila

The internet of things or 'IoT' can help us make applications focused in diverse fields and potence their impact [1]. For example, in the case of an accident or natural disaster , the emergency service can monitor the injured using sensor for fisiologic variables through a multi-hop wireless network or 'WSN'. The collected data of each person can then be send in a continuous way to the emergency team. Directly if the sensors are close to the sink, or through the others node in the network. It need to be considered that the devices recollecting the signals required to be wearable, making one of the principal problems the consumed energy needed to do the wireless communication, because of this , It is extremely important that communication protocols have a efficient performance that allows to send the largest amount of payload using the least amount of control packages. In this sense, the RPL protocol has been shown to be efficient for this type of networks, and is considered the de facto standard. Additionally, the communications radio is one of the components that consume the most energy. One of the most used strategies to reduce the energy consumption of the nodes is to introduce a work cycle for the radius (RDC) of the nodes. However, this RDC layer can introduce delays for sending packets or loss of packets because the duty cycle frequency of the transmitter and the receiver may not be synchronized. This could cause a greater overload due to the fact that said packages must be retransmitted. Therefore, it is true that the implementation of a mechanism that ensures most of the time of maintaining the radius of the devices in a null or low consumption state would reduce in a large part the total energy

consumed, but on the same way, it need have to take into account all the superior mechanisms and how they work to ensure that there is no negative impact on the desired behavior of the device. In this thesis, the impact of incorporating an RDC layer or a mechanism that varies the working cycle of the device radius in the prominent routing algorithm for the IoT RPL is experimentally measured. The test bed implemented is based on the 6LoWPAN standard, one of the most used in the industry to carry out IoT implementations. This standard was implemented in the COOJA network simulator and the CONTIKI operating system. The performance of RPL is measured using metrics such as PDR and number of retransmissions. The results shows that a layer of RDC under the RPL routing algorithm greatly affects its performance, demonstrating by increase in the number of retransmissions and the loss of PDR for most of the analyzed cases.

DEDICATORIA

A mi familia

A mis padres por estar siempre apoyándome y enseñándome cosas nuevas en el día a día, siempre preocupándose por mi; a mi hermano menor Eliaf por estar conmigo, como hermano, compañero de juegos y amigo; a mis abuelos y abuelas, tíos y tías que me apoyaron y a todos aquellos que participaron directa o indirectamente en la elaboración de esta tesis.

¡Gracias a ustedes!

Al Amor

Finalmente, a mi novia Carolina, has estado conmigo incluso en los momentos más duros de este trabajo, no fue fácil, pero siempre me motivas a continuar, no darme por vencido y dar lo mejor de mi, muchas gracias Nya Nya.

AGRADECIMIENTOS

A la Universidad Autónoma de Baja California por brindarme las herramientas necesarias durante mi formación profesional para mi crecimiento académico y personal. Al CICESE por mostrarme el camino de la investigación y abrirme las puertas para la realización de esta tesis. Me voy con la alegría de haber podido realizar mis estudios universitarios en lo que considero las mejores instituciones para hacerlo.

Quiero agradecer a mis co-directores de tesis, el Dr. Gabriel Alejandro Galaviz Mosqueda y al Dr. Humberto Cervantes De Ávila por su orientación, guía y apoyo durante mi investigación.

A los miembros de mi comité de tesis: Dr. Juan Iván Nieto Hipólito, MC. Edwin Martínez Aragón y el MC. Michel Macarty Jose A. por sus consejos, comentarios y observaciones que ayudaron a mejorar y enriquecer este trabajo de investigación.

Un especial agradecimiento a los integrantes del grupo ARTS, en particular al M.C. Christian Soto por permitirme ingresar a su área de trabajo y poder observar de cerca cómo es que se desarrollan ideas y proyectos como nunca había visto, así como guiarme en este corto tiempo que compartí con ellos.

A mis padres y mi hermano y mi novia, sé que siempre podré contar con ellos en cualquier situación. Les debo mis logros y la persona que soy.

A todos mis amigos Que nos apoyamos mutuamente en nuestra formación profesional y que sin lugar a dudas, los considero como familia: Carlos Vasquez alias 'fafillo', Jose Carlos Lozano alias 'el Jazz', Oscar Atondo alias 'Franky boy', Alexis Crespo alias el 'Kreppo', Daniel Murillo alias 'Dany', David Riaño alias 'Remix' y Stephanie Ochoa alias 'Fany'.

Índice general

1. Introducción	1
1.1. IoT	1
1.2. Planteamiento de problema	3
1.3. Objetivo general	4
1.4. Objetivos particulares	4
1.5. Metodología	4
1.6. Organización de la tesis	5
2. IoT, 6LoWPAN Y RPL	6
2.1. Introducción	6
2.2. Internet of Things	6
2.3. Internet de las cosas Medicas (M-IoT)	7
2.4. IPV6	8
2.5. RPL	8
2.6. IPV6 Y LoWPANs	9
3. CONTIKI Y SU CAPA RDC, SIMULADOR DE REDES COOJA	11
3.1. Introducción	11
3.2. CONTIKI	11
3.3. CAPA RDC DE CONTIKI	12
3.4. COOJA	13
3.5. SIMULACION DE LA CAPA RDC EN CONJUNTO CON EL PRTOCOLO RPL . . .	14

<i>ÍNDICE GENERAL</i>	<i>x</i>
3.5.1. ESCENARIOS DE SIMULACION PROPUESTOS	16
3.6. CAPTURA DE METRICAS	17
4. ANALISIS DEL IMPACTO DE UNA CAPA RDC SOBRE EL PROTOCOLO RPL	19
4.1. Introducción	19
4.2. Definición de las métricas a evaluar	20
4.3. Análisis del desempeño de la red solo con el protocolo RPL activo	20
4.4. RPL con modificaciones propuestas por CONTIKI-OS	23
4.5. Resultados con RPL y la capa RDC activa de CONTIKI-OS	26
4.5.1. Frecuencia de envío 1 HZ	27
4.5.2. Frecuencia de envío 5 HZ	30
4.5.3. Frecuencia de envío 10 HZ	33
4.6. Conclusiones del análisis del efecto del ciclo de trabajo sobre el protocolo RPL	35
5. CONCLUSIONES Y TRABAJO FUTURO	37
5.1. Introducción	37
5.2. Conclusiones	37
5.3. Trabajo Futuro	39
Anexo I: Diagrama de flujo para la realización de simulaciones continuas	40
Bibliografía	41

Índice de figuras

1.1. Diagrama que engloba la descripción del IoT (apptricity 2018)	1
2.1. Diagrama del modelo OSI	10
3.1. Imagen del contenido de ejemplo de un archivo 'project-conf.h'	15
3.2. Topología de la simulación de red de 6LoWPAN, en un escenario multisalto. . .	16
3.3. Herramienta de LOG de COOJA	18
4.1. PDR normalizado de la red variando el numero de saltos.	21
4.2. Tiempo en segundos del retraso promedio del arribo de paquetes al sumidero dado el numero de saltos realizados	22
4.3. Cantidad de paquetes retransmitidos dado el numero de saltos realizados . . .	23
4.4. PDR normalizado de la red dado el numero de saltos.	24
4.5. Retraso de los paquetes dado el numero de saltos realizados	25
4.6. Cantidad de paquetes retransmitidos dado el numero de saltos realizados . . .	26
4.7. Herramienta 'Radio Messages' de COOJA en la cual se muestran los paquetes generados por los nodos de la red.	27
4.8. PDR normalizado de la red dado el numero de saltos.	28
4.9. Retraso de los paquetes dado el numero de saltos realizados.	28
4.10. Cantidad de paquetes retransmitidos dado el numero de saltos realizados. . . .	29
4.11. PDR normalizado de la red dado el numero de saltos.	30
4.12. Retraso de los paquetes dado el numero de saltos realizados.	31
4.13. Cantidad de paquetes retransmitidos dado el numero de saltos realizados. . . .	32
4.14. PDR normalizado de la red dado el numero de saltos.	33
4.15. Retraso de los paquetes dado el numero de saltos realizados.	34
4.16. Cantidad de paquetes retransmitidos dado el numero de saltos realizados. . . .	35

Índice de tablas

3.1. Tabla de valores asignados a las variables para las simulaciones	17
---	----

Capítulo 1

Introducción

1.1. IoT

Desde la invención del Internet este no ha parado de crecer, sin embargo su crecimiento no solo se debe a la cantidad de población humana que hace uso del mismo si no también a los dispositivos conectados que intercambian información a través de Internet, línea blanca como refrigeradores y lavadoras hasta cafeteras y televisiones (*Defining The Internet of Things: A Work In Progress*, s.f.). El concepto de 'IoT' no está rígidamente ligado a un solo camino de desarrollo, por lo cual , podemos encontrar movimientos dirigidos a hogares inteligentes, oficinas inteligentes, ciudades inteligentes y aplicaciones para la salud y bienestar del ser humano (Mohammadi, Al-Fuqaha, Sorour, y Guizani, s.f.).

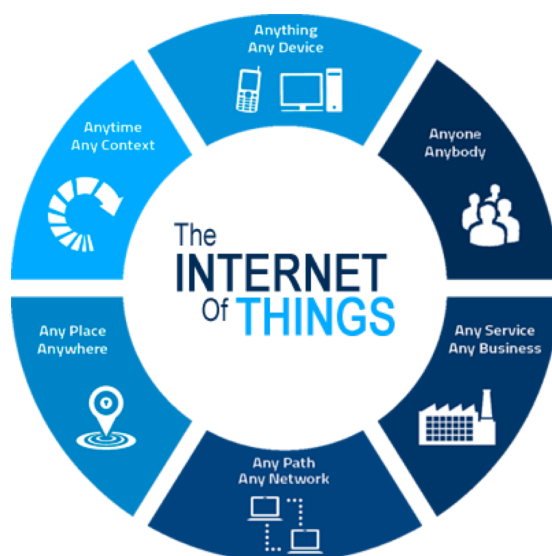


Figura 1.1: Diagrama que engloba la descripción del IoT (apptricity 2018)

Las tecnologías que se utilizan actualmente para llevar a cabo implementaciones de IoT son 'Bluetooth low energy' o BLE introducido en la versión 4.0 de Bluetooth, el cual ofrece una alternativa de bajo consumo a los dispositivos que utilizan este estándar de comunicación(SIG INTRODUCES BLUETOOTH LOW ENERGY WIRELESS TECHNOLOGY THE NEXT GENERATION OF BLUETOOTH WIRELESS TECHNOLOGY | *Bluetooth Technology Website*, s.f.), así como, el protocolo 'ZigBee' que cuenta con versiones actualizadas para aplicaciones IoT. El grupo de trabajo 6lo de la IETF('Internet Engineering Task Force') propuso un estándar abierto que provee una pila de protocolos, los cuales están enfocados a la comunicación efectiva entre de redes con IPV6 de un lado y diversas redes de comunicación por el otro, llamado 6LoWPAN o 'IPV6 over Low-Power Wireless Personal Area Networks'. Debido a que 6LoWPAN define las capas superiores dejando las capas inferiores abiertas a distintas tecnologías de comunicación se ha optado por utilizar IEEE 802.15.4 en la mayoría de los casos debido a sus características pensadas para dispositivos de bajo consumo energético y bajo procesamiento (*6LoWPAN overview - ARM mbed*, s.f.).

Durante los últimos años han surgido modificaciones para los protocolos y estándares anteriormente mencionados, así como propuestas para las 'capas' superiores como tal es el caso de 'IPv6 Routing Protocol for Low-Power and Lossy Networks' o simplemente RPL" (Schumacher, Kushalnagar, y Montenegro, s.f.) el cual define una técnica de enrutamiento pensada para dispositivos de bajas prestaciones como lo son bajas tasas de transferencia de información, interconexiones con enlaces débiles o que sus patrones de tráfico no está definido y pueden ser punto-a-punto, punto-a-multipunto o multipunto-a-punto.

Con el surgimiento de RPL los sistemas operativos para IoT con pilas de software o 'stacks' ya desarrolladas empezaron a adoptar las nuevas tecnologías como fue el caso de CONTIKI (*Contiki: The Open Source Operating System for the Internet of Things*, s.f.) que a la par de implementar el protocolo RPL para el enrutamiento también implementó una mejora denominada 'capa RDC' o 'capa de ciclo de trabajo del radio' con su protocolo 'ContikiMAC' (Dunkels, s.f.) basado en que el radio de los dispositivos no necesita estar en la totalidad del tiempo encendido, si no, mantener encendido un periodo de tiempo cada cierta cantidad de milisegundos ayudando de esta manera a aumentar el tiempo de vida del dispositivo.

Las investigaciones recientes se enfocan en la mayor parte en como un escenario con el protocolo RPL la implementación de 'ContikiMac' mantienen la vida de la red (Liu, Sheng, Yin, Ali, y Roggen, s.f.) así como trabajos donde se plantean ciclos de trabajo adaptativos para los nodos de la red (Youssef, Elsayed, y Zahran, s.f.) sin embargo un punto importante de investigación es cómo estas nuevas implementaciones con una capa de RDC.^a afectan a las capas superiores como lo son la capa de enrutamiento que para el caso de IoT para muchos

casos es RPL considerando hasta qué punto es ahorrrativo o factible añadir una capa de ciclo de trabajo para el radio, considerando siempre las limitaciones que esta puede añadir.

1.2. Planteamiento de problema

Debido a el potencial, beneficios y desafíos que presenta el paradigma del IoT, en años recientes ha habido un crecimiento en la implementacion de sistemas de recolección de información bajo este paradigma. Esto tanto en la comunidad de investigadores y empresas privadas por igual. Por ejemplo, utilizando el paradigma del IoT pueden habilitarse aplicaciones como capturar y enviar grandes cantidades de información de una planta industrial con dispositivos de bajo costo, o aplicaciones tan relevantes como monitorear los comportamientos de pacientes dentro del hospital.

Como tal la transmisión de información de cualquier tipo no es algo trivial, debido a todas las variables que se tienen que considerar para poder llevar a cabo una comunicación exitosa de la información que se desea transmitir, como por ejemplo el canal o medio donde estarán interactuando el transmisor y el receptor, así como también la cantidad de información que se desea transmitir y el tiempo en el que se desea recibir. De la misma manera si los dispositivos con los que se trabaja están sujetos a una fuente limitada de energía para trabajar y si se encontraran en movimiento durante su operación.

La gran parte de los trabajos de investigación realizados hasta la fecha utilizan redes 6LoWPAN, y por el lado de software de simulación se utilizan OMNET++, COOJA simulador, TOSSIM, Ns-2, NS-3 y Jsim (Liu y cols., s.f.), así como, sistemas operativos con "6LowPAN ready" como CONTIKI siendo tanto de código abierto como orientado a redes de sensores inalámbricos (Reusing, s.f.), otras alternativas también de código abierto son TinyOS el cual a diferencia de CONTIKI el cual se encuentra programado en el lenguaje C este su base es el lenguaje Nest y se organiza en componentes (x Working Group, s.f.), a finales del 2017 se lanzo mbed-OS un sistema operativo '6LowPAN READY' por parte de 'arm' que esta pensado en la nube y 'programar una sola vez para muchos sistemas embebidos para luego controlarlos en la nube' (*What is Mbed? | Mbed*, s.f.). Estos sistemas operativos incorporan capas únicas de cada uno como es el caso de CONTIKI que al estándar 6LowPAN le incluye una subcapa RDC" para el control del ciclo de trabajo del radio del dispositivo con el fin de reducir el consumo energetico de los nodos de la red.

Hoy en día se tienen herramientas de simulación con protocolos pensados para el IoT y software robustos como sistemas operativos ayudan bastante para poder probar y hacer las

implementaciones de manera rápida, tenemos que asegurarnos que los protocolos estandarizados como el 6LowPAN o IPV6 se comporten de manera correcta ante capas agregadas como la capa 'RDC' de Contiki-oS o las APis de seguridad de mbed-OS, por lo cual sin tomar en cuenta estas capas agregadas los resultados obtenidos pueden estar sobrestimados por cual hace necesario comenzar a cuantificar el efecto del ciclo de trabajo del radio en el desempeño de los algoritmos.

1.3. Objetivo general

El objetivo de esta investigación es medir experimentalmente el impacto del ciclo de trabajo del radio de comunicación en el enrutamiento de RPL sobre escenarios de IoT en términos de métricas de calidad de servicio como paquetes entregados, cantidad de retransmisiones y tiempo de retraso desde que el paquete es enviado por el transmisor hasta que llega a su destino. Esto se realizara bajo el estándar 6LowPAN utilizando el simulador Cooja.

1.4. Objetivos particulares

- Implementar una cama de prueba experimental con nodos que utilicen el estándar 6LoWPAN.
- Implementar el algoritmo de enrutamiento RPL para redes 6LoWPAN.
- Medir con la cama de pruebas implementada el impacto de diferentes ciclos de trabajo del transceptor de radio sobre el algoritmo de enrutamiento en términos de tasa de paquetes entregados y número de retransmisiones.

1.5. Metodología

1. Revisión de bibliografía.

- Debido a que se necesita información sobre los algoritmos actuales utilizados en las redes 6LowPAN..

2. Definición del escenario de evaluación.

- En esta parte se planteara y definirá el escenario de evaluación e incluirá algoritmo de enrutamiento.

3. Generación de la cama de pruebas.
 - Se seleccionarán las herramientas necesarias para desarrollar una cama de pruebas la cual tendrá la finalidad de contener el escenario de evaluación.
4. Definición de métricas a considerar en la evaluación del algoritmo de enrutamiento.
 - Se definirán las métricas a considerar necesarias para llevar a cabo la evaluación del algoritmo de enrutamiento.
5. Obtención y análisis de resultados.
 - Tras llevar a cabo el escenario planteado y la evaluación del algoritmo de enrutamiento se obtendrán los resultados y se procederá a evaluar los mismos.
6. Generar conclusiones a partir del análisis de resultados.
 - En esta parte se generarán las conclusiones a partir del análisis de los resultados obtenidos en el escenario evaluado con las métricas definidas anteriormente.

1.6. Organización de la tesis

En el capítulo 2 se describen las redes de IoT con las características de los protocolos base de la investigación 6LoWPAN y RPL, ya que son fundamentales para el análisis de los resultados, por último se mencionan los resultados de la bibliografía analizada para definir el escenario de evaluación.

En el capítulo 3 se presenta el sistema operativo CONTIKI y su implementación de la capa 'RDC', así como también la información sobre las herramientas utilizadas para llevar a cabo las simulaciones de los escenarios que se proponen someter a prueba, el simulador de redes COOJA, parámetros y descripciones de las simulaciones, así como la implementación de la red 6LoWPAN en el escenario de envío de información dado un número de saltos deseado junto con las herramientas de análisis para las simulaciones.

En el capítulo 4 se describen las métricas deseadas a capturar, presentan los resultados obtenidos de las simulaciones realizadas, variando el escenario en número de saltos realizados por el transmisor de la red para llegar a un nodo central denominado 'sumidero', variando el ciclo de trabajo de la capa RDC de CONTIKI para todos los nodos y comparando con diferentes tasas de transmisión del nodo transmisor de la red.

En el capítulo 5 como última parte, se plasman las conclusiones y el trabajo futuro que surge del desarrollo de esta investigación.

Capítulo 2

IoT, 6LoWPAN Y RPL

2.1. Introducción

En la actualidad existe un paradigma tecnológico llamado 'Internet Of Things' o 'IoT' , el cual permite lograra proyectos que hasta unos cuantos años atrás parecían muy difíciles de desarrollar. Con la oleada de proyectos 'IoT' comenzaron a diversificarse y madurar ramas como el 'M-IoT', el cual adapta el 'IoT' para proyectos médicos. Los organizamos como la IEEE, toma parte en la estandarización y la creación de protocolos como 'RPL' y '6LoWPAN' y son necesarios para que exista una integración y unificación de estos para aplicaciones. En este capítulo se hace énfasis la descripción de los paradigmas 'IoT' y 'M-IoT' así como los protocolos 'RPL' y '6LoWPAN'.

2.2. Internet of Things

El Internet of Things o IoT inicia con los trabajos de 'Xrox Parc' durante los años de 1990 (McEwen y Cassimally, s.f.), su trabajo ayudó en la época, a intentar comprender un escenario donde el poder de cómputo bajaba sus costos y podía ser incluido en los objetos de la vida diaria, definiendo en el tiempo de su trabajo a este paradigma como "Ubiquitous computing". Tras pasar el tiempo en 1994 Reza Raji describe en sus trabajos el cómo utilizar pequeños paquetes de información pero en un red con muchos dispositivos conectados, la cual puede lograr integrar y automatizar desde aplicaciones para un hogar hasta fábricas completas (Raji, s.f.). Pero no es hasta el 2010 que Dave Evans en "The Internet of Things How the Next Evolution of the Internet Is Changing Everything" (Evans, s.f.) describe el IoT como un paradigma que

presenta un salto gigante en la forma de recolección, análisis y distribución de contenido, que puede ser convertido en información importante y concisa.

Las aplicaciones para IoT pueden ser tan variadas como y extensas como seguridad para hogares (Kang, Moon, y Park, s.f.), mejoramiento de producción agrícola (Meola, s.f.) hasta monitorización de ciudades completas. Actualmente compañías privadas como 'Amazon' introducen productos al mercado pensados en IoT como 'Alexa', producto el cual permite automatizar acciones dentro del hogar como encender el televisor, conectarse con otros dispositivos IoT y abrir o cerrar persianas, ver que es lo que falta en tu nevera e interactuar con la calefacción del hogar y ajustar los parámetros de manera automática.

Tal proliferación de dispositivos de IoT está acompañada con retos que se tienen que resolver, tal como el abastecimiento de energía en caso de que sean móviles, como es que se comunican los unos a los otros y cómo se comunican con el usuario final. Los retos tecnológicos y de redes están forzando a invenciones como el estándar 6LoWPAN (Shelby, s.f.). Creado por un grupo perteneciente a la IEEE, incorpora la última versión del protocolo de Internet IPv6 en diseños de redes pensados para dispositivos con bajo consumo, poder de procesamiento limitado y que usualmente dependen de una fuente de energía limitada como una batería.

2.3. Internet de las cosas Medicas (M-IoT)

Un enfoque tecnológico a sido introducido recientemente, donde se juntan las funcionalidades del 'm-health' o 'mobile-Health' y 'IoT'. El término 'Mobile-Health' es utilizado para referirse a la aplicación de la medicina soportada por dispositivos móviles, como lo son teléfonos celulares, tabletas electrónicas y relojes inteligentes, el principal objetivo del 'M-Health' es la recolección de información útil a los médicos, investigadores y el envío de instrucciones al paciente, la información consta de los signos vitales del paciente en tiempo real (Istepanian, Hu, Philip, y Sungoor, s.f.-b). En principio el 'M-IoT' introduce un nuevo paradigma de conductividad para el cuidado de la salud, el cual interconecta tecnologías de comunicación basadas en 'IPv6' como lo son '6LowPAN' con redes como 4G.

Las investigaciones y aplicaciones que rodean a 'M-IoT' son muy variadas, desde planteamiento del diseño de sensores para distintas variables, como el nivel de glucosa (Dimitrov, s.f.), electrocardiogramas móviles, así como sistemas de monitorización de más de una variable (Kamel Boulos y Al-Shorbaji, s.f.). Sin embargo, aún hay bastantes problemas que resolver, como el problema de la seguridad de la información (*Can we expect the Internet of Things in healthcare?*, s.f.), muchos de los escenarios planteados alojan la información recopilada

en lo que se conoce como un servidor descentralizado o 'la nube', trayendo consigo problemas de privacidad y seguridad de la información recopilada, algunos intentos por resolver esta problemática se presentan en regulaciones propuestas por administraciones como 'U.S. Food and Drug Administration' ("Manufacturers Sharing Patient-Specific Information from Medical Devices with Patients Upon Request - Guidance for Industry and Food and Drug Administration Staff", s.f.).

2.4. IPV6

El 'Internet Protocol V6' o IPV6 es la versión sucesoria al IPV4, tales protocolos se utilizan para interconectar sistemas por intercambio de paquetes en redes de comunicación, los mismos protocolos proveen bloques de transmisión de información llamados datagramas de la fuente hasta el destino, donde tanto la fuente como el destino son anfitriones identificados por direcciones de longitud fijas (Postel, s.f.). El IPV6 surge de la necesidad de modificar el protocolo IPV4 siendo la principal necesidad el aumento del número de direcciones posibles, pasando de una dirección de 32 bits a una de 128 bits significando un aumento exponencial en las direcciones posibles para los dispositivos .

Entre los beneficios del cambio de versión a IPV6 se encuentran:

- Formato de simplificación del encabezado: Donde algunos campos de los encabezados de IPV6 fueron modificados para ser opciones, reduciendo gasto de recursos para el manejo del paquete así como también limitar el costo del ancho de banda de la cabecera de IPV6
- Capacidad de etiquetas de flujo Esta mejora al protocolo permite el etiquetado de paquetes pertenecientes a un tipo particular de tráfico, donde el transmisor pide que sea manejado de manera específica, como por ejemplo 'servicio en tiempo real'.
- Capacidad de autenticación y privacidad: Haciendo uso de extensiones para el soporte de autenticación, integridad de información y de manera opcional la confidencialidad de la información

2.5. RPL

El enrutamiento es una parte de suma importancia para las redes inalámbricas, hablando de IoT , muchas de estas redes se consideran como "LLNs." "Low-Power and Lossy Networks" las

cuales tienen características como bajo consumo energético, bajo poder de procesamiento, memoria y energía limitadas y muchas veces alimentadas por baterías, así como tasas de transferencia muy bajas (Winter {\textless}wintert@acm.org{\textgreater}, s.f.).

RPL opera con 'objetivos' asignando un número dependiendo de que tan bien se cumple el objetivo definido los cuales pueden ser eficiencia energética, mínima latencia posible o menor número de saltos. Debido a su diseño RPL necesita transmitir paquetes de información para estimar y formar los caminos dentro de la red un mecanismo que cumple con las características necesarias es el documentado en el [RFC6553] el cual inicia el proceso en un sumidero de información y realiza la identificación de los nodos como si fuera una pirámide, donde al pasar un salto agrega una suma al identificador del camino denominado 'RANK' y está sujeto al 'objetivo' con el cual el enrutamiento se está formando, de esto se puede inferir que si se utiliza un 'objetivo' de ahorro de energía no necesariamente pase por el camino más corto o con el menor número de saltos al sumidero de información.

El funcionamiento estructurado en el documento de descripción de 'RPL' menciona que este tiene un mecanismo de envío de paquetes configurado para enviar de manera periódica paquetes de control con los cuales mantiene la información de los nodos vecinos actualizada denominados 'DIO', 'DAOS' y 'DIS'.

- DODAG Information Solicitation (DIS). El 'DIS' es un paquete de control de 'RPL' utilizado para solicitar un 'DODAG' o 'Directed Acyclic Graph' el cual contiene la información de los caminos disponibles para el nodo.
- DODAG Information Object (DIO). Este paquete de control contiene información que permite al nodo descubrir una instancia con RPL y aprender sus configuraciones.
- Destination Advertisement Object (DAO). Este paquete de control es utilizado para propagar información hacia el sumidero de la red, este es un 'unicast' y espera un paquete de 'ACK'.

2.6. IPV6 Y LoWPANs

Dentro de los estándares propuestos en los últimos años con la llegada del IoT se encuentra 6LoWPAN, este se define para los dispositivos que son caracterizados por tener un corto rango de transmisión, baja tasa de transmisión, son de bajo consumo bajo costo. así como también,

poca capacidad de procesamiento, memoria limitada y que frecuentemente están sujetos a una cantidad de energía finita para funcionar (*6LoWPAN overview - ARM mbed*, s.f.),

Las denominadas LowPAN son redes de comunicación simples y de bajo costo las cuales permiten conectividad inalámbrica en dispositivos con energía limitada para trabajar, usualmente se plantean para conectar el entorno físico hacia aplicaciones donde la captura de variables físicas es una parte primordial, pero por lo mismo realizar una transmisión de datos de IP no es una tarea insignificante y con el fin de resolver esta problemática se presenta un modelo de comunicaciones basado en capas llamado 'Open Systems Interconnection' u 'OSI' por sus siglas en inglés, su desarrollo fue un evento que marcó la estandarización de la arquitectura para las redes de computadoras, ya que, resumió toda la experiencia de la época y su implementación se sigue considerándose al día a día (Olifer, Natalia, y Victori, s.f.), su contexto para resolver el problema de la comunicación se puede ver en la figura 3 donde cada sistema tiene siete capas internas o una pila de protocolos.

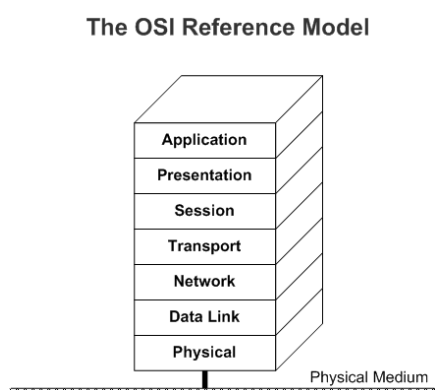


Figura 2.1: Diagrama del modelo OSI

Con la llegada del modelo 'OSI' varios organismos se dedicaron a realizar estándares de comunicación o protocolo los cuales definieron las capas desde su funcionamiento hasta características y uno de los más importantes es el 'Institute of Electrical and Electronics Engineers' o 'IEEE' (*IEEE Standards Education*, s.f.). Dentro de la IEEE existen grupos encargados de diseñar capas para aplicaciones específicas uno de estos es el grupo llamado IEE 802.15.4 el cual se encarga de la definición de las capas denominadas "PHY." Capa Física la capa "MAC." 'Control de Acceso al Medio' dentro de redes WPAN o 'Wireless Personal Area Networks'

La inclusión de varias tecnologías para formar una mayor se puede observar en '6LoWPAN' donde las denominadas 'WPANs' o redes de área personal hacen uso de los estándares que están adoptando por necesidad como el 'IPv6' (Johnson, s.f.), esa fusión de tecnologías fue creada debido a que las tendencias para IoT y la necesidad de conectar dispositivos de bajo costo y bajas capacidades (Shelby, s.f.).

Capítulo 3

CONTIKI Y SU CAPA RDC, SIMULADOR DE REDES COOJA

3.1. Introducción

En esta sección se trata la información del sistema operativo CONTIKI y su pila de protocolos que contiene una capa de RDC para el manejo del radio del dispositivo, el protocolo que se incorpora para esta capa de 'RDC' llamado 'ContikiMac', así mismo se describe el simulador utilizado durante este trabajo de tesis siendo este una herramienta fundamental para el análisis de información de nodos con CONTIKI instalado y de la misma manera se describirán los escenarios de simulación propuestos y la lógica de las herramientas que se utilizaron para obtener la información de las mismas.

3.2. CONTIKI

Un sistema operativo o 'OS' por sus siglas en inglés, es una pieza de software usualmente muy robusto, que controla la ejecución de aplicaciones y sirve como interfaz entre las mismas y el hardware del dispositivo sobre el cual está instalado (Dunkels, Gronvall, y Voigt, s.f.), por lo cual dentro del modelo OSI representa la última capa. El 'OS' se plantea con tres objetivos principales:

- facilidad, con la finalidad que un dispositivo con un sistema operativo instalado sea mas fácil de utilizar.

- eficiencia, un sistema operativo ayuda a utilizar los recursos de hardware de manera eficiente.
- Habilidad de evolucionar, un sistema operativo debe de ser construido de una manera que permita un desarrollo efectivo y sea capaz de introducir nuevas funciones sin interferir con el servicio.

Con el aumento en el uso de redes de sensores inalámbricas se necesitaron implementaciones de software que fueran fácilmente aplicadas a un número considerable de dispositivos, de la misma manera que consideran las características limitadas en capacidad de almacenamiento y procesamiento individual (Reusing, s.f.). Una de las propuestas para resolver esta problemática es CONTIKI un sistema operativo pensado para sistemas con un limitado espacio de almacenamiento (Dunkels y cols., s.f.) creado en el año 2004, una de las características de CONTIKI que lo diferencian de otros sistemas operativos con el mismo fin es que su 'kernel' está basado en eventos por lo que permite utilizar de manera más eficiente el espacio en memoria, de la misma manera también cuenta con librerías que habilitan la posibilidad de ejecutar multiprocesos.

3.3. CAPA RDC DE CONTIKI

Hoy en día CONTIKI es altamente utilizado y estudiado debido a las características únicas que ha adquirido a lo largo del tiempo en sus actualizaciones como lo es la implementación de una capa llamada 'Radio Duty Cycle' o 'RDC' y su protocolo 'ContikiMac' (Dunkels, s.f.). La finalidad de la capa 'RDC' de CONTIKI como su nombre lo indica, controlar dado un ciclo de trabajo el encendido y trabajo del radio con la finalidad de tener un ahorro energético en el dispositivo, esto por su parte añade situaciones que deben de ser atendidas, como el proceso que se tiene que seguir para realizar una comunicación efectiva suponiendo dos casos principales, que el radio de los vecinos en la red se encuentre encendido y cuando no se encuentra encendido.

El protocolo 'ContikiMac' resuelve la problemática de los dos casos presentados anteriormente de la siguiente manera, cuando el dispositivo pasa del estado de radio apagado a un estado de radio encendido censa el medio para detectar si un paquete está siendo transmitido utilizando una herramienta de 'ContikiMac' llamada 'Clear Channel Assessment' o 'CCA' la cual mide en nivel de potencia la energía en el medio y si pasa de un cierto valor preestablecido, retorna una decisión para indicar si esta en uso el medio o no. En caso de que se detecte

la transmisión de un paquete se mantiene encendido el dispositivo hasta el detectar una re-transmisión completa de un paquete para después mandar un 'ACK' al nodo que se encuentra enviando el paquete. Analizando el caso de un nodo transmisor, este manda el paquete de las capas superiores al 'buffer' de la capa 'MAC', subsecuente mente cuando la capa de 'RDC' del nodo entra en estado de radio encendió manda este paquete un número predeterminado de veces sin incluir el número de retransmisiones programado en la capa 'MAC' si después de alguna retransmisión recibe un 'ACK' de un vecino 'ContikiMac' entiende que el paquete fue transmitido de manera correcta y continua con el ciclo de trabajo preestablecido.

La implementación de 'ContikiMac' ha atraído trabajos de investigación con el fin de continuar mejorando como lo es, una propuesta de ciclo adaptativo en lugar de uno fijo (Youssef y cols., s.f.), pensado para dispositivos que cuenta con energía limitada, se encarga de disminuir el ciclo de trabajo a la par que disminuye la energía a disposición del dispositivo, dando resultados de un aumento en el tiempo de vida de la red hasta en un 136%.

3.4. COOJA

COOJA es un simulador de redes, el cual permite la emulación de plataformas de hardware comúnmente usado para redes IoT (Mehmood, s.f.). es una aplicación de CONTIKI, la cual se concentra en redes de sensores sin proponer ningún nodo base en particular.

Entre las características que podemos encontrar en COOJA se encuentran los distintos estándares que soporta para los 'nodos' los cuales son :

- Tr 1100
- TI CC2420
- CONTIKI-RPL
- IEEE 802.15.4
- uIPv6 y uIPv4

De la misma manera nos encontramos distintos modelos de propagación aplicables para las simulaciones , siendo el primero el modelo 'Disk Graph Medium' o 'UDGM', el cual toma el rango de transmisión ideal en forma de disco, y siempre que se encuentre dentro del rango definido son transmitidos de manera correcta, fuera del rango de transmisión los paquetes se pierden y .El segundo modelos es 'UDGM' con pérdida por distancia, el cual considera

interferencias de radio, utilizando dos variables denominadas 'Success ratio Rx' y 'Success ratio Tx'. El tercer modelo es 'Directed Graph Radio Médium' el cual trabaja con tiempos de propagación. El cuarto modelo es 'Multipath Ray-tracer Mediu' o 'MRM' y utiliza métodos de rastreo como los cálculos de Friis para calcular la potencia del receptor, de la misma manera es capaz de computar difracciones, reflexiones y refracciones.

3.5. SIMULACION DE LA CAPA RDC EN CONJUNTO CON EL PRTOCOLO RPL

La organización de los proyectos de CONTIKI se basa en dos archivos principales, el primero es llamado 'Makefile' en el cual se declaran las rutas de los archivos base, al igual que variables para la activación de aplicaciones secundarias como 'POWERTRACKER' para la captura de información de gasto energético. El segundo archivo es 'project-conf.h' donde se declaran y re-definen variables para los archivos de 'debug' y 'stack' de red de CONTIKI, un ejemplo de este archivo se encuentra en la figura 3.1.

Para conocer las configuraciones que se pueden re-definir en 'project-conf.h' se necesita realizar una investigación exhaustiva de los archivos 'core' de CONTIKI, debido a que no existe una documentación completa de CONTIKI. Para el caso particular de '6LowPAN' las variables de configuración están en los archivos:

- El archivo 'rpl-config.h', donde se definen las variables para 'RPL' como lo son los 'RPL DIO INTERVAL MIN' y 'RPL-DIO-INTERVAL-DOUBLING', donde se definen los tiempos máximos y mínimos de transmisión de los paquetes 'DIO'.
- El archivo 'csma.c', donde se definen las variables del protocolo de la capa 'MAC' o 'CSMA', algunos de los parámetros que se pueden configurar son 'CSMA-MAX-BE' que define el 'Maximum backoff exponent' y 'CSMA-MAX-MAX-FRAME-RETRIES', donde se define la cantidad de retransmisiones máximas por paquete.
- El archivo 'contikimac.c', donde se definen parámetros como 'RDC-CONF-HARDWARE-ACK', que define si el hardware es el encargado de mandar un 'ACK' de manera automática cuando se recibe un paquete. 'RDC-CONF-HARDWARE-CSMA', donde define si el sistema

```

#ifndef PROJECT_CONF_H_
#define PROJECT_CONF_H_

#ifndef WITH_NON_STORING
#define WITH_NON_STORING 0 /* Set this to run with non-storing mode */
#endif /* WITH_NON_STORING */

#undef NBR_TABLE_CONF_MAX_NEIGHBORS
#undef UIP_CONF_MAX_ROUTES

/* configure number of neighbors and routes */
#define NBR_TABLE_CONF_MAX_NEIGHBORS 10
#define UIP_CONF_MAX_ROUTES 10

#undef NETSTACK_CONF_MAC
#define NETSTACK_CONF_MAC csma_driver

#undef NETSTACK_CONF_RDC
#define NETSTACK_CONF_RDC contikimac_driver
#undef NULLRDC_CONF_802154_AUTOACK
#define NULLRDC_CONF_802154_AUTOACK 0

/* Define as minutes */
#define RPL_CONF_DEFAULT_LIFETIME_UNIT 60

/* 10 minutes lifetime of routes */
#define RPL_CONF_DEFAULT_LIFETIME 10

#define RPL_CONF_DEFAULT_ROUTE_INFINITE_LIFETIME 1

#if WITH_NON_STORING
#undef RPL_NS_CONF_LINK_NUM
#define RPL_NS_CONF_LINK_NUM 40 /* Number of links maintained at the root. Can be set to 0 at non-root */
#undef UIP_CONF_MAX_ROUTES
#define UIP_CONF_MAX_ROUTES 0 /* No need for routes */
#undef RPL_CONF_MOP
#define RPL_CONF_MOP RPL_MOP_NON_STORING /* Mode of operation */
#endif /* WITH_NON_STORING */

#endif

```

Figura 3.1: Imagen del contenido de ejemplo de un archivo 'project-conf.h'

3.5.1. ESCENARIOS DE SIMULACION PROPUESTOS

Utilizando COOJA se plantearon escenarios para la evaluación del impacto de una capa 'RDC' sobre el estándar 'RPL', el primer acercamiento se muestra en la figura 3.2 , donde se tienen 3 nodos identificados por un identificador único, para todos los escenarios simulados el nodo con el identificador '1' fue declarado como el sumidero de la red, de la misma manera el identificador con el número mayor fue asociado al transmisor, y los nodos restantes a los retransmisores o nodos que no generan información.

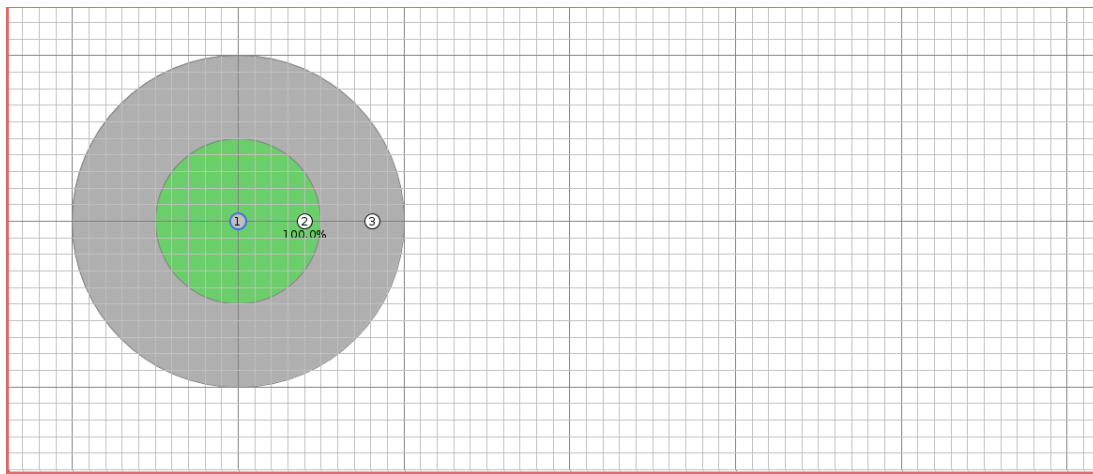


Figura 3.2: Topología de la simulación de red de 6LoWPAN, en un escenario multisalto.

La simulación tiene un modelo de pérdidas UDGM. Como se describió anteriormente este tipo de modelo tiene un patrón de pérdidas constantes dentro del área de cobertura (área verde) y un patrón de interferencia que se extiende hasta fuera de la misma (área 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.

Para un análisis más completo los nodos que componen la red 6LowPAN propuesta, no son móviles, además, los nodos se inicializa en el mismo instante de tiempo. El proceso de asesoramiento se ejecuta antes de que el nodo transmisor de la red comience a realizar la generación de paquetes, durante la simulación se realizó un barrido de 1 hasta 10 saltos con parámetros mencionados en la tabla 3.1.

Metricas de Simulacion	Valor Asignado
Tiempo Total de la simulacion	4 min
Distancia entre nodos	40m
Cantidad de Nodos Cliente	0-9
Cantidad de Nodos Transmisores	1
Cantidad de Nodos Sumidero	1
Numero de iteraciones por salto	100
Probabilidad del radio en Tx	100 %
Probabilidad del radio en Rx	100 %
Rango de Transmisión	50m
Rango de Interferencia	100m
Protocolo RPL	Valor Asignado
DIO MIN	3 (8 ms)
DIO Doublings	20 (2.3 hrs)

Tabla 3.1: Tabla de valores asignados a las variables para las simulaciones

3.6. CAPTURA DE METRICAS

Uno de los desafíos de trabajar con un simulador como COOJA es el método de recolección de información, los caminos propuestos por la comunidad de CONTIKI para resolver esta problemática son protocolos como 'CONTIKI-COLLECT' el cual es una implementación modificada del protocolo 'COLLECTION TREE' (Gnawali, Fonseca, Jamieson, Moss, y Levis, s.f.), este protocolo utiliza dos mecanismos principales, el primero 'validación por adaptación' el cual utiliza el tráfico para obtener la información de la topología, y el segundo 'beaconing adaptativo' el cual al igual que 'RPL' extiende el tiempo en que los paquetes de control son enviados a través de la red si se detecta estable.

La segunda opción que propone COOJA es es la herramienta 'headless radio logger' (*Contiki: The Open Source Operating System for the Internet of Things*, s.f.), con la cual se pueden capturar los paquetes generados por los nodos de la red y guardarlos en un .pcap, para posteriormente ser analizados con frameworks especializados como 'Scapy' (community, s.f.) el cual permite la manipulación de paquetes de manera sencilla, y software como 'Wireshark' (*Wireshark · Frequently Asked Questions*, s.f.), con muchas herramientas para análisis de red. Sin embargo la opción que se explora en este trabajo de tesis es la obtención de información por 'COOJA LOG', esta opción nos permite conocer información individual de cada nodo de manera más sencilla que las opciones pasadas, debido a que los nodos imprimen en consola eventos específicos de interés y después son anexados en el archivo 'COOJA.LOG', algunos de estos eventos que no son posibles de conocer en las otras opciones de recopilación de información son el tiempo de procesamiento entre una capa y otra, la cantidad de retrans-

misiones por paquete en cada nodo, la cantidad de paquetes descartados en cada nodo, así como funciones individuales de los mecanismos de cada capa, en la figura 3.3 podemos ver como es desplegada la información en la herramienta LOG de COOJA.

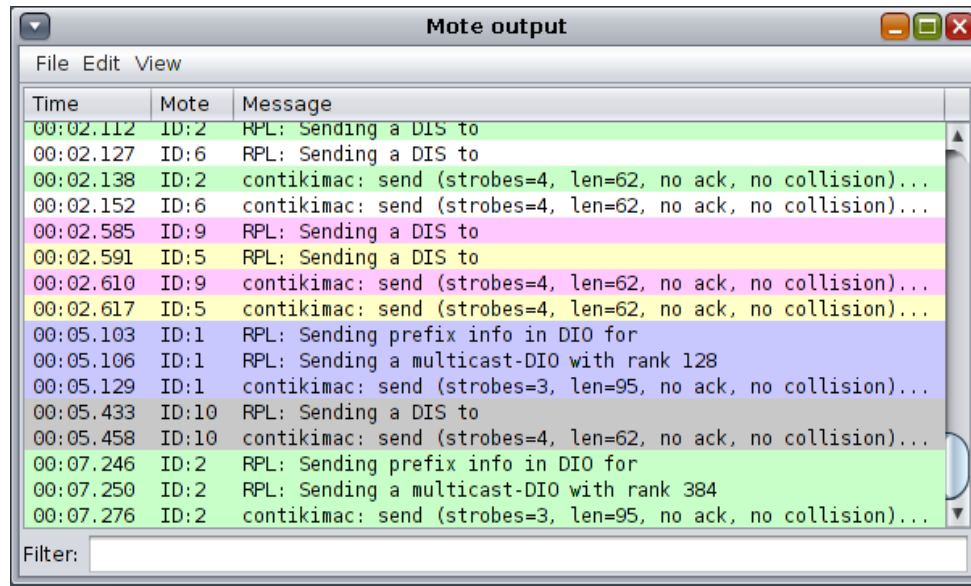


Figura 3.3: Herramienta de LOG de COOJA

Si bien con las simulaciones realizadas para este trabajo de tesis se busca el análisis del impacto de la capa 'RDC' sobre el protocolo 'RPL', se involucran la creación de distintas herramientas para alcanzar este objetivo, como la repetición de simulaciones para diferentes valores de los parámetros y configuraciones, estas repeticiones se necesitan en grandes cantidades. Dado estas razones el proceso para la ejecución de simulaciones se automatizo como se muestra en la figura siguiente.

Capítulo 4

ANALISIS DEL IMPACTO DE UNA CAPA RDC SOBRE EL PROTOCOLO RPL

4.1. Introducción

En esta sección se mostrarán los resultados obtenidos en la simulación de una red con el estándar 6LoWPAN y una capa RDC debajo de este con el fin de evaluar el desempeño y el impacto que causa sobre el enrutamiento de la red para escenarios con distintas cantidades de saltos necesarios para llevar la información del nodo transmisor hasta el sumidero de información, en este caso utilizando el dispositivo 'Z mote' y con el radio altamente utilizando en implementaciones para IoT 'cc2420'.

La capa RDC es una mecanismo altamente relevante para implementaciones de IoT con el estandar 6LoWPAN debido a los beneficios que teóricamente aporta a la red donde se implementa, sin embargo el impacto que causa sobre protocolos como RPL pocas veces es tomando debido a que las especificaciones definidas para RPL no están pensadas para llevar una capa de RDC por debajo por lo cual se han propuesto modificaciones por 'Contiki-OS' para una mejora en el desempeño de redes utilizando el protocolo RPL y una capa RDC. Es por esta razón que en este capítulo se mostrarán los resultados obtenidos de las distintas simulaciones donde fue puesto a prueba distintas frecuencias de trabajo de la capa RDC que porta Contiki-OS llamada 'ContikiMac' las cuales fueron 8 HZ ,32 HZ y 64 HZ, y de la misma manera con distintas frecuencias de transmisión de paquetes UDP 1 HZ , 5 HZ y 10 HZ. Los escenarios de simulación fueron seleccionados en un aumento de número de saltos de 1 hasta 10 de manera incremental, de los cuales se obtuvieron métricas de pérdida de paquetes, retransmisiones realizadas, paquetes descartados y tiempo de retraso de paquetes.

4.2. Definición de las métricas a evaluar

Las métricas definidas continuación ayudaran a valorar el impacto que tiene una capa RDC como lo es 'ContikiMac' de 'CONTIKI'.

- **Proporción de Entrega de Paquetes (Package Delivery Ratio).** Es el porcentaje de paquetes recuperados en el nodo sumidero enviados desde la capa UDP del transmisor.
- **Retransmisiones realizadas.** Para este trabajo de tesis, se consideran como retransmisiones a la cantidad de veces que la capa MAC inicio su mecanismo de retransmisión de paquetes para que sea recibido por el siguiente nodo en la cadena o el nodo sumidero.
- **Tiempo de retraso de paquetes (Delay).** Es el tiempo que un paquete enviado desde la capa UDP del transmisor de la red es recibido exitosamente por el sumidero, este considera el tiempo del paquete para todos los saltos que dio en la red.

El análisis de la de la red será evaluado con las métricas anteriormente especificadas, de igual manera es importante mencionar que los resultados obtenidos fueron validados por la ejecución de varias simulaciones sucesivas promediando sus resultados. Se realizó un barrido de frecuencias de 8 HZ, 32 HZ y 64 HZ para la configuración de la capa 'RDC' a su vez variando la cantidad de nodos que forman la red de la misma manera aumentando los saltos que tiene que dar la información para viajar del transmisor hasta el sumidero hasta en un máximo de 10 saltos.

4.3. Análisis del desempeño de la red solo con el protocolo RPL activo

El análisis de la de la red será evaluado con las métricas anteriormente especificadas, de igual manera es importante mencionar que los resultados obtenidos fueron validados por la ejecución de varias simulaciones sucesivas promediando sus resultados. Como primer acercamiento se simularon escenarios donde solo se tenia el protocolo RPL sin capa de 'RDC' debajo y con un medio de transmisión sin pérdidas, figura4.1, con un aumento de 1 salto por escenario hasta un máximo de 10 reflejado en el eje horizontal y en el eje vertical con el título 'PDR' se considera un valor máximo de 1, significando este último que al alcanzar el valor máximo todos los paquetes transmitidos fueron recuperados exitosamente en el sumidero,

en la figura anteriormente mencionada se puede observar que el 'PDR' es casi perfecto en el mayor de los casos presentando pérdidas no mayores al 0.1 %. De la misma manera si es notable una pérdida de 'PDR' al aumentar la cantidad de saltos. De esta figura podemos concluir que el solo al aumentar la cantidad de saltos del transmisor al sumidero presenta pérdidas de 'PDR', sin embargo estas pérdidas son menores al 0.1 % siempre y cuando se tenga un medio de transmisión sin pérdidas o perfecto.



Figura 4.1: PDR normalizado de la red variando el numero de saltos.

La figura 4.2, muestra los resultados obtenidos para retardo de los paquetes (en segundos) en el eje vertical conforme aumenta el número de saltos de estos para el protocolo RPL sin considerar la capa de RDC. Podemos observar cómo al aumentar la cantidad de saltos también aumenta el tiempo de retraso o 'DELAY' con una tendencia clara. Este retraso se debe, a que, para cada salto de información existe un tiempo desde que un nodo recibe el paquete y pasa a ser agendada su retransmisión para hacerlo llegar hasta el sumidero, por lo tanto se puede deducir que al aumentar el número de saltos de la misma manera aumenta la cantidad de veces que este proceso de retransmisiones con camino al sumidero suceden agregando para cada salto un tiempo de retraso claramente visible en la gráfica anteriormente mencionada.

Los resultados para la métrica de 'cantidad de retransmisiones' se pueden observar en la figura 4.3, donde se grafica la cantidad de retransmisiones realizadas (eje vertical) contra la cantidad de saltos en la simulación (eje horizontal) con un mínimo de 1 salto y un máximo de 10 saltos. Con solo el protocolo activo RPL podemos observar al igual que los resultados del 'DELAY' (figura 4.2) la cantidad de retransmisiones aumenta al incrementar la cantidad de saltos, cabe resaltar que la tendencia de incremento no está tan marcada como en las

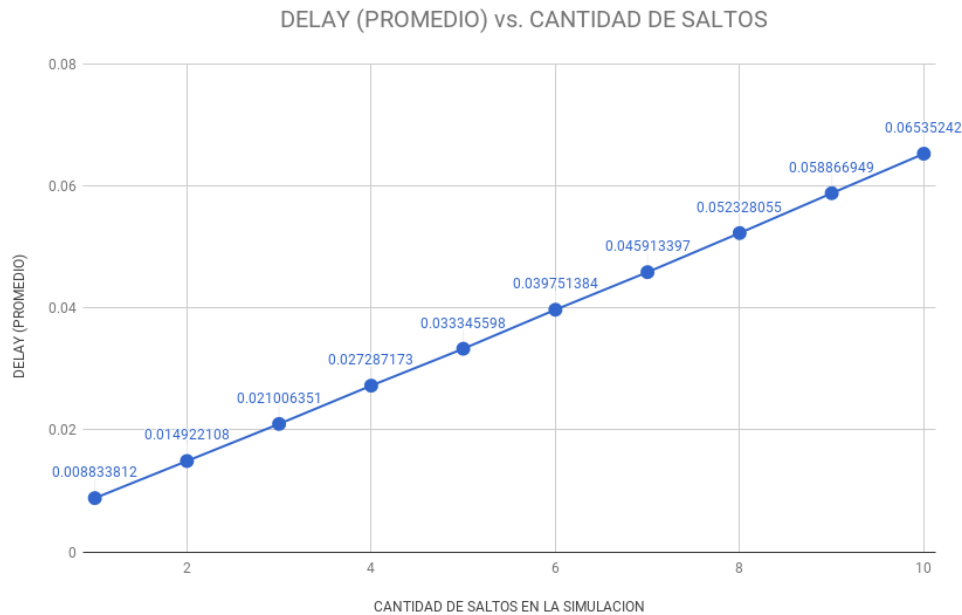


Figura 4.2: Tiempo en segundos del retraso promedio del arribo de paquetes al sumidero dado el numero de saltos realizados

gráficas anteriores (figura 4.1 y figura 4.2). Tomando en cuenta que el escenario de simulación contiene una tasa de éxito del 100 % al ingresar un paquete al canal es intuitivo concluir que las retransmisiones realizadas se generaron debido a que los nodos vecinos estaban en el proceso de transmitir los paquetes de mantenimiento y creación de la red del protocolo RPL.

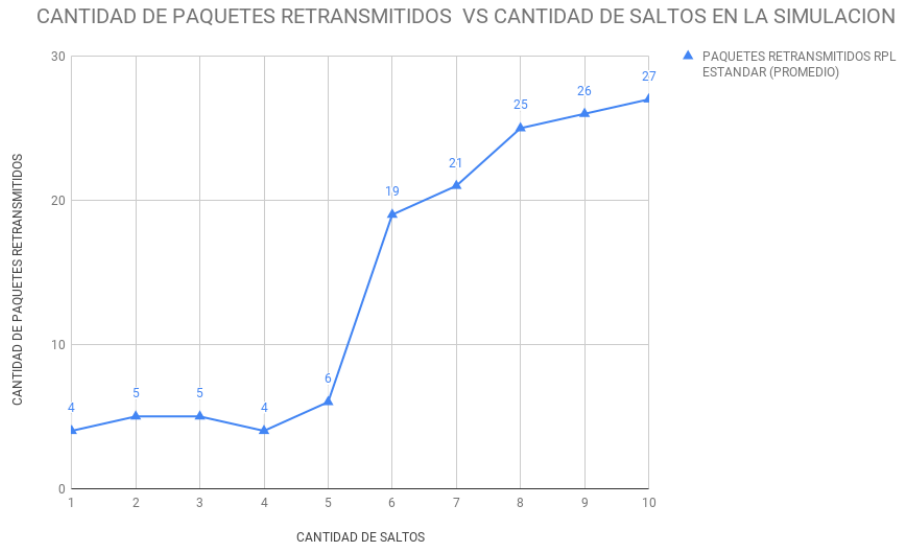


Figura 4.3: Cantidad de paquetes retransmitidos dado el numero de saltos realizados

4.4. RPL con modificaciones propuestas por CONTIKI-OS

Al agregar una capa para el control del ciclo de trabajo del radio CONTIKI-OS decidió modificar las especificaciones del estándar 'RPL' para que sean compatibles y no genere conflicto con su pila de software, esto debido a que hasta la fecha la capa 'RPL' no contempla un mecanismo debajo que controle el ciclo del trabajo del radio por lo cual configuraciones como el tiempo de la generación de paquetes de control para el mantenimiento temprano de la red salen se desalinean por completo para frecuencias sugeridas por CONTIKI-OS para su capa RDC como lo son 8 HZ o 16 HZ.

Las modificaciones efectuadas por CONTIKI-OS son las siguientes:

- **Intervalo de DIO.** De acuerdo a las especificaciones, el valor a asignar es de 2^3 ms lo cual nos da un valor de 8 milisegundos, lo cual es un valor muy bajo cuando se trabaja con una capa de RDC, donde usualmente se plantean intervalos de cientos de milisegundos, por lo cual se utiliza un valor de 2^{12} ms o 4.096 segundos.
- **Tiempo máximo de aumento para los DIO de RPL.** El estándar sugiere un valor máximo de 2^{23} lo cual nos da 8388.608 segundos, sin embargo esto es inadecuado cuando se empieza con un intervalo de 2^{12} ms debido a que el intervalo de generación de paquetes para el mantenimiento de la red incrementaría de manera desmesurada y mantener la red se tornaría una tarea compleja sobre todo para redes donde los nodos son móviles,

por lo cual se propone un aumento máximo de 2^8 ms o como final 2^{12+8} ms dando un resultado de 1048.576 segundos.

Con las modificaciones al protocolo 'RPL' por parte de CONTIKI, podemos observar de primera instancia una línea de tendencia en los resultados de 'PDR' (figura 4.4, la cual es muy parecida a los obtenidos con el protocolo 'RPL' según el estándar (figura 4.1). De la misma manera se gráfica la cantidad de retransmisiones realizadas (eje vertical) contra la cantidad de saltos en la simulación (eje horizontal) con un mínimo de 1 salto y un máximo de 10 saltos.

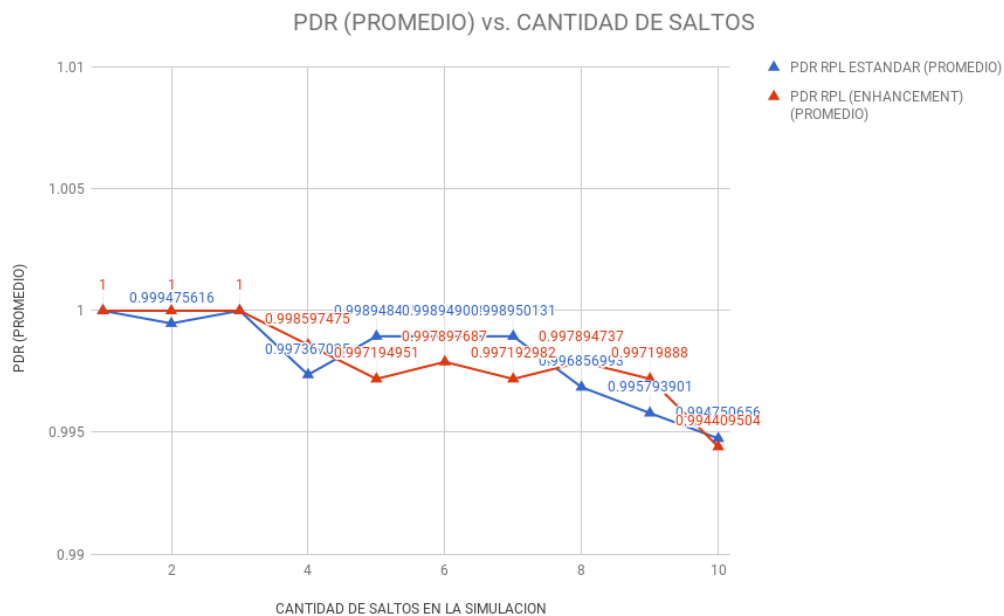


Figura 4.4: PDR normalizado de la red dado el numero de saltos.

Por el lado del tiempo de 'DELAY' o retraso, plasmado en la figura 4.5, los resultados obtenidos para retardo de los paquetes (en segundos) en el eje vertical conforme aumenta el número de saltos reflejado en el eje horizontal. Podemos observar cómo al aumentar la cantidad de saltos también aumenta el tiempo de retraso o 'DELAY' con una tendencia clara, y ambos resultados sobre puestos, esto nos indica que las modificaciones de CONTIKI no afectan al tiempo de retraso y se continúa con el mismo desempeño mostrado en la figura 4.2 cuando se tenía el protocolo 'RPL' según lo especificado en el estándar.

Los resultados de la cantidad retransmisiones contemplando las modificaciones propuestas por CONTIKI, se encuentran en la figura 4.6, podemos observar una diferencia notable en las tendencias conforme aumenta el número de saltos, para el protocolo 'RPL' sin modificaciones con una mínimo de 4 retransmisiones cuando se tiene 1 salto, hasta un máximo de

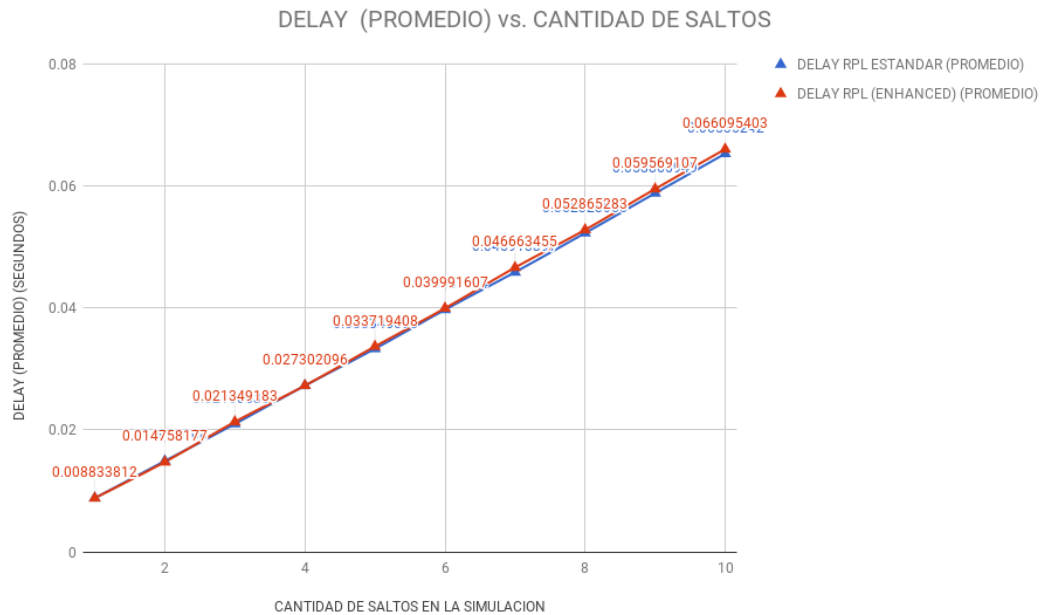


Figura 4.5: Retraso de los paquetes dado el número de saltos realizados

27 retransmisiones cuando el número de saltos se encuentra en 10, por el lado del protocolo 'RPL' con modificaciones de CONTIKI (línea roja) el número mínimo de retransmisiones es 0, y el máximo es 10 retransmisiones cuando se tienen 10 saltos, obteniendo una disminución en el número de retransmisiones del 62.96 %.

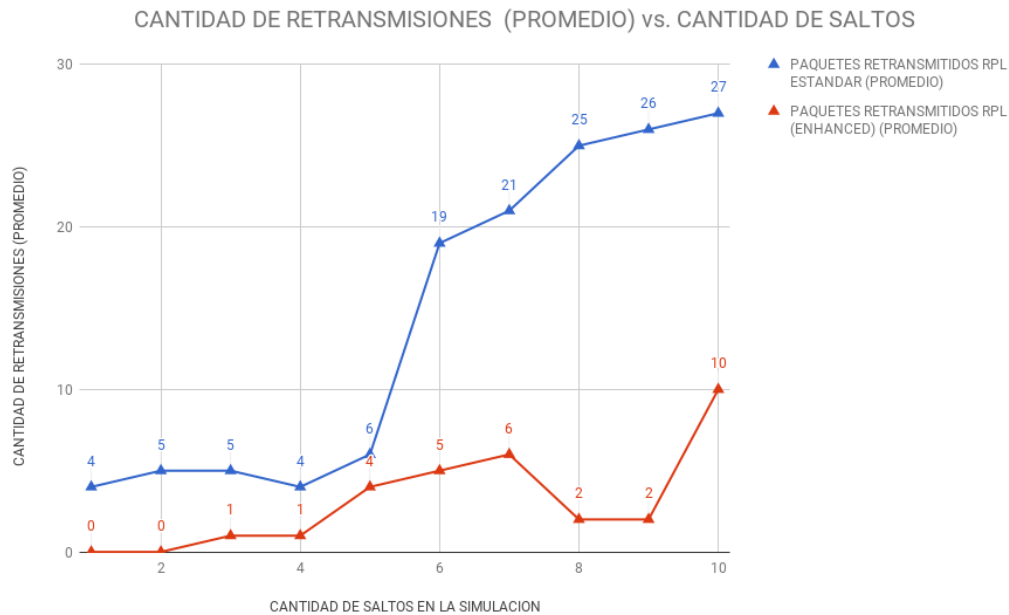


Figura 4.6: Cantidad de paquetes retransmitidos dado el numero de saltos realizados

4.5. Resultados con RPL y la capa RDC activa de CONTIKI-OS

El primer acercamiento para la evaluación de 'RPL' con una capa 'RDC' se llevó a cabo con la herramienta 'Radio Messages' de COOJA para poder observar los paquetes de la red, los resultados se muestran en la figura 4.7, donde se puede observar una serie de 'ACK' después de cada paquete recibido de manera exitosa, esto no se presenta cuando no esta presenta una capa de 'RDC', el impacto que esto tiene en el medio es un flujo en reversa el cual será más presenciado en los nodos intermedios, debido a que por un lado el nodo escucha un mensaje proveniente del transmisor este puede estar en conflicto o colisionar con un 'ACK' de otro nodo vecino.

No.	Time	From	To	Data
357	01:16.660	3	4	102: 15.4 D C1:0C:00:00:00:00:00:03 C1:0C:00:00:00:00:00:04 IPHC IPv6 ICM...
358	01:16.664	4	3	5: 15.4 A
359	01:16.671	2	1	66: 15.4 D C1:0C:00:00:00:00:00:02 C1:0C:00:00:00:00:00:01 IPHC IPv6 UDP...
360	01:16.674	1	2	5: 15.4 A
361	01:16.687	2	1	66: 15.4 D C1:0C:00:00:00:00:00:02 C1:0C:00:00:00:00:00:01 IPHC IPv6 UDP...
362	01:16.690	1	2	5: 15.4 A
363+1	01:16.703	3	-	102: 15.4 D C1:0C:00:00:00:00:00:03 C1:0C:00:00:00:00:00:04 IPHC IPv6 ICM...
365	01:16.711	3	2	102: 15.4 D C1:0C:00:00:00:00:00:03 C1:0C:00:00:00:00:00:04 IPHC IPv6 ICM...
366	01:16.728	3	-	102: 15.4 D C1:0C:00:00:00:00:00:03 C1:0C:00:00:00:00:00:04 IPHC IPv6 ICM...
367	01:16.732	3	4	102: 15.4 D C1:0C:00:00:00:00:00:03 C1:0C:00:00:00:00:00:04 IPHC IPv6 ICM...
368	01:16.735	4	3	5: 15.4 A
369	01:16.746	4	-	76: 15.4 D C1:0C:00:00:00:00:00:04 C1:0C:00:00:00:00:00:03 IPHC IPv6 ICM...
370	01:16.749	4	3	76: 15.4 D C1:0C:00:00:00:00:00:04 C1:0C:00:00:00:00:00:03 IPHC IPv6 ICM...
371	01:16.752	3	4	5: 15.4 A
372+1	01:16.797	3	-	102: 15.4 D C1:0C:00:00:00:00:00:03 C1:0C:00:00:00:00:00:04 IPHC IPv6 ICM...
374	01:16.805	4	3	5: 15.4 A

Figura 4.7: Herramienta 'Radio Messages' de COOJA en la cual se muestran los paquetes generados por los nodos de la red.

4.5.1. Frecuencia de envío 1 HZ

Con la implementación de la capa RDC de CONTIKI-OS "ContikiMac" podemos observar los siguientes resultados cuando se envían paquetes con una frecuencia de 1HZ para distintos escenarios, los cuales difieren solo en la cantidad de saltos que tiene que dar un paquete para llegar del transmisor hasta el sumidero, en la figura 4.4 se muestran los diferentes 'PDR' obtenidos variando la frecuencia asignada a la capa 'RDC', así como también cuando se tiene solo el protocolo 'RPL' con las modificaciones de CONTIKI.

Analizando los resultados se observa que se obtiene un PDR casi perfecto para todos los escenarios, cuando se tiene una frecuencia de envío de paquetes de 1 HZ, esto nos ayuda a entender que la capa 'RDC' no presenta un cambio significativo, sin embargo 1HZ son frecuencias muy bajas para algunos escenarios sobre todo para aplicaciones médicas donde se necesita recolectar información de manera más frecuente como un 'ECG'.

Para la métrica de 'DELAY' o retraso se obtiene que al aumentar la capa de 'RDC' disminuye el retraso de los paquetes por un margen significativo cuando se pasan de una frecuencia de 8HZ a una de 32 HZ, sin embargo para una de 32 HZ a una de 64HZ el cambio es mínimo, por su misma parte para la versión modificada de 'RPL' se asemeja mucho a la versión del 'RPL' como está estipulado en el estándar.

En el caso de los resultados para la métrica de retransmisiones mostrados en la figura 4.10 podemos observar claramente como las frecuencias más altas en la capa de 'RDC' tienen un menor número de retransmisiones para cada caso.

Este comportamiento para los paquetes descartados se presenta debido a la capa de RDC y los tiempos que tiene el radio del dispositivo en activo, dada la naturaleza de la capa de RDC

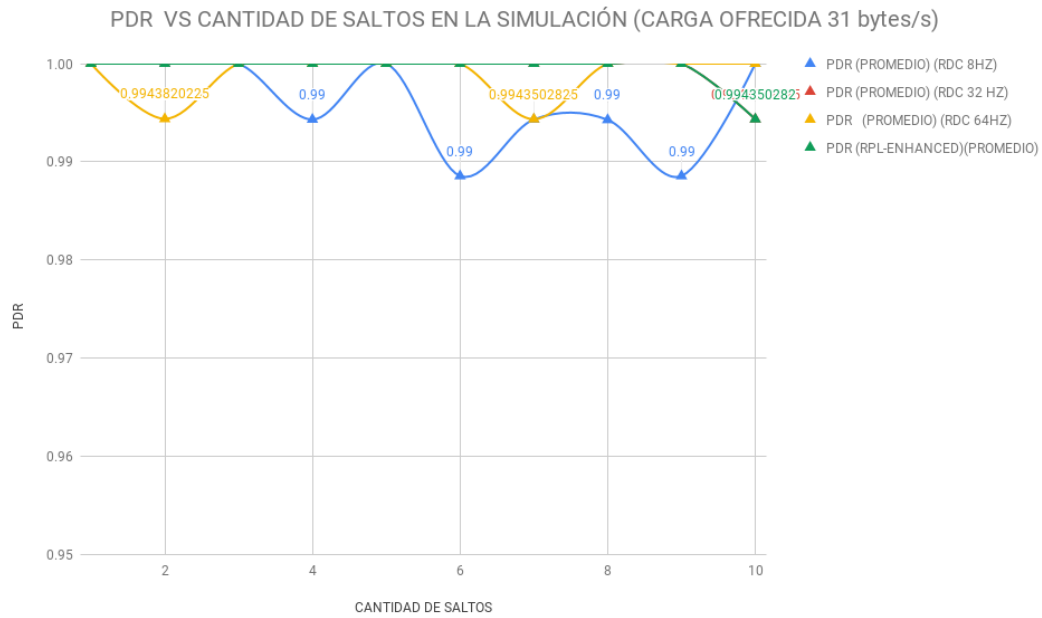


Figura 4.8: PDR normalizado de la red dado el numero de saltos.

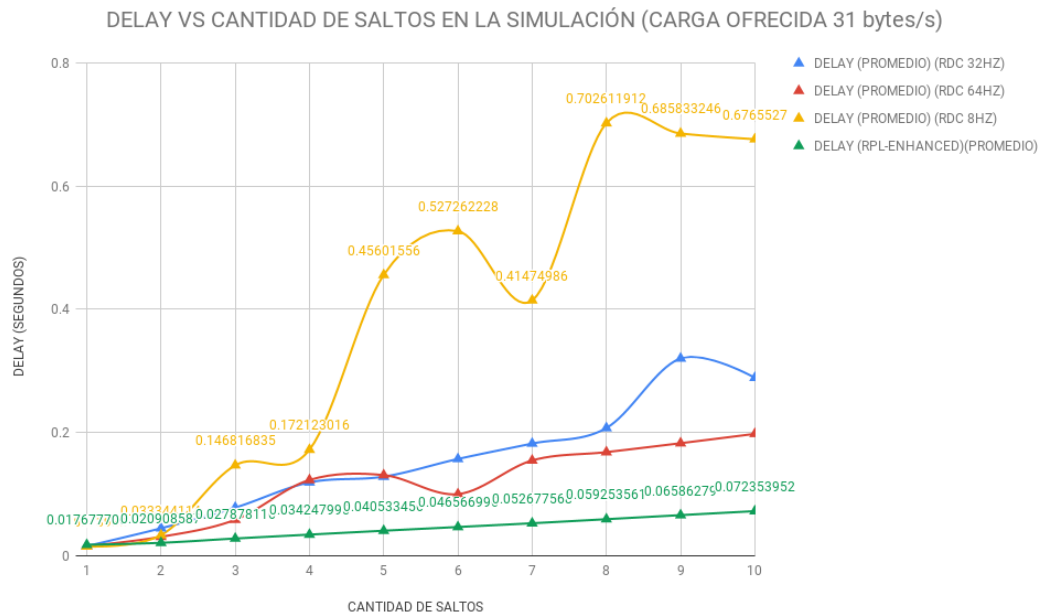


Figura 4.9: Retraso de los paquetes dado el numero de saltos realizados.

el radio del dispositivo se mantiene en estado de bajo consumo o de receptor la mayor parte del tiempo, para el caso de una frecuencia de 8 HZ en la capa de 'RDC' el radio del dispositivo cada se enciende cada 125 ms para detectar los paquetes "strokes" que genera la capa RDC cada que se quiere transmitir un paquete y mantenerse activo si es el caso. Si bien el tener en activo el radio del dispositivo una cantidad de menor de tiempo consume menos energía pero un aumento en el número de retransmisiones empieza a reducir este ahorro, sin embargo para el caso de una tasa de transferencia de 1 HZ, con el protocolo 'RPL' y una frecuencia en la capa de 'RDC' de 8 HZ siendo esta la más baja manejada durante las simulaciones de este trabajo, no deterioran en gran medida el posible ahorro energético de la introducción de la capa 'RDC' en la red.

CANTIDAD DE PAQUETES RETRANSMITIDOS (NORMALIZADOS) VS CANTIDAD DE SALTOS EN LA SIMULACIÓN (CARGA OFRECIDA 31 bytes/s)

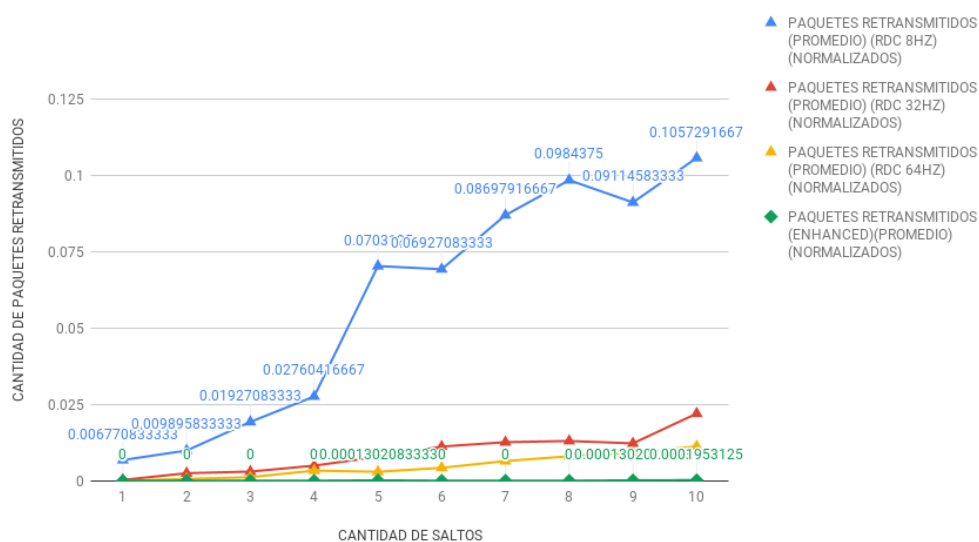


Figura 4.10: Cantidad de paquetes retransmitidos dado el numero de saltos realizados.

4.5.2. Frecuencia de envío 5 HZ

En esta segunda serie de simulaciones se cambió la frecuencia de generación de paquetes a 5 HZ, por lo que podemos obtener una carga ofrecida de 155 bytes/s, los resultados de la métrica de 'PDR' podemos observarlos en la figura 4.11, donde tenemos el 'PDR' normalizado en el eje vertical y la cantidad de saltos de 1 a 10 en el eje horizontal, un aspecto a destacar es podemos el cambio completo en el 'PDR' obtenido cuando se cuenta con una frecuencia en la capa de 'RDC' de 8 HZ, haciendo un contraste con el caso anterior (figura 4.8, sin embargo conservando un patrón similar de comportamiento para las frecuencias de 5 HZ y 10 HZ.

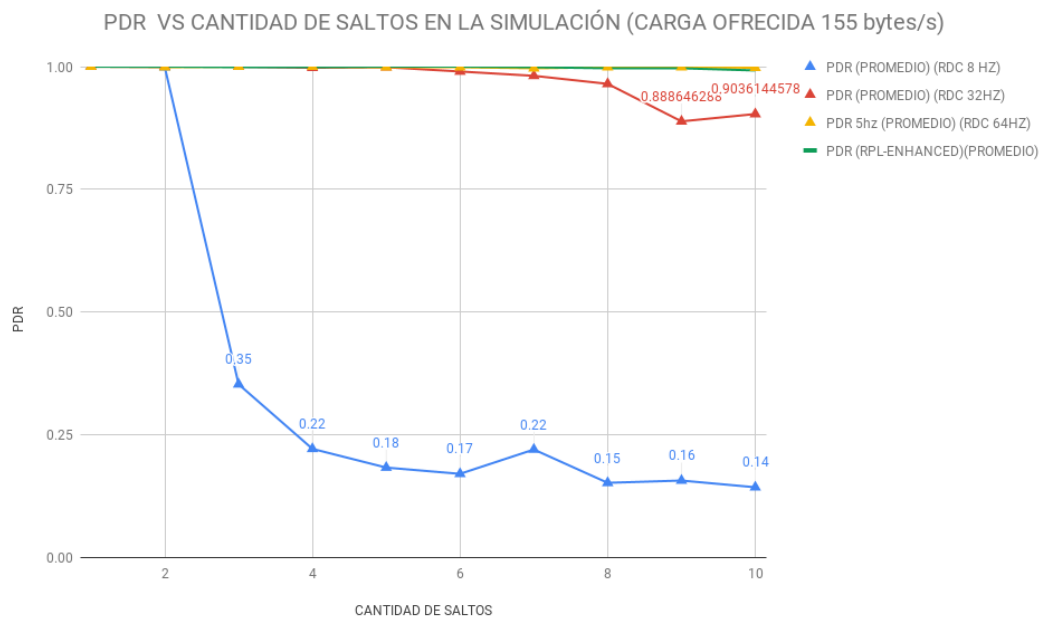


Figura 4.11: PDR normalizado de la red dado el numero de saltos.

Esta pérdida en el 'PDR' para la frecuencia de 8 HZ en la capa 'RDC', se atribuye a la cantidad de veces por segundo que el nodo tiene para realizar una transmisión. La capa de 'RDC' funciona como un cuello de botella para la transmisión de paquetes, al tener menos oportunidades para 'despertar' disminuyen las veces que el nodo puede transmitir y encontrar a sus vecinos despiertos para realizar una transmisión. De igual manera ese cuello de botella afecta a todos los mecanismos y protocolos superiores a la capa 'RDC' como la capa 'MAC' debido a que no solo tiene que agendar una retransmisión de los paquetes generados de capas de UDP o superiores, sino también los paquetes de mantenimiento de la red del protocolo RPL haciendo que el buffer del dispositivo se sature rápidamente y este empiece a descartarlos. por lo cual podemos deducir que se tiene que ajustar el ciclo de trabajo de la capa de 'RDC' a una que supere la frecuencia con la cual se generan los paquetes.

En la figura 4.12 podemos observar los resultados para la métrica de 'DELAY' cuando la carga ofrecida es de 155 bytes, en el eje vertical podemos observar las divisiones en segundos, y en el eje horizontal la cantidad de saltos realizados en las simulaciones de 1 hasta un máximo de 10, los diferentes ciclos de trabajo analizados están identificados con un color en específico, centrándonos en los resultados obtenidos para la frecuencia de 8 HZ en la capa de 'RDC' podemos observar como es considerablemente más alto que los otras frecuencias sobre todo cuando se tienen 10 saltos, llegando a un punto máximo de 2.9 segundos, y vemos como va decreciendo conforme disminuye la cantidad de saltos, solo alcanzando un desempeño comparable al obtenido en las otras frecuencias cuando se tienen 2 saltos o menos. Por el lado de las frecuencias superiores a 8 HZ vemos como se tienen un 'DELAY' considerablemente bajo, pero que empieza a aumentar notoriamente cuando se tienen 8 saltos o más, sobretodo para la frecuencia de 32 HZ.

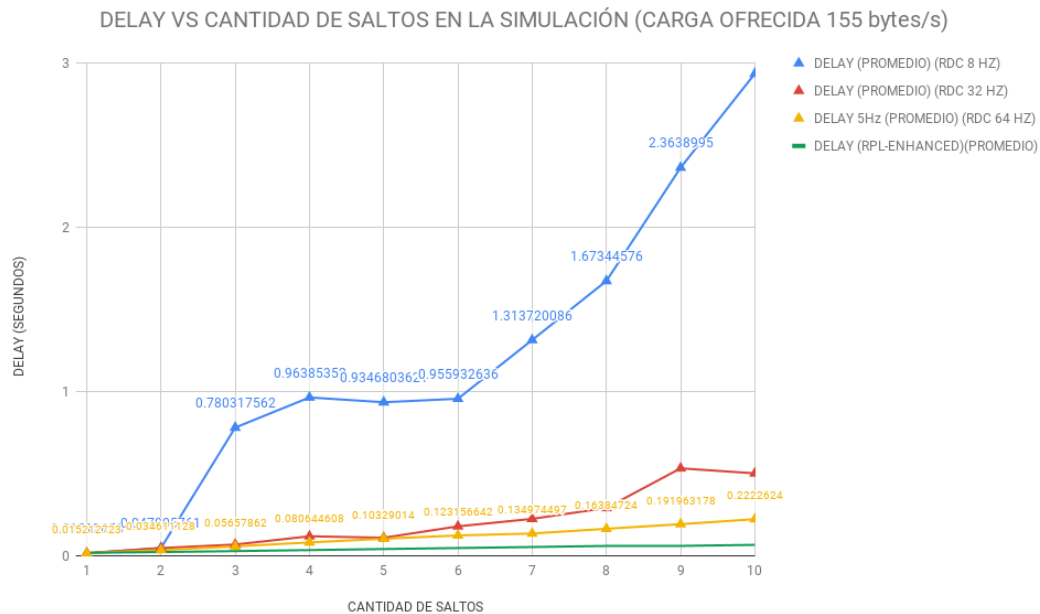


Figura 4.12: Retraso de los paquetes dado el numero de saltos realizados.

Los resultados de la cantidad de retransmisiones obtenidas cuando se tiene una carga ofrecida de 155 bytes, se pueden observar en la figura 4.13, para una comparación equitativa estos están normalizados dado la cantidad máxima de paquetes transmitidos en las simulaciones, cuando se observan los resultados para una frecuencia en 'RDC' de 8 HZ, podemos notar cómo después de tener 3 saltos en las simulaciones se tiene un total 0.58, significando que del 100 % de oportunidades que se tienen para realizar una transmisión de un paquete en una frecuencia de 8HZ, se utilizó aproximadamente el 60 %, y vemos como llega al 100 % cuando se tienen 9 saltos, con estos resultados podemos corroborar lo mencionado anteriormente, referente a la pérdida considerable de 'PDR' mostrado en la figura 4.11 para una frecuencia de 8 HZ, para estos casos una pérdida de 'PDR' significa que los nodos se están quedando sin oportunidades para retransmitir los paquetes perdidos y de la misma manera agotando el número de oportunidades que pueden transmitir en el canal, lo subsecuente es estos paquetes son descartados del buffer del nodo.

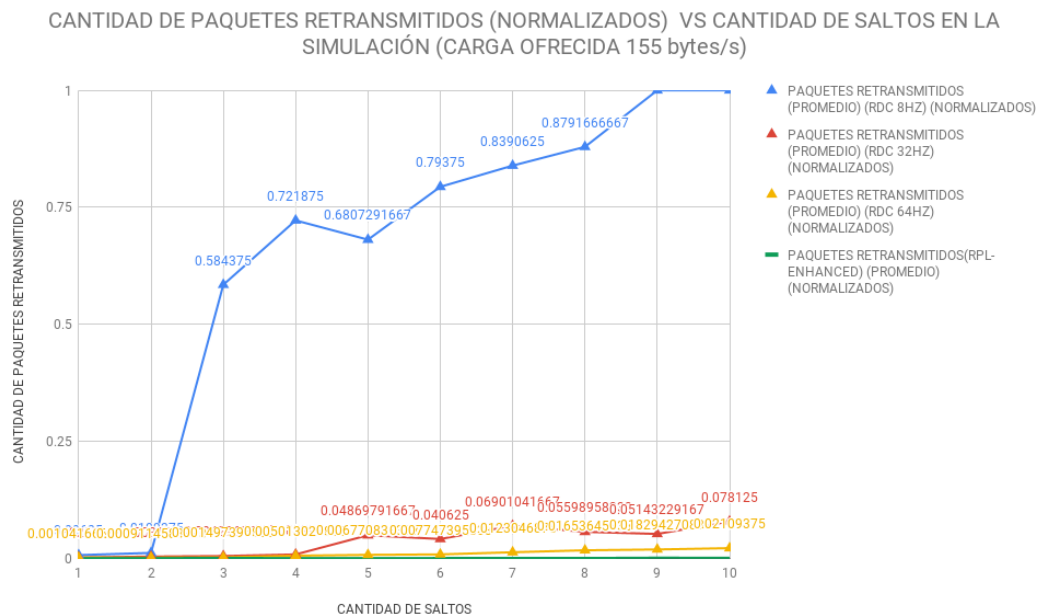


Figura 4.13: Cantidad de paquetes retransmitidos dado el numero de saltos realizados.

4.5.3. Frecuencia de envío 10 HZ

En esta serie de simulaciones se incrementó la frecuencia de envío de paquetes a 10 HZ, de esta manera ofreciendo una carga de 310 bytes/s, comenzando con la métrica de 'PDR', podemos observar como los resultados obtenidos en la figura 4.14 son muy diferentes a los obtenidos a las figuras 4.8 y 4.11, comenzando con la frecuencia de 8 HZ y de 32 HZ, se tienen resultados similares, para el caso de la frecuencia de 64 hz (color amarillo en la figura 4.14) observamos como de 1 a 3 saltos mantiene un 'PDR' del 100 %, sin embargo después comienza a descender hasta llegar a su punto más bajo de 47% cuando se tienen 10 saltos. Para los resultados de las frecuencias de 8 HZ y 32 HZ observamos cómo mantienen un 'PDR' similar conforme aumenta el número de saltos, esto puede deberse a que las frecuencias de 8 HZ y 32 HZ saturan el número máximo de oportunidades que tienen para transmitir.

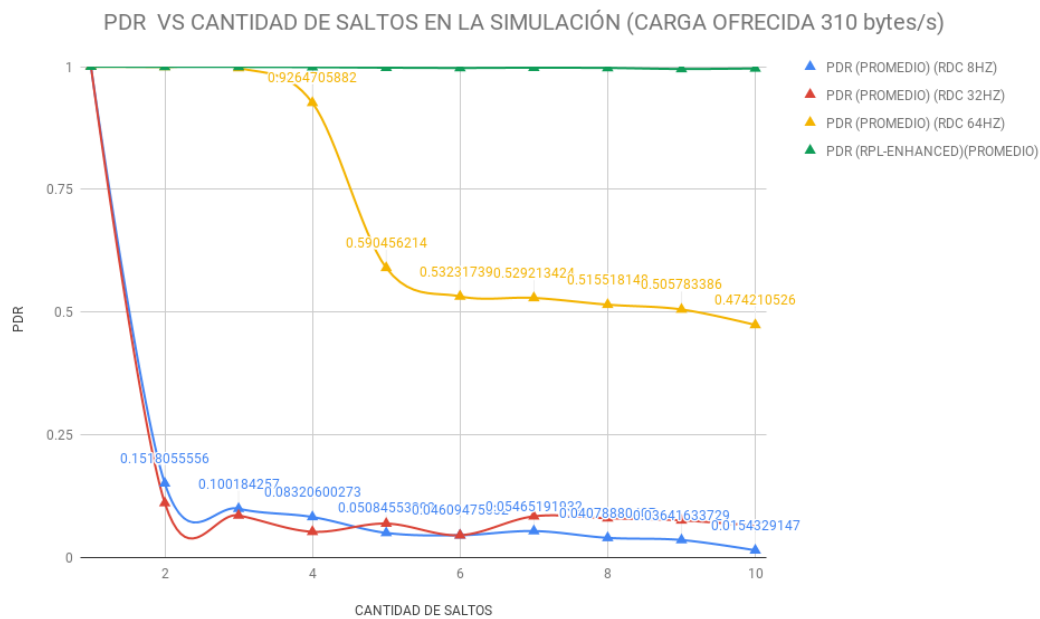


Figura 4.14: PDR normalizado de la red dado el numero de saltos.

Los resultados de la métrica de 'DELAY' se observan en la figura 4.15, donde a comparación de los resultados mostrados anteriormente en las figuras 4.12 y 4.9, observamos un comportamiento nuevo para los resultados de las frecuencias de 8 HZ y 32 HZ, donde de 1 a 4 saltos se obtiene una creciente en el tiempo de retardo pero después empieza a decrementar, este comportamiento puede deberse al cuello de botella que genera la capa de 'RDC' , por lo que los resultados después de los 5 saltos son solo los paquetes que son recuperados con las mejores características del canal, por lo tanto podemos concluir, que el comportamiento en el aumento en la cantidad de retardo cuando se tienen de 1 a 4 saltos, se debe a que es el

crecimiento que se tiene antes de que el cuello de botella generado por la capa de 'RDC' se sature.

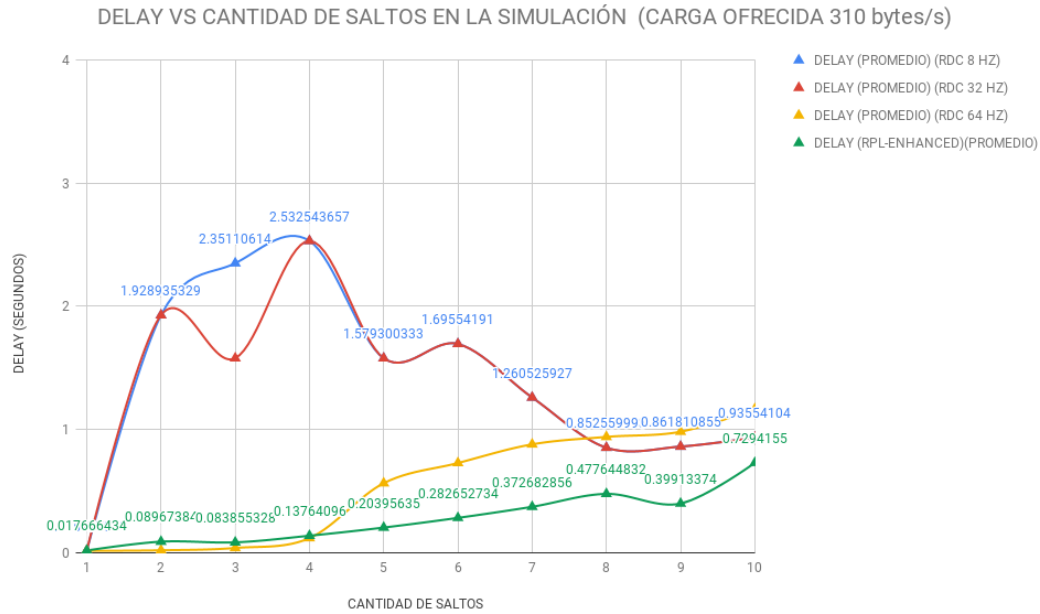


Figura 4.15: Retraso de los paquetes dado el numero de saltos realizados.

En la figura 4.16 podemos observar los resultados para la métrica de 'cantidad de retransmisiones', donde observamos cómo se disparan los resultados para las frecuencias de 8 HZ y 32 HZ, a comparación de la frecuencia de 64 HZ, corroborando la hipótesis del punto anterior. Los resultados de la frecuencia de 64 HZ (color amarillo) nos indica que nunca alcanzó a saturarse con el cuello de botella que genera la capa de 'RDC', como en el caso de las otras dos frecuencias, por lo cual su 'PDR' no colapsó recalando cómo fue el caso de las frecuencias de 8 HZ y 32 HZ.

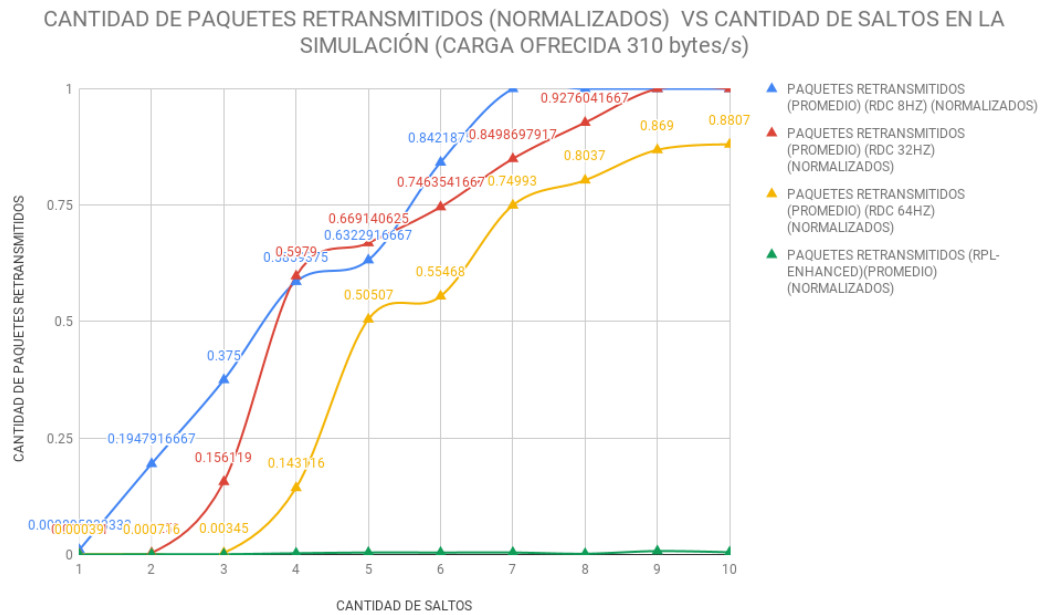


Figura 4.16: Cantidad de paquetes retransmitidos dado el numero de saltos realizados.

4.6. Conclusiones del análisis del efecto del ciclo de trabajo sobre el protocolo RPL

Los resultados de las simulaciones para los escenarios de una red 6LoWPAN utilizando Contiki-OS y distintas frecuencias para su capa 'RDC', muestran que el mejor de los desempeños se obtienen cuando se utiliza una frecuencia de 64HZ. Debido a que con esta frecuencia los nodos tienen un mayor numero de oportunidades de transmitir un paquete (teóricamente 64 oportunidades) comparado con las frecuencias de 8HZ donde se tienen teóricamente 8 oportunidades de transmisión, y en 32HZ donde se tienen teóricamente 32 oportunidades de transmisión.

Para el análisis del efecto de la cantidad de saltos en la red, se concluye que el a los 4 saltos de información para todos los casos que la frecuencia de la capa 'RDC' es menor que la frecuencia de envío de paquetes, el 'PDR' disminuye considerablemente, de la misma manera, el tiempo de retraso alcanza su valor máximo y el numero de paquetes descartados empieza a crecer rápidamente. Sin embargo cuando se tiene una frecuencia en la capa 'RDC' mayor que con la cual se generan paquetes por parte del transmisor, el porcentaje de 'PDR' perdido comienza a decrecer, este comportamiento es reflejado mas notoriamente en la figura 4.14, donde para una frecuencia de 8 HZ el 'PDR' a 10 saltos es menor al 0.1, para una frecuencia de 64 HZ a 10 saltos el 'PDR' se encuentra alrededor de 0.47.

De la misma manera es importante destacar que un aumento en la frecuencia de la capa 'RDC', puede significar un aumento en el costo energetico, debido a que el radio del dispositivo se enciende un numero mayor de veces y cuando se habla de nodos de bajo consumo y bajas capacidades también se incluyen celdas de energía o baterías, por lo tanto al aumentar la frecuencia de la capa 'RDC' también afectamos el tiempo que un nodo puede mantenerse activo en la red, se presentan costos en el 'PDR', donde si se requiriere de una aplicación que necesite ofrecer una carga mayor a 310 bytes/s o a frecuencias muy elevadas mayores a 5 HZ le constara mantener un 'PDR' aceptable cuando tenga una frecuencia de 'RDC' muy baja o menor a la cual esta generando paquetes de información.

Por ultimo, se puede concluir que el protocolo 'RPL' no esta diseñado pensando en soportar una capa de 'RDC' debajo, debido a que las configuraciones para los paquetes de mantenimiento de la red exceden el tiempo que otorga una frecuencia recomendada en la capa 'RDC', hablando de 8 HZ, 32 HZ y 64 HZ, cabe mencionar que se ha demostrado que las modificaciones propuestas por CONTIKI para el protocolo RPL son acertadas y van en un camino correcto hacia lograr una mejora al protocolo 'RPL' para que sea compatible con una capa de 'RDC', sin embargo no se descarta una que se proponga un algoritmo de enrutamiento mas inteligente para este tipo de implementaciones.

Capítulo 5

CONCLUSIONES Y TRABAJO FUTURO

5.1. Introducción

En el presente trabajo de investigación se evaluó el impacto que una capa de 'RDC' tiene sobre el estándar 'RPL', pensando en una implementación de una red 6LoWPAN, haciendo uso de simulaciones suficientes para alcanzar un porcentaje estadístico aceptable. Durante el proceso de este trabajo surgieron problemáticas, las cuales se les trato de dar la mejor solución posible, y de esta manera adquirir nuevas herramientas y conocimientos. En este capítulo también se presentan las propuestas de un posible trabajo futuro, el cual permitirá llevar a cabo una mejora de este trabajo de investigación.

La principal aportación de este trabajo de tesis, radica en la evaluación del impacto que una capa de 'RDC' como la de CONTIKI tiene sobre un protocolo de enrutamiento como lo es 'RPL', de tal manera que, realizando este análisis pueda identificarse los pros y contras que presenta 'RPL' con 'RDC'. Además de las aportaciones mencionadas, una muy importante es la lógica de las herramientas de análisis, el cual es de gran importancia para posteriores trabajos que requieran de analizar simulaciones del simulador de redes COOJA.

5.2. Conclusiones

Como se ha observado a lo largo de este trabajo de investigación, no existe en la literatura un análisis del impacto que tiene la incorporación de una capa de 'RDC' sobre el protocolo 'RPL' por lo cual fue necesario llevar a cabo una recopilación de trabajos anteriores alrededor de 6LoWPAN y 'RPL' para plantear un posible camino a seguir en esta investigación. Respecto a los trabajos encontrados se pueden mencionar los siguientes puntos:

- Se realizó una búsqueda en la literatura de las últimas investigaciones alrededor de 6LoWPAN, el protocolo 'RPL' y CONTIKI con la finalidad de determinar un camino a seguir, con el fin de lograr un análisis del impacto de una capa 'RDC' sobre 'RPL'. La investigación resultante puede ser de utilidad para otras investigaciones o desarrollos que incorporen redes IEEE 802.15.4 con enrutamiento en base a 'RPL' y una capa de 'RDC'.
- Ninguno de los trabajos encontrados realiza un análisis con tasas de envíos de paquetes relativamente altos, sin embargo todos aportan herramientas y métricas que fueron consideradas para la realización de esta investigación.

Con respecto al simulador de redes COOJA y el sistema operativo CONTIKI se establecieron las siguientes conclusiones:

- La investigación realizada para comprender a fondo el sistema operativo CONTIKI y sus herramientas como COOJA, determinó que existe poca o nula documentación para el manejo y comprensión de las mismas, la mayor parte de la información se encuentra comentada dentro de los archivos del código fuente de CONTIKI, por lo cual aunque se anexan ejemplos es muy difícil comprender el como utilizar o implementar partes específicas con el fin de realizar un escenario no propuesto en los ejemplos.
- CONTIKI es un sistema muy completo con una trayectoria que lo ha convertido en un sistema operativo robusto, y cuenta con investigaciones utilizándolo como herramienta debido a las características que ofrece
- Actualmente no se cuentan con herramientas especializadas para trabajar con COOJA en la obtención de métricas para en análisis de las características de redes, el método propuesto por CONTIKI el protocolo 'Contitki-Collect' incorpora una carga extra al nodo, por lo cual un desarrollo de herramientas preferentemente bajo un estándar es necesario.

Con respecto al comportamiento y el impacto de la capa de 'RDC' sobre 'RPL':

- Tras los resultados obtenidos, se puede concluir que el protocolo 'RPL' no soporta en su totalidad una capa de 'RDC', a tal punto que se han tenido que considerar modificaciones al estándar para lograr un mejor desempeño, diseñar un protocolo de enrutamiento que su diseño este pensado para soportar una capa de 'RDC' es una de las soluciones a este problema.

- El retardo en una implementación del protocolo de 'RPL' con una capa de 'RDC' está estrictamente relacionado con la frecuencia a la que trabaja la capa de 'RDC', al asignarse una frecuencia más baja el retardo es más alto, que trabajando con frecuencias más altas, y de la misma manera con frecuencias de bajas en la capa de 'RDC' se detectó un cuello de botella importante para la cantidad de paquetes que puede ser transmitido limitando el la carga que se ofrece a la red.

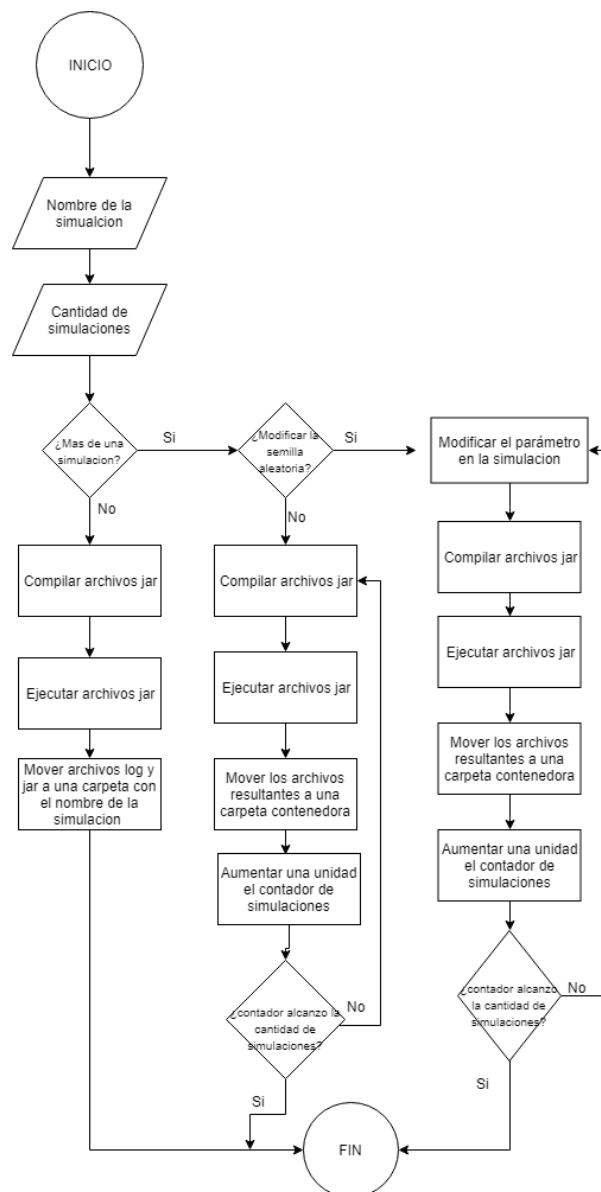
5.3. Trabajo Futuro

Este trabajo permite conocer el impacto que genera una capa de 'RDC' en el desempeño del protocolo 'RPL'. El análisis arroja una conclusión con una posible respuesta a la problemática encontrada, la cual es diseñar un posible nuevo algoritmo de enrutamiento que soporte una capa de 'RDC'. Con el diseño de un nuevo algoritmo se pueden obtener mejores resultados para las métricas de acuerdo con la arquitectura de la red y esto permitiría contar con una pila de protocolos que obtenga una ahorro energético y no afecte mecanismos superiores. Al contar con estos elementos, se puede brindar mejores servicios en las aplicaciones IoT.

En esta sección se enlistan los puntos en los que este trabajo de investigación se puede extender y las áreas de oportunidades relacionadas:

- Diseño de un protocolo de enrutamiento que contemple el tener una capa de 'RDC', o una modificación documentada al protocolo 'RPL' para hacerlo compatible con una capa de 'RDC'.
- Realizar la implementación física de las simulaciones realizadas en este trabajo, en un 'mote' que soporte el sistema CONTIKI, y corroborar los resultados obtenidos en este trabajo de investigación.
- Diseño y comprobación de herramientas de para la captura de métricas de las simulaciones elaboradas con COOJA, tomando en cuenta los archivos '.pcap' y '.log'.

Anexo I: Diagrama de flujo para la realización de simulaciones continuas



Bibliografía

- 6lowpan overview - ARM mbed.* (s.f.). Descargado 2018-07-11, de https://docs.mbed.com/docs/arm-ipv66lowpan-stack/en/latest/quick_start_intro/
- 6lowpan status pages.* (s.f.). Descargado 2018-07-16, de <https://tools.ietf.org/wg/6lowpan/draft-ietf-6lowpan-format/>
- 6lowpan: The wireless embedded internet.* (s.f.). Descargado 2018-07-11, de <https://www.wiley.com/en-mx/6LoWPAN%3A+The+Wireless+Embedded+Internet+-p-9780470747995>
- 6lowpan: The wireless embedded internet | communication technology - networks | communication technology | general & introductory electrical & electronics engineering | subjects | wiley.* (s.f.). Descargado 2018-07-11, de <https://www.wiley.com/en-sg/6LoWPAN%20The%20Wireless%20Embedded%20Internet-p-9780470747995>
- About - IEEE internet of things.* (s.f.). Descargado 2018-07-11, de <https://iot.ieee.org/about.html>
- Al-Kashoash, H. A. A., Hassen, E., Kharrufa, H., y Kemp, A. H. (s.f.). Analytical modelling of congestion for 6lowpan networks. Descargado 2018-07-20, de <http://www.sciencedirect.com/science/article/pii/S2405959516302107> doi: 10.1016/j.ict.2017.11.001
- Alpern, N. J., y Shimonski, R. J. (s.f.). CHAPTER 5 - the OSI model and networking protocols. En N. J. Alpern y R. J. Shimonski (Eds.), *Eleventh hour network+* (pp. 73–88). Syngress. Descargado 2018-07-19, de <http://www.sciencedirect.com/science/article/pii/B9781597494281000060> doi: 10.1016/B978-1-59749-428-1.00006-0
- armmbed. (s.f.). *mbed connect asia 2016 intro to mbed OS* [Technology]. Technology. Descargado 2018-07-17, de <https://www.slideshare.net/armmbed/mbed-connect-asia-2016-intro-to-mbed-os>
- Can we expect the internet of things in healthcare?* (s.f.). Descargado 2018-08-03, de <https://internetofthingsagenda.techtarget.com/feature/Can-we-expect-the-Internet-of-Things-in-healthcare>
- Cascio, W. F., y Montealegre, R. (s.f.). How technology is changing work and organizations. , 3(1), 349–375. Descargado 2018-07-16, de <http://www.annualreviews.org/doi/10>

- .1146/annurev-orgpsych-041015-062352 doi: 10.1146/annurev-orgpsych-041015-062352
- Chakrabarti, S., Shelby, Z., y Nordmark, E. (s.f.). *Neighbor discovery optimization for IPv6 over low-power wireless personal area networks (6lowpans)*. Descargado 2018-07-11, de <https://tools.ietf.org/html/rfc6775>
- Chang, L.-H., Lee, T.-H., Chen, S.-J., y Liao, C.-Y. (s.f.). Energy-efficient oriented routing algorithm in wireless sensor networks. En (pp. 3813–3818). IEEE. Descargado 2018-07-16, de <http://ieeexplore.ieee.org/document/6722404/> doi: 10.1109/SMC.2013.651
- community, P. B. a. t. S. (s.f.). *Scapy*. Descargado 2018-08-07, de <https://secdev.github.io/>
- Contiki: The open source operating system for the internet of things*. (s.f.). Descargado 2018-07-29, de <http://www.contiki-os.org/#why>
- cooja-radiologger-headless: Headless radio logger for COOJA*. (s.f.). CETIC. Descargado 2018-08-07, de <https://github.com/cetic/cooja-radiologger-headless> (original-date: 2013-07-02T15:00:37Z)
- Deering {\\textless}deering@cisco.com{\\textgreater}, S. E. (s.f.). *Internet protocol, version 6 (IPv6) specification*. Descargado 2018-07-17, de <https://tools.ietf.org/html/rfc2460>
- Defining the internet of things: A work in progress*. (s.f.). Descargado 2018-07-10, de <https://www.ecnmag.com/blog/2015/11/defining-internet-things-work-progress>
- Dimitrov, D. V. (s.f.). Medical internet of things and big data in healthcare. , 22(3), 156. Descargado 2018-07-16, de <https://synapse.koreamed.org/D0Ix.php?id=10.4258/hir.2016.22.3.156> doi: 10.4258/hir.2016.22.3.156
- Djamaa, B., Richardson, M., Aouf, N., y Walters, B. (s.f.). Unicast/multicast performance in single-hop duty-cycled 6lowpan networks. En (pp. 140–145). IEEE. Descargado 2018-07-16, de <http://ieeexplore.ieee.org/document/6923813/> doi: 10.1109/CSNDSP.2014.6923813
- Dunkels, A. (s.f.). The ContikiMAC radio duty cycling protocol. , 11.
- Dunkels, A., Gronvall, B., y Voigt, T. (s.f.). Contiki - a lightweight and flexible operating system for tiny networked sensors. En *29th annual IEEE international conference on local computer networks* (pp. 455–462). doi: 10.1109/LCN.2004.38
- Eriksson, J., Finne, N., Tsiftes, N., Duquennoy, S., y Voigt, T. (s.f.). Scaling RPL to dense and large networks with constrained memory. En (pp. 126–134). Junction Publishing. Descargado 2018-07-17, de <http://libcon.rec.uabc.mx:3042/citation.cfm?id=3234847.3234863>
- Evans, D. (s.f.). How the next evolution of the internet is changing everything. , 11.

- Farooq, M. O., y Kunz, T. (s.f.). Contiki-based IEEE 802.15.4 channel capacity estimation and suitability of its CSMA-CA MAC layer protocol for real-time multimedia applications. , 2015, 1–9. Descargado 2018-07-16, de <http://www.hindawi.com/journals/misy/2015/398637/> doi: 10.1155/2015/398637
- Germanakos, P., Mourlas, C., y Samaras, G. (s.f.). A mobile agent approach for ubiquitous and personalized eHealth information systems. , 10.
- Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., y Levis, P. (s.f.). Collection tree protocol. En (p. 1). ACM Press. Descargado 2018-08-07, de <http://portal.acm.org/citation.cfm?doid=1644038.1644040> doi: 10.1145/1644038.1644040
- IEEE standards education. (s.f.). Descargado 2018-07-19, de <http://standards.ieee.org/about/stdsedu/index.html>
- Isomaki, M., Nieminen, J., Gomez, C., Shelby, Z., Savolainen, T., y Patil, B. (s.f.). *IPv6 over BLUETOOTH(r) low energy*. Descargado 2018-07-29, de <https://tools.ietf.org/html/rfc7668>
- Istepanian, R. S. H., Hu, S., Philip, N. Y., y Sungoor, A. (s.f.-a). The potential of internet of m-health things "m-IoT" for non-invasive glucose level sensing. , 2011, 5264–5266. doi: 10.1109/IEMBS.2011.6091302
- Istepanian, R. S. H., Hu, S., Philip, N. Y., y Sungoor, A. (s.f.-b). The potential of internet of m-health things "m-IoT" for non-invasive glucose level sensing. En *2011 annual international conference of the IEEE engineering in medicine and biology society* (pp. 5264–5266). doi: 10.1109/IEMBS.2011.6091302
- Jadhao, A. R., y Solapure, S. S. (s.f.). Analysis of routing protocol for low power and lossy networks (RPL) using cooja simulator. En (pp. 2364–2368). IEEE. Descargado 2018-07-16, de <http://ieeexplore.ieee.org/document/8300183/> doi: 10.1109/WISPNET.2017.8300183
- Johnson, A. (s.f.). *The internet is now officially too big as IP addresses run out*. Descargado 2018-07-30, de <https://www.nbcnews.com/news/us-news/internet-now-officially-too-big-ip-addresses-run-out-n386081>
- Kamel Boulos, M. N., y Al-Shorbaji, N. M. (s.f.). On the internet of things, smart cities and the WHO healthy cities. , 13(1), 10. Descargado 2018-07-16, de <http://ij-healthgeographics.biomedcentral.com/articles/10.1186/1476-072X-13-10> doi: 10.1186/1476-072X-13-10
- Kang, W. M., Moon, S. Y., y Park, J. H. (s.f.). An enhanced security framework for home appliances in smart home. , 7(1). Descargado 2018-08-03, de <http://hcis-journal.springeropen.com/articles/10.1186/s13673-017-0087-4> doi: 10.1186/s13673-017-0087-4

- King, G. (s.f.). *RDC phase optimization*. The Contiki Open Source OS for the Internet of Things. Descargado 2018-07-30, de <https://github.com/contiki-os/contiki> (original-date: 2012-10-24T05:59:36Z)
- Kumar, R., y Rajasekaran, M. P. (s.f.). An IoT based patient monitoring system using raspberry pi. En (pp. 1–4). IEEE. Descargado 2018-08-05, de <http://ieeexplore.ieee.org/document/7725378/> doi: 10.1109/ICCTIDE.2016.7725378
- Kushalnagar, N., Montenegro, G., Culler, D. E., y Hui, J. W. (s.f.). *Transmission of IPv6 packets over IEEE 802.15.4 networks*. Descargado 2018-07-16, de <https://tools.ietf.org/html/rfc4944>
- Lekidis, A., y Katsaros, P. (s.f.). Model-based design of energy-efficient applications for IoT systems. , 272, 24–38. Descargado 2018-07-16, de <http://arxiv.org/abs/1807.01242> doi: 10.4204/EPTCS.272.3
- Leone, R., Leguay, J., Medagliani, P., y Chaudet, C. (s.f.). Tee: Traffic-based energy estimators for duty-cycled wireless sensor networks. En (pp. 6749–6754). IEEE. Descargado 2018-07-16, de <http://ieeexplore.ieee.org/document/7249401/> doi: 10.1109/ICC.2015.7249401
- Liu, X., Sheng, Z., Yin, C., Ali, F., y Roggen, D. (s.f.). Performance analysis of routing protocol for low power and lossy networks (RPL) in large scale networks. , 4(6), 2172–2185. Descargado 2018-07-16, de <http://ieeexplore.ieee.org/document/8048510/> doi: 10.1109/JIOT.2017.2755980
- Manufacturers sharing patient-specific information from medical devices with patients upon request - guidance for industry and food and drug administration staff. (s.f.). , 5.
- McEwen, A., y Cassimally, H. (s.f.). *Designing the internet of things* (Reprinted with corrections ed.). Wiley. (OCLC: 862794270)
- Mehmood, T. (s.f.). COOJA network simulator: Exploring the infinite possible ways to compute the performance metrics of IOT based smart devices to understand the working of IOT based compression & routing protocols. , 7.
- Meola, A. (s.f.). *Why IoT, big data & smart farming are the future of agriculture*. Descargado 2018-08-03, de <https://www.businessinsider.com/internet-of-things--smart-agriculture-2016-10>
- Mohammadi, M., Al-Fuqaha, A., Sorour, S., y Guizani, M. (s.f.). Deep learning for IoT big data and streaming analytics: A survey. Descargado 2018-07-16, de <http://arxiv.org/abs/1712.04301>
- Olifer, Natalia, y Victori, O. (s.f.). *Redes de computadoras: principios, tecnología y protocolos para el diseño de redes*. McGraw-Hill Interamericana. Descargado 2018-07-18, de <https://libcon.rec.uabc.mx:4431/lib/uabcsp/detail.action?docID=3216837>.

- Palensky, P., y Kupzog, F. (s.f.). Smart grids. , 38(1), 201–226. Descargado 2018-07-16, de <http://www.annualreviews.org/doi/10.1146/annurev-environ-031312-102947> doi: 10.1146/annurev-environ-031312-102947
- Petersen, H., Baccelli, E., Wählisch, M., Schmidt, T. C., y Schiller, J. (s.f.). The role of the internet of things in network resilience. Descargado 2018-07-16, de <http://arxiv.org/abs/1406.6614>
- Postel, J. (s.f.). *Internet protocol*. Descargado 2018-07-17, de <https://tools.ietf.org/html/rfc791>
- ProQuest ebook central reader*. (s.f.). Descargado 2018-07-19, de <https://libcon.rec.uabc.mx:4431/lib/uabcsp/reader.action?docID=3216837&query=modelo+osi>
- Raji, R. S. (s.f.). Smart networks for control. , 31(6), 49–55. doi: 10.1109/6.284793
- Reinfurt, L., Breitenbücher, U., Falkenthal, M., Leymann, F., y Riegg, A. (s.f.). Internet of things patterns. En (pp. 1–21). ACM Press. Descargado 2018-07-16, de <http://dl.acm.org/citation.cfm?doid=3011784.3011789> doi: 10.1145/3011784.3011789
- Reusing, T. (s.f.). Comparison of operating systems TinyOS and contiki. doi: 10.2313/net-2012-08-2_02
- Rodrigues Cotrim, J., y Kleinschmidt, J. H. (s.f.). Performance evaluation of RPL on a mobile scenario with different ContikiMAC radio duty cycles. En (pp. 1–6). IEEE. Descargado 2018-07-16, de <http://ieeexplore.ieee.org/document/7968675/> doi: 10.1109/HPSR.2017.7968675
- San jose implements intel technology for a smarter city*. (s.f.). Descargado 2018-08-03, de <https://newsroom.intel.com/news-releases/san-jose-implements-intel-technology-for-a-smarter-city/>
- Schumacher, C. P. P., Kushalnagar, N., y Montenegro, G. (s.f.). *IPv6 over low-power wireless personal area networks (6lowpans): Overview, assumptions, problem statement, and goals*. Descargado 2018-07-17, de <https://tools.ietf.org/html/rfc4919>
- Shelby, Z. (s.f.). *6lowpan: The wireless embedded internet - part 1: Why 6lowpan?* Descargado 2018-07-30, de https://www.eetimes.com/document.asp?doc_id=1278794
- SIG INTRODUCES BLUETOOTH LOW ENERGY WIRELESS TECHNOLOGY THE NEXT GENERATION OF BLUETOOTH WIRELESS TECHNOLOGY* | *bluetooth technology website*. (s.f.). Descargado 2018-07-29, de <https://www.bluetooth.com/news/pressreleases/2009/12/17/sig-introduces-bluetooth-low-energy-wireless-technologythe-next-generation-of-bluetooth-wireless-technology>
- Skiba, D. J. (s.f.). The internet of things (IoT). , 34(1), 63–64. Descargado 2018-07-16, de <http://nlnjournals.org/doi/abs/10.5480/1536-5026-34.1.63> doi: 10.5480/1536-5026-34.1.63

- Stallings, W. (s.f.). *Operating systems internals and desing principles* (Seventh Edition ed.). Pearson. Descargado 2018-08-05, de http://dinus.ac.id/repository/docs/ajar/Operating_System.pdf
- Stehlík, M. (s.f.). Comparison of simulators for wireless sensor networks. , 92.
- Tantayakul, K., Kamolphiwong, S., y Angchuan, T. (s.f.). IPv6@HOME. En (p. 92). ACM. Descargado 2018-07-17, de <http://libcon.rec.uabc.mx:3042/citation.cfm?id=1506270.1506383> doi: 10.1145/1506270.1506383
- tedhudek. (s.f.). *Windows network architecture and the OSI model*. Descargado 2018-07-19, de <https://docs.microsoft.com/en-us/windows-hardware/drivers/network/windows-network-architecture-and-the-osi-model>
- What is mbed? | mbed*. (s.f.). Descargado 2018-07-16, de www.mbed.com/en
- Winter {\textless}wintert@acm.org{\textgreater}, T. (s.f.). *RPL: IPv6 routing protocol for low-power and lossy networks*. Descargado 2018-07-29, de <https://tools.ietf.org/html/rfc6550>
- Wireshark · frequently asked questions*. (s.f.). Descargado 2018-08-09, de <https://www.wireshark.org/faq.html#q1.2>
- x Working Group, T. T. . (s.f.). TinyOS 2.0. En (pp. 320–320). ACM. Descargado 2018-07-16, de <http://libcon.rec.uabc.mx:3042/citation.cfm?id=1098918.1098985> doi: 10.1145/1098918.1098985
- Youssef, M. F., Elsayed, K. M. F., y Zahran, A. H. (s.f.). Adaptive radio duty cycling in ContikiMAC: Proposal and analysis. En (pp. 491–495). IEEE. Descargado 2018-07-16, de <http://ieeexplore.ieee.org/document/6803216/> doi: 10.1109/WF-IoT.2014.6803216
- Zhang, T., y Li, X. (s.f.). Evaluating and analyzing the performance of RPL in contiki. En (pp. 19–24). ACM. Descargado 2018-07-17, de <http://libcon.rec.uabc.mx:3042/citation.cfm?id=2633675.2633678> doi: 10.1145/2633675.2633678
- Zyga, L. (s.f.). *Internet growth follows moore's law too*. Descargado 2018-07-10, de <https://phys.org/news/2009-01-internet-growth-law.html>