

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



---

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

---

**Evaluación del impacto del ciclo de trabajo del transceptor de  
radio al proceso de formación de redes inalámbricas de  
sensores multisalto en el contexto de la Internet de las Cosas  
(IoT)**

Tesis

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

Presenta:

**Rodolfo Alejandro Heron Cárdenas**

Ensenada, Baja California, México  
2019

Tesis defendida por  
**Rodolfo Alejandro Heron Cárdenas**

y aprobada por el siguiente Comité

---

**Dr. Gabriel Alejandro Galaviz Mosqueda**  
Codirector de tesis

---

**Dr. Salvador Villarreal Reyes**  
Codirector de tesis

Miembros del comité

**Dr. Roberto Conte Galván**

**Dra. Ana Laura Padilla Ortiz**

**Dr. Raúl Rivera Rodríguez**

**Dr. Jesús Favela Vara**



---

**Dr. Daniel Saucedo Carvajal**  
Coordinador del Posgrado en Electrónica y  
Telecomunicaciones

---

**Dra. Rufina Hernández Martínez**  
Directora de Estudios de Posgrado

*Rodolfo Alejandro Heron Cárdenas © 2019*

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

Resumen de la tesis que presenta **Rodolfo Alejandro Heron Cárdenas** como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con orientación en Telecomunicaciones.

**Evaluación del impacto del ciclo de trabajo del transceptor de radio al proceso de formación de redes inalámbricas de sensores multisalto en el contexto de la Internet de las Cosas (IoT)**

Resumen aprobado por:

---

Dr. Gabriel Alejandro Galaviz Mosqueda  
**Codirector de tesis**

---

Dr. Salvador Villarreal Reyes  
**Codirector de tesis**

El paradigma de redes definidas por software (SDN) promete traer beneficios y ser una alternativa de solución para los retos que presentan las redes inalámbricas de sensores (WSN) pertenecientes al Internet de las cosas, por lo cual se han realizado esfuerzos para adaptar este paradigma a los requerimientos de las WSN para formar una red SDWSN. El reto más importante en las WSN es la energía debido a que los nodos están operados por baterías y el elemento que presenta mayor consumo de energía es el de radio de comunicación; esto limita el tiempo de vida del nodo en la red. Por ello se recomienda utilizar técnicas de ciclo de trabajo para disminuir este consumo. El enfoque SDWSN reportado en la plataforma *Software Defined Networking solution for Wireless Sensor Networks (SDN-WISE)* está diseñado bajo el estándar de las WSN i.e, IEEE 802.15.4 y adapta una capa de ciclo de trabajo en su pila de protocolos, esto lo hace ser el enfoque SDWSN más completo observado en la literatura. Sin embargo, reducir el consumo de energía mediante técnicas de ciclo de trabajo puede presentar costos al proceso de formación y descubrimiento de diferentes topologías en una red SDWSN. Las evaluaciones de las redes SDWSN observadas en la literatura no toman en cuenta el impacto de agregar una capa de este tipo para formar la red. Es por ello que este trabajo de tesis propone evaluar el proceso de formación de una red SDWSN mediante el proceso de descubrimiento de topología del enfoque de SDN-WISE, para determinar el desempeño de utilizar el paradigma de SDN sobre la WSN y medir el costo de utilizar técnicas de ciclo de trabajo en el radio en términos de consumo de energía, tiempo de convergencia y cantidad de paquetes de control generados. La evaluación es llevada a cabo utilizando el simulador de redes Cooja del sistema operativo Contiki y los resultados de esta evaluación son comparados con el proceso de formación de la red del protocolo de enrutamiento de facto de las WSN i.e. *Routing Protocol for Low-Power and Lossy Networks (RPL)*.

**Palabras clave:** Redes Definidas por Software, WSN, SDWSN, SDN-WISE, ContikiMac, Contiki Os, descubrimiento de topología

Abstract of the thesis presented by **Rodolfo Alejandro Heron Cárdenas** as a partial requirement to obtain the Master of Science degree in Electronics and Telecommunications with orientation in Telecommunications

**Evaluation of the impact of the duty cycle on the formation of multihop wireless sensor network process considering the context of the Internet of Things (IoT)**

Abstract approved by:

---

Dr. Gabriel Alejandro Galaviz Mosqueda  
**Thesis Codirector**

---

Dr. Salvador Villarreal Reyes  
**Thesis Codirector**

The paradigm of Software Defined Networks promises to bring benefits and being an alternative of solution to the Wireless Sensor Networks (WSN) challenges that are involved in the Internet of Things. Many approaches have been done to adapt this paradigm to the requirements of the WSN to form an SDWSN network. The most important challenge of the WSN is energy, due to the network nodes are operated by batteries and the element that brings more consumption of energy is the communication radio, which limits the lifetime of the node in the network. Therefore it is recommended to use duty cycle techniques to decrease this consumption. The SDN approach presented in *Software Defined Networking solution for Wireless Sensor Networks* (SDN-WISE) is designed under the WSN standard, i.e. IEEE 802.15.4 and adapts a duty cycle layer in its network stack. That makes it the most complete approach seen in the literature. However, reducing the energy consumption by duty cycle techniques could bring some tradeoffs to the formation and discovery of different topology process in an SDWSN network. The evaluations of the SDWSN networks observed in the literature don't take into account the impact of aggregating a layer of this type to form the network. It is for this reason, that this work of thesis proposes to evaluate the formation of SDWSN network process using the Topology Discovery process of SDN-WISE, in order to determine the performance of using the SDN paradigm over the WSN and measure the cost of using duty cycle techniques on the radio, in terms of energy consumption, convergence time and overhead. The evaluation is made using the network simulator Cooja of ContikiOS and the evaluation results are compared with the formation of the network process of the de facto routing protocol of the WSN i.e. *Routing Protocol for Low-Power and Lossy Networks* (RPL).

**Keywords: Software Defined Networks, WSN, SDWSN, SDN-WISE, ContikiMac, Contiki Os, Topology Discover**

## Dedicatoria

---

A mis padres **Rodolfo Alejandro Heron Castillo** y **Zita Araceli Cárdenas Rojas** por siempre impulsarme sobre sus hombros y motivarme para llegar a lo más alto y a mí hermana **Denisse Ashley Heron Cárdenas**, por su amor, motivación y soporte.

*–Sobre hombros de Gigantes –*

*–2<sup>da</sup> Corintios 3 : 5 –*

## Agradecimientos

---

Doy gracias primeramente a Dios, su voluntad me ha guiado hasta aquí. Él es quien me ha brindado todo el conocimiento, permitido lograr las metas y superar los retos.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) con número de becario 634281 por el apoyo económico proporcionado durante los dos años de mis estudios de maestría.

Al Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE) por permitirme tenerla como casa por más de dos años en donde he encontrado conocimiento y a partir de ahí aportar más conocimiento.

Quiero expresar mi más sincero agradecimiento a mis directores de tesis, al Dr. Gabriel Alejandro Galaviz Mosqueda y al Dr. Salvador Villarreal Reyes por permitirme entrar a experimentar el saber científico y la investigación, gracias por la paciencia, la confianza y la guía para resolver los retos presentados en este camino sinuoso, ayudándome a adquirir no solo las herramientas necesarias, y sobre todo muchas gracias por brindarme su amistad. El que comparte conocimiento genera más conocimiento.

A los miembros de mi comité de tesis: Dra. Ana Laura Padilla Ortiz, Dr. Roberto Conte Galván, Dr. Raúl Rivera Rodríguez y Dr. Jesús Favela Vara por sus oportunos consejos, comentarios, y observaciones que ayudaron a lograr realizar este trabajo de investigación.

Un agradecimiento especial a Elizabeth Avilés por desempeñar con gran pasión su trabajo y realizar la revisión bibliográfica exhaustiva, además de proporcionar comentarios oportunos para mejorar la presentación de este trabajo de tesis.

A los profesores del posgrado en Electrónica y Telecomunicaciones que compartieron un gran número de enseñanzas y abrieron mi panorama del conocimiento en telecomunicaciones con sus clases.

A los integrantes del grupo de trabajo ARTS en especial al M.C. Christian Soto, al M.C. Manuel Casillas y al Dr. Aldo Ramos, con sus oportunas recomendaciones, consejos y ayuda hicieron que el trayecto de este trabajo de tesis se facilitara, además muchas gracias por brindarme su amistad.

A mis padres, hermana y familia quienes sé que contaré con su amor incondicional siempre, me han apoyado holísticamente, los logros que Dios me ha permitido tener son de ustedes.

Un sincero agradecimiento desde mi corazón a la familia Murillo Cossio: Alma, Guillermo, Eliu y Vanessa; a la familia Villegas Rodríguez: Irma, Armando, Elisa, Paloma, Esteban, mami Luz y hermano Leonardo<sup>†</sup> y a la familia Verdugo Aguilar: Jonnathan y Diana por permitirme encontrar en ustedes un segundo hogar,

han sido parte fundamental en mi formación académica de maestría, tenerlos en mi vida es una gran bendición de Dios. Gracias por todos los consejos, ánimos, amistad, cariño, cuidado y una lista sin fin de momentos que han compartido conmigo. Siempre los llevaré en mi corazón.

A mis compañeros y amigos que realicé durante el posgrado, muchas gracias por todos los momentos compartidos, ustedes han sido de ayuda para mi crecimiento profesional, especial agradecimiento a Beatriz, Iván, Elizvan, Gustavo, Enrique, Miguel, Edgar, Adán, Alejandro, Eduardo, Neftali y Marisela.

Por último y no menos importante a los amigos que realicé durante mi estancia en Ensenada, muchas gracias por permitirme formar parte de su vida: Elisa, Diana Aguilar, Diana Rodríguez, Toñito, David, Eliu, Jonnathan, Dania y Brenda.

## Tabla de contenido

---

Resumen en español .....	ii
Resumen en inglés .....	iii
Dedicatoria .....	iv
Agradecimientos .....	v
Lista de figuras .....	x
Lista de tablas.....	xiii
<b>Capítulo 1. Introducción .....</b>	<b>1</b>
1.1 Antecedentes .....	2
1.2 Planteamiento del problema .....	5
1.4 Objetivos .....	8
1.4.1 Objetivo general.....	8
1.4.2. Objetivos específicos.....	8
1.5 Metodología .....	9
1.6 Organización de la tesis.....	10
<b>Capítulo 2. Procesos de formación de la WSN .....</b>	<b>12</b>
2.1 Introducción .....	12
2.2 Redes WSN con enrutamiento RPL .....	12
2.2.1 Mensajes de control para la formación de una red en RPL .....	14
2.2.2 Formación de una red RPL .....	17
2.3 Redes SDWSN.....	18
2.3.1 Redes SDN .....	19
2.3.2 Redes SDN sobre WSN .....	21
2.4 Redes WSN utilizando SDN-WISE.....	26
2.4.1 Mensajes de control para la formación de la red SDN-WISE .....	28
2.4.2 Formación de una red SDN-WISE.....	29
<b>Capítulo 3. Herramientas del entorno del simulador.....</b>	<b>32</b>
3.1 Introducción .....	32
3.2 Contiki Os .....	32
3.3 Contiki Os – Pila de protocolos.....	32



3.3.1. Contiki Os- Capa de red.....	33
3.3.2. Contiki Os - Capa de acceso al medio.....	34
3.3.3 Contiki Os- Capa de RDC.....	35
3.3.4 Contiki Os- Capa de Empaquetamiento .....	40
3.3.5 Contiki Os- Capa de Radio .....	41
3.4 Simulador de redes Cooja .....	41
3.5 Módulo de medición de energía. ....	44
<b>Capítulo 4. Configuraciones de evaluación.....</b>	<b>47</b>
4.1 Introducción .....	47
4.2 Configuraciones de la red RPL.....	47
4.3 Configuraciones de la red SDWSN .....	50
4.4 Métricas de evaluación de desempeño .....	54
4.4.1 Tiempo de Convergencia.....	54
4.4.2 Consumo de Energía.....	54
4.4.3 Overhead de la red.....	55
4.5 Escenarios de evaluación .....	57
4.5.1 Escenario #1 Topología multsalto .....	57
4.5.2 Escenario #2 Topología de tipo malla a un solo salto .....	58
4.5.3 Escenario #3 Topología combinada múltiples usuarios – múltiples saltos .....	59
4.5.4 Escenario #4 Topología basada en una aplicación.....	60
4.6 Validación del simulador .....	62
4.6.1 Escenario de simulación de validación.....	62
4.6.2 Comparativa con una cama de pruebas.....	67
4.6.2.1 CC2560STK SensorTag.....	67
4.6.2.2 Zolertia Z1.....	67
4.6.2.3 Cama de pruebas y configuraciones para validación por implementación. ....	69
4.7 Metodología de evaluación.....	75
4.7.1 Efecto de ciclo de trabajo sobre formación de red en SDWSN.....	77
4.7.2 Mecanismo de re-distribución de paquete Openpath.....	79

<b>Capítulo 5. Análisis del impacto del ciclo de trabajo en la formación de la red</b> .....	82
5.1 Introducción .....	82
5.2 Análisis del impacto del ciclo de trabajo en formación de la red SDWSN en comparación con la red RPL.....	82
5.2.1 Análisis de resultados comparativos para escenario #1 Topología multisalto .....	84
5.2.2 Análisis de resultados comparativos para escenario #2 Topología multi-usuario.....	91
5.2.3 Análisis de resultados comparativos para escenario #3 Topología combinada múltiples saltos-múltiples usuarios .....	99
5.2.4 Análisis de resultados comparativos para escenario #4 Topología con aplicación.....	106
5.2.5 Conclusiones de análisis de resultados .....	114
<b>Capítulo 6. Conclusiones y trabajo a futuro</b> .....	118
6.1 Introducción .....	118
6.2 Conclusiones sobre validación del simulador .....	118
6.3 Conclusiones sobre formación de la red SDWSN en comparación con RPL e impacto de ciclo de trabajo.....	119
6.4 Trabajo a futuro .....	120
<b>Literatura citada</b> .....	122

## Lista de figuras

---

Figura	Página
1	Ejemplo de aplicación m-IoT. ....1
2	Comparación de topología estrella y malla. ....2
3	Comparación de un enrutamiento distribuido con un enrutamiento centralizado.....4
4	Agenda de ciclo de trabajo del radio transceptor en nodos sensores. ....6
5	Representación de una DODAG en RPL.....13
6	Modos de operación en enrutamiento RPL a) Modo no storing de RPL b) Modo storing de RPL. ....16
7	a) Proceso de formación de la DODAG en RPL b) DODAG multi salto. ....18
8	Arquitectura de redes definidas por software. ....20
9	Topología de SDN sobre topología WSN: Formación SDWSN. ....22
10	Arquitectura de SDN propuesta por SensorOpenflow (Luo et al, 2012). ....23
11	Arquitectura de SDN propuesta por SDWN (Costanzo et al., 2012). ....24
12	Arquitectura de SDN propuesta por SDN-WISE (Galluccio et al., 2015).....27
13	Módulos para formación de topología de SDWSN.....30
14	a) Proceso de descubrimiento de topología en red SDN-WISE, b) Topología multisalto SDN-WISE. ....31
15	Pila de protocolos de ContikiOs.....33
16	Algoritmo de CSMA-CA no ranurado.....35
17	Transmisión de paquetes unicast con ContikiMac. ....36
18	Transmisión de paquetes broadcast con ContikiMac. ....37
19	Esquema de Distribución de tiempo para transmisión de paquetes. ....38
20	Transmisión de bloqueo de fase por parte de ContikiMAC. ....40
21	Entorno de simulación Cooja.....41
22	Herramienta del simulador : Radio message. ....42
23	Modelo UDGM: rango de cobertura (área verde), rango de interferencia (área gris) y tasas de transmisión y recepción. ....43
24	Comunicación entre el resumidero y el controlador en red SD-WISE. ....52
25	Plugin controlador SDN-WISE. a) Consola del controlador b) Ventana de grafica de la red.....52
26	Escenario de topología de múltiples saltos .....57
27	Escenario de topología de múltiples usuarios.....58
28	Escenario de topología de múltiples usuarios y múltiples saltos. ....59
29	Escenario de topología de aplicación de monitoreo y cuidado de adulto mayor.....61
30	Enfoque de validación del simulador Cooja. ....62
31	Escenario para validación por medio de simulador. ....63
32	Comparativa de pérdida de paquetes promedio de entre simulador Omnet++ y Cooja.....66
33	Dispositivos utilizados para validación a) SensortTag CC2650 b) Z1 Zolertia. ....68
34	Escenario de validación por implementación. ....69
35	Distribución de canales para tecnología IEEE 802.15.4 e IEEE 802.11. ....70

36	Espectro del rango de frecuencias de 2.402 a 2.485 GHz con interferencia en el canal a) Espectro obtenido mediante la herramienta WiSpy b) Espectro obtenido mediante analizador de espectros. ....	71
37	Espectro del rango de frecuencias de 2.402 a 2.485 GHz sin interferencia en el canal 25 a) Espectro obtenido mediante la herramienta WiSpy b) Espectro obtenido mediante analizador de espectros. ....	72
38	Pérdida de paquetes en validación por implementación con escenarios de interferencia y administración de energía. ....	73
39	Pérdida de paquetes en validación por implementación en un escenario lo mayormente posible controlado. ....	74
40	Modelo de ejecución de simulación y análisis. ....	75
41	Distribución de convergencia de la red SDN-WISE para escenario multisalto. ....	77
42	Distribución de convergencia de la red SDN-WISE para escenario con escenario de múltiples usuarios. ....	77
43	Distribución de convergencia de la red SDN-WISE para escenario de múltiples saltos-múltiples usuarios. ....	78
44	Distribución de convergencia de la red SDN-WISE para escenario de aplicación de monitoreo adulto mayor (múltiples usuarios). ....	78
45	Algoritmo de redistribución de paquetes Openpath perdidos. ....	81
46	Comparación energía consumida. Topología multisalto, configuración R20By. ....	84
47	Comparación del tiempo de convergencia. Topología multisalto, configuración R20By. ....	84
48	Comparación cantidad de paquetes de control. Topología multisalto, configuración R20By. ....	84
49	Comparación energía consumida. Topología multisalto, configuración R10By. ....	86
50	Comparación del tiempo de convergencia. Topología multisalto, configuración R10By. ....	86
51	Comparación cantidad de paquetes de control. Topología multisalto, configuración R10By. ....	86
52	Comparación energía consumida. Topología multisalto, configuración R5By. ....	89
53	Comparación del tiempo de convergencia. Topología multisalto, configuración R5By. ....	89
54	Comparación cantidad de paquetes de control. Topología multisalto, configuración R5By. ....	89
55	Comparación energía consumida. Topología multi-usuario, configuración R20By. ....	92
56	Comparación del tiempo de convergencia. Topología multi-usuario, configuración R20By. ....	92
57	Comparación cantidad de paquetes de control. Topología multi-usuario, configuración R20By. ....	92
58	Comparación energía consumida. Topología multi-usuario, configuración R10By. ....	94
59	Comparación del tiempo de convergencia. Topología multi-usuario, configuración R10By. ....	94
60	Comparación cantidad de paquetes de control. Topología multi-usuario, configuración R10By. ....	95
61	Comparación energía consumida. Topología multi-usuario, configuración R5By. ....	97
62	Comparación del tiempo de convergencia. Topología multi-usuario, configuración R5By. ....	97
63	Comparación cantidad de paquetes de control. Topología multi-usuario, configuración R5By. ....	97
64	Comparación del tiempo de convergencia. Topología multi-usuario-multi-salto, configuración R20By. ....	99
65	Comparación de energía consumida. Topología multi-usuario-multi-salto, configuración R20By. ....	99
66	Comparación de cantidad de paquetes de control Topología multi-usuario-multi-salto, configuración R20By. ....	100
67	Comparación de energía consumida. Topología multi-usuario-multi-salto, configuración R10By. ....	101

68	Comparación del tiempo de convergencia. Topología multi-usuario-multi-salto, configuración R10By.....	101
69	Comparación de cantidad de paquetes de control. Topología multi-usuario-multi-salto, configuración R10By.....	102
70	Comparación de energía consumida. Topología multi-usuario-multi-salto, configuración R5By.....	104
71	Comparación del tiempo de convergencia. Topología multi-usuario-multi-salto, configuración R5By.....	104
72	Comparación de cantidad de paquetes de control. Topología multi-usuario-multi-salto, configuración R5By.....	104
73	Comparación de energía consumida. Topología aplicación, configuración R20By.....	107
74	Comparación del tiempo de convergencia. Topología aplicación, configuración R20By.....	107
75	Comparación de cantidad de paquetes. Topología aplicación, configuración R20By.....	107
76	Comparación del tiempo de convergencia. Topología aplicación, configuración R10By.....	109
77	Comparación de energía consumida. Topología aplicación, configuración R10By.....	109
78	Comparación de cantidad de paquetes de control. Topología aplicación, configuración R10By.....	109
79	Comparación de energía consumida. Topología aplicación, configuración R5By.....	111
80	Comparación del tiempo de convergencia. Topología aplicación, configuración R5By.....	111
81	Comparación de cantidad de paquetes de control. Topología aplicación, configuración R5By.....	112

## Lista de tablas

---

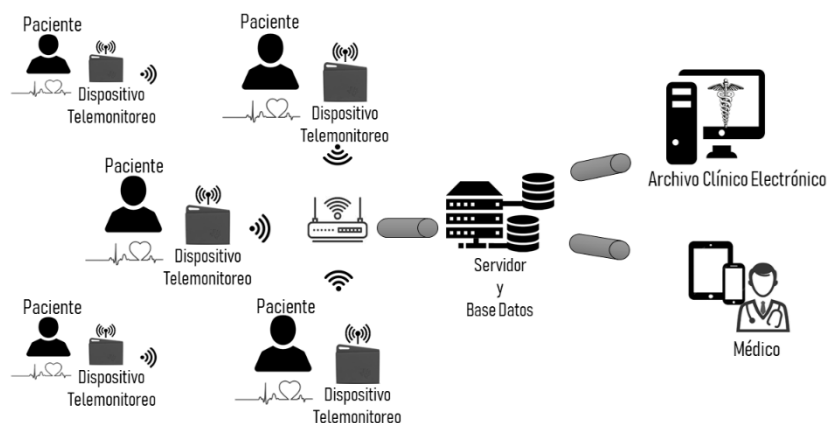
Tabla	Página
1 Descripción de mensajes de control de RPL.....	15
2 Comparativa de soluciones SDN sobre WSN.....	25
3 Campos del encabezado del paquete de SDN-WISE. ....	28
4 Mensajes de control para red SDWSN. ....	29
5 Cantidad de paquetes generados por ContikiMac.....	37
6 Modos de operación Powertrace.....	45
7 Análisis de trama Powertrace.....	45
8 Parámetros de configuración para el protocolo de RPL.....	49
9 Configuraciones utilizadas para la evaluación de la red SDWSN. ....	53
10 Distribución de topología multi saltos.....	58
11 Distribución de topología un solo salto.....	59
12 Distribución de topología multi usuario y muti salto. ....	60
13 Distribución de topología aplicación. ....	61
14 Parámetros de configuración del mecanismo de CSMA no ranurado. ....	63
15 Parámetros de configuración para validación por simulador. ....	64
16 Análisis de empaquetamiento para validación del simulador. ....	65
17 Características de tarjetas de desarrollo para evaluación. ....	68
18 Parámetros de configuración para evaluación de implementación. ....	70
19 Parámetros de configuración de evaluación de redes SDWSN y RPL.´ .....	75
20 Parámetros de configuración de evaluación de redes SDWSN y RPL. ....	76
21 Configuración de periodicidad de paquetes de control para red SDWSN. ....	83
22 Diferencia de tiempo de convergencia de la red (Topología #1) entre configuración R20By y configuración de periodicidad en RPL. ....	85
23 Diferencia de tiempo de convergencia de la red (Topología #1) entre configuración R10By y configuración de periodicidad en RPL. ....	87
24 Diferencia de porcentaje de consumo de energía de la red (Topología #1) entre configuración R20By y R10By. ....	88
25 Diferencia de tiempo de convergencia de la red (Topología #1) entre configuración R5By y configuración de periodicidad en RPL. ....	90
26 Diferencia de porcentaje de consumo de energía de la red (Topología #1) entre configuración R10By y R5By.....	91
27 Diferencia de porcentaje de consumo de energía de la red (Topología #1) entre configuración R10By y R5By.....	91
28 Diferencia de tiempo de convergencia de la red (Topología #2) entre configuración R20By y configuración de periodicidad en RPL. ....	93
29 Diferencia de tiempo de convergencia de la red (Topología #2) entre configuración R10By y configuración de periodicidad en RPL. ....	95
30 Diferencia de porcentaje de consumo de energía de la red (Topología #2) entre configuración R20By y R10By. ....	96

31	Diferencia de tiempo de convergencia de la red (Topología #2) entre configuración R5By y configuración de periodicidad en RPL. ....	98
32	Diferencia de porcentaje de consumo de energía de la red (Topología #2) entre configuración R10By y R5By.....	98
33	Diferencia de tiempo de convergencia de la red (Topología #3) entre configuración R20By y configuración de periodicidad en RPL. ....	100
34	Diferencia de tiempo de convergencia de la red (Topología #3) entre configuración R10By y configuración. de periodicidad en RPL.....	102
35	Diferencia de porcentaje de consumo de energía de la red (Topología #3) entre configuración R20By y R10By. ....	103
36	Diferencia de tiempo de convergencia de la red (Topología #3) entre configuración R5By y configuración de periodicidad en RPL. ....	105
37	Diferencia de porcentaje de consumo de energía de la red (Topología #3) entre configuración R10By y R5By.....	105
38	Diferencia de tiempo de convergencia de la red (Topología #4) entre configuración R20By y configuración de periodicidad en RPL. ....	107
39	Diferencia de tiempo de convergencia de la red (Topología #4) entre configuración R10By y configuración de periodicidad en RPL. ....	110
40	Diferencia de porcentaje de consumo de energía de la red (Topología #4) entre configuración R20By y R10By.....	110
41	Diferencia de tiempo de convergencia de la red (Topología #4) entre configuración R5By y configuración de periodicidad en RPL. ....	112
42	Diferencia de porcentaje de consumo de energía de la red (Topología #4) entre configuración R10By y R5By.....	113
43	Análisis de resultados de desempeño en la formación de la red.....	116
44	Características de desempeño de la red RPL y SDWSN sobre topologías.....	117

## Capítulo 1. Introducción

Hoy en día nuestra vida se ve impactada por la tecnología, prueba de ello se observa en el paradigma de la “Internet de las Cosas” (*IoT*, por sus siglas en inglés), el cual refiere a la interconexión de múltiples dispositivos a Internet intercambiando información entre sí mediante el desarrollo de sensores inteligentes, tecnologías de comunicación y protocolos (Al-Fuqaha et al., 2015). Así, se proporcionan diferentes soluciones a nuevas aplicaciones que tienen un papel notable en el mundo, como la automatización de la industria (Xu et al., 2014), aplicaciones en el transporte (Bajaj et al., 2018), en el hogar (Lohan y Singh, 2019), en la medicina (Yacchirema y Esteve, 2017), entre otras (Giusto et al., 2010; Shahid y Aneja, 2012; Gubbi et al., 2013; Capella et al., 2016).

Una de las aplicaciones del IoT es el cuidado de la salud referido como “m-*IoT*” y se logra mediante la utilización de redes inalámbricas de sensores (*WSN* por sus siglas en inglés), dando lugar al monitoreo de señales fisiológicas (ECG, temperatura, presión, oxigenación en la sangre, etc.) de pacientes hospitalizados, un ejemplo de esta red es la que se observa en la figura 1.



**Figura 1** Ejemplo de aplicación m-*IoT*.

En (Mendivil, 2017) se evalúa un sistema de monitorización del cuidado de la salud enfocado a adultos mayores a través de WSN. En dicho trabajo utilizan tecnologías de largo alcance dentro del contexto IoT y el sistema de monitorización da información de localización de la persona en caso de que sufra alguna caída. Además, también se muestra el perfil de tráfico de las señales utilizadas para su aplicación.

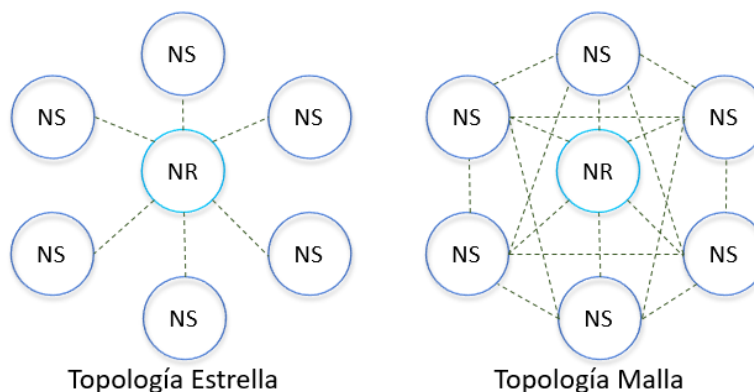


Existen diversas tecnologías inalámbricas que han sido utilizadas en las redes inalámbricas de sensores como Zigbee, Bluetooth Low Energy, 6LoWPAN, Z-Wave (Ko et al., 2010; Redondi et al., 2013; Hussain et al., 2015) las cuales utilizan diferentes algoritmos para formar la WSN.

## 1.1 Antecedentes

Las redes inalámbricas de sensores están compuestas por múltiples nodos individuales que interactúan con su entorno al medir, controlar y obtener información de parámetros como; temperatura; presión y demás variables físicas (Dludla et al., 2013). Dichos nodos están compuestos por diferentes funciones como cálculo de operaciones, sensado, almacenamiento, y comunicación inalámbrica, de tal forma que estén contenidos en dispositivos de bajo costo, pequeña escala y organizados en una topología para lograr una tarea en común (Karl y Willig, 2006). Estas WSNs están compuestas principalmente por múltiples Nodos Sensores (NS) que se comunican entre sí dentro de su rango de cobertura, enviando la información sensada del medio hacia uno o varios Nodos Resumideros (NR).

Las WSNs pueden tener diferentes topologías, entre las más relevantes por ser las más utilizadas están las topologías de tipo estrella y malla (ver Figura 2). En la topología tipo estrella se facilita el proceso de establecimiento de la red y de enrutamiento, considerando que solo existe una sola entidad (el NR) encargada de gestionar el ingreso de nuevos nodos a la red. Sin embargo, un problema importante con la topología estrella es la escalabilidad, cuando el número de nodos crece es necesario una mayor cantidad de recursos del resumidero para gestionar la red. Además, la topología tipo estrella limita el alcance de la red al área de cobertura del resumidero.



**Figura 2** Comparación de topología estrella y malla.

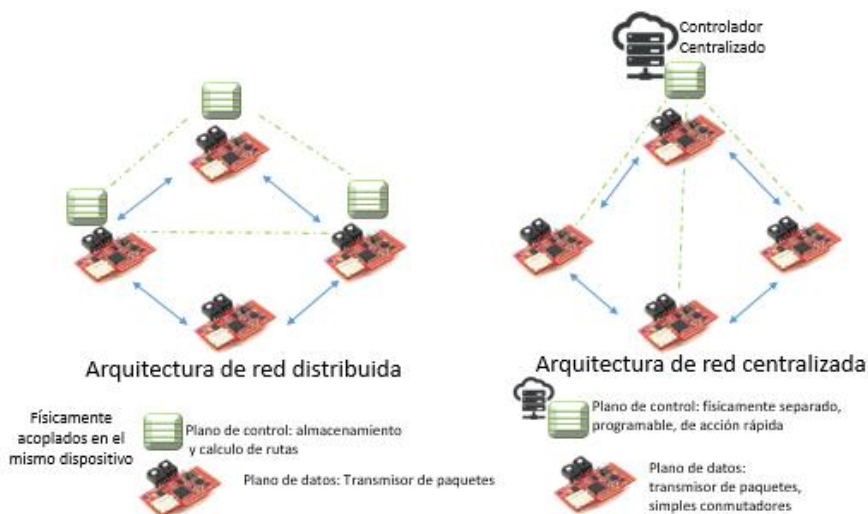
Por lo anterior, se han considerado otras topologías de red para el despliegue de una WSN. Una de las más comunes es la topología en malla, en esta topología los nodos están interconectados entre si y pueden llegar al resumidero a un solo salto o a través de múltiples saltos. Es importante mencionar que en la topología malla, la entrada de nuevos nodos se puede dar de forma distribuida (directamente por un nodo que ya es parte de la red). La topología de red tipo malla ofrece diversas ventajas sobre la tipo estrella, como una comunicación más confiable y generación de diferentes caminos para el envío de datos, con ello se logra menor probabilidad de pérdida de paquetes (Sharma et al., 2013; Farej y Abdul-hameed, 2015). Sin embargo, deben diseñarse mecanismos de descubrimiento de topología y de enrutamiento para los nodos que no están dentro del área de cobertura del resumidero y poder dar formación a la red.

El protocolo de enrutamiento es un proceso en el cual se seleccionan los caminos adecuados para que la información viaje desde una fuente a un destino. Estos protocolos de enrutamiento deben considerar las limitaciones inherentes de las WSN para poder formar las rutas más eficientes bajo diversos criterios, como el consumo de energía, el nivel del enlace o el número de transmisiones esperadas, con el objetivo de mejorar la eficiencia de la transmisión. Estos algoritmos pueden ser clasificados tomando en cuenta la importancia de los datos, la jerarquía de los nodos o la localización de los mismos (Shabbir y Rizwan, 2017). Todas las clasificaciones de los protocolos de enrutamiento que existen para las WSN convergen en dos tipos: algoritmos de enrutamiento centralizados y algoritmos de enrutamiento distribuidos. En el proceso de realizar el enrutamiento, ambos son considerados como protocolos dinámicos donde se intercambia información de forma periódica para recolectar el estado de la red.

En los protocolos de enrutamiento distribuido, los nodos en la red operan en un modo par a par (*peer-to-peer*) sin tener un nodo central. Cada NS obtiene el estado de la red de otros NS y hacen el cálculo de la ruta con la información obtenida. De esta forma, cada NS en la red tiene que mantener tablas de enrutamiento y ejecutar por sí mismo un algoritmo para calcular la ruta.

Por otro lado, un nodo central es elemento clave para el protocolo de enrutamiento centralizado, el cual recolecta la información de la red por medio de los otros NS. El diagrama de la red con la información de todos los nodos es configurado en un solo nodo. Posterior a esto, el nodo central calcula el camino más corto, en base a cierta métrica, para cada NS con sus posibles destinos en la red y disemina la información a cada uno.

La figura 3 muestra una idea gráfica de cómo es la descripción de estos protocolos sobre una WSN.



**Figura 3** Comparación de un enrutamiento distribuido con un enrutamiento centralizado.

El proceso de formación de la red no es una tarea trivial, los nodos en la red necesitan formar una topología para comunicarse entre ellos y así hacer la red operable, además que una vez formada la ruta, la red debe seguir un mantenimiento por si nuevas rutas son necesarias. Esto hace que el protocolo de enrutamiento sea uno de los elementos más importantes en el diseño de una WSN. El protocolo actualmente considerado como el estándar de facto en las WSN es el desarrollado por el grupo de trabajo de la IETF nombrado como Protocolo de Enrutamiento para Redes con Baja energía y con Pérdidas (RPL por sus siglas en inglés) (IETF, 2012). Este protocolo presenta un enrutamiento distribuido y permite a los paquetes con direccionamiento IPv6 provenientes de Internet ser enviados sobre los protocolos del estándar IEEE 802.15.4 (estándar de comunicación que define la capa física y de acceso al medio de redes inalámbricas de área personal con bajas tasas de transmisión de datos) en redes llamadas 6LoWPAN.

Sin embargo, con el advenimiento de nuevos retos en la administración, uso y despliegue de las WSN por parte de los paradigmas como el Internet de las Cosas, se necesitan nuevas soluciones innovadoras que proporcionen prestaciones adecuadas y ayuden a minimizar los retos que las WSN presentan, abriendo aún más su campo de aplicación. Motivado por los próximos retos a enfrentarse se ha desarrollado la inserción del paradigma de redes definidas por software (SDN por sus siglas en inglés) a las WSN.

Las redes definidas por software son un paradigma de redes que nació con el objetivo de determinar el camino de los paquetes de red sobre conmutadores dentro de una red alámbrica. Este paradigma de redes,

propone traer una separación de funciones de los elementos de la red, donde centralizará el plano de control de cada dispositivo en uno solo, dejando al resto de dispositivos solamente con el plano de datos, este paradigma está basado en un enrutamiento centralizado.

El paradigma de SDN ha sido utilizado para ser desplegado en diversos tipos de redes en lugar de las arquitecturas de red tradicional en búsqueda de aprovechar sus beneficios (Wang et al., 2016), por ejemplo redes de área de campus (Mckeown et al., 2008), redes celulares (Li et al., 2012) o centrales de datos (Banikazemi et al., 2013). Estudios como el presentado en (Kobo et al., 2017) han identificado al paradigma de SDN como potencial solución a la mayoría de las limitaciones de las WSN, con el objetivo de llevar una mejor administración de la red. En conclusión el paradigma SDWSN busca aliviar la mayoría de los retos de las WSN en búsqueda de eficiencia en la red.

## 1.2 Planteamiento del problema

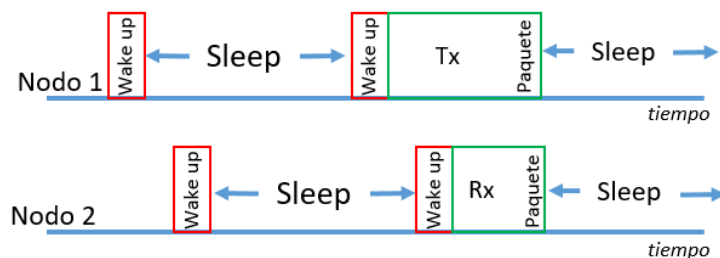
Las WSN presentan retos relacionados con limitaciones en recursos del hardware, como el procesamiento, memoria, energía y capacidades de comunicación (De Gante et al., 2014). Dentro de los principales retos se encuentra el bajo consumo de energía requerido para que los nodos tengan un tiempo de vida en la red prolongado, debido a que los nodos son dispositivos de tamaño reducido, miniaturizados, que están limitados a emplear baterías con baja potencia, en algunos casos irremplazables. De manera adicional, en aplicaciones en las que los sensores se despliegan en lugares con difícil acceso (Son et al., 2006; Hasenfratz et al., 2014; Kim and Cho, 2017) se complica el remplazo de las baterías. Por lo tanto, los sistemas, procesos y protocolos de comunicación de las redes inalámbricas de sensores (WSNs) deben minimizar el consumo de energía para prolongar el tiempo de vida de los nodos en la red (Akyildiz et al., 2002).

Una solución para poder prolongar el tiempo de vida del nodo en la red es utilizar técnicas de cosechado de energía, esto es extraer energía del ambiente, por ejemplo al añadir un subsistema de celdas solares o transductores (Xue et al., 2009). Sin embargo este tipo de soluciones limitan el despliegue de las WSN a ciertas áreas en específico solamente (Akbari, 2014).

Los módulos a los cuales se les atribuye un consumo de energía en el nodo sensor son: monitorización, comunicación y procesamiento de datos. El modulo que más consumo de energía presenta es el de radio de comunicación tanto para transmisión como recepción (Shnayder et al., 2004; Jurdak et al., 2007). Una

prueba de este consumo se presenta en el estado de *idle listening*, esto es, el radio se mantendrá encendido a la espera de escuchar una transmisión para recibir algún paquete, teniendo el caso que el paquete nunca llegue a ser recibido, este estado se observa comúnmente en la mayoría de las aplicaciones de redes de sensores (Ye et al., 2004). El análisis llevado a cabo al nodo sensor Micaz (Crossbow, s/f) muestra que el consumo de corriente para transmisión ronda entre 3.7 mA y 21.5 mA, dependiendo de la potencia de transmisión. Sin embargo, el consumo de corriente para el modo escucha *listening* es de 7 mA sin importar la actividad de transmisión, costando la mayor parte de la energía de la batería (Zhou y Guan, 2009).

Para aliviar este problema de consumo de energía, se recomienda aplicar técnicas de ciclo de trabajo al radio transceptor. El ciclo de trabajo es una técnica donde el radio periódicamente pasa la mayor parte del ciclo en un estado de sueño en baja energía (alrededor del 99% del periodo), despertando para escuchar el medio por un breve tiempo. Por lo tanto la operación del ciclo de trabajo resulta en que un nodo sigue un ciclo de duerme-despierta-muestrea-y-comunica, como se observa en la figura 4. Así conforme el ciclo de trabajo aumenta, los nodos pueden pasar más tiempo dormidos resultando en un método eficiente en reducir la energía disipada en las WSN (Lin et al., 2004).



**Figura 4** Agenda de ciclo de trabajo del radio transceptor en nodos sensores.

Sin embargo, reducir el ciclo de trabajo decrementa el tiempo de actividad del nodo, causando que la conectividad entre la red se vuelva intermitente. Además, ocasiona un retraso de comunicación, principalmente en redes con bajo ciclo de trabajo, debido a la latencia que el nodo transmisor espere a que el nodo receptor despierte.

Como se ha mencionado anteriormente, una propuesta reciente es utilizar las redes definidas por software en las WSN (SDWSN). Este paradigma se enfoca en que la mayor parte de los procesos, p.ej. enrutamiento,

se lleven a cabo en un controlador central. Esto permite que la mayoría de los nodos tengan capacidades de cómputo reducidas, requiriendo menor consumo de energía y con un menor costo.

Para desplegar el paradigma de SDWSN es necesario el intercambio de paquetes de control entre los nodos y el controlador para construir las tablas de flujo en el controlador que posteriormente serán enviadas a los nodos sensores. En (Costanzo et al., 2012) se mostró que es posible aplicar el paradigma de redes definidas por software en WSN. Sin embargo, no se consideran técnicas de ciclo de trabajo, las cuales son sumamente importantes considerando la cantidad de nodos sensores y aplicaciones que se visualizan en el contexto de IoT. En este trabajo de investigación se evaluará el proceso de formación de una red WSN utilizando el paradigma de redes definidas por software cuando se aplican técnicas de ciclo de trabajo al transceptor de radio tomando en cuenta características de las topologías encontradas en las aplicaciones de las WSN.

### **1.3 Justificación**

En la literatura se reportan esfuerzos por detallar los métodos para realizar la inserción del paradigma de SDN a las WSN, la mayoría de ellos de forma teórica proponen un modelo en el cual basarse para realizarlo, solo unos cuantos trabajos llevan la tarea de llevar a la práctica la evaluación tomando en cuenta los requerimientos que las WSN presentan. Sin embargo, una pequeña muestra de estos trabajos se ha tomado el menester de evaluar el desempeño que esta inserción presenta a las WSN. Estos esfuerzos han traído resultados significativos en métricas de desempeño como la pérdida de paquetes, caudal eficaz, *overhead*, tiempo de envío de paquetes y retraso de envío de paquetes. Haciendo que estos resultados sean comparados con los obtenidos en otras plataformas de comunicación o algoritmos de enrutamiento como Zigbee y RPL.

El desarrollo de adaptar el paradigma de SDN a las WSN debe ser tal que trate de tomar en cuenta todas las características de las WSN para presentar una solución holística, esto ha llevado a promover que la inserción proporcione soporte a utilizar técnicas de ciclo de trabajo para reducir el consumo de energía del radio. Sin embargo a pesar de los esfuerzos realizados, no se ha hecho un análisis que evalúe el impacto al implementar las técnicas de ciclo de trabajo a las SDWSN ni la elección de configuraciones de estas técnicas para lograr el óptimo desempeño de la red.

Además, el proceso de formación de la red SDWSN requiere de intercambio de mensajes de control que son configurados con ciertas características. Las configuraciones de periodicidad de envío de estos mensajes de control tampoco han sido evaluadas para determinar cuál es la configuración óptima para formar una red de este tipo, trivializando el uso de estas configuraciones sin la cuestión de observar cómo afecta el desempeño de la red en los trabajos presentados. Es notorio de que el tiempo que se lleva trabajando en las soluciones SDWSN es relativamente corto, y por ello las redes SDWSN no tienen un estándar definido como lo tiene el enrutamiento RPL; donde se han llevado a cabo un número significativo de pruebas para establecer las configuraciones adecuadas para que el enrutamiento para la formación de la red pueda ser óptimo.

Es por ello, la motivación de este trabajo de tesis es agregar una evaluación que sea de apoyo en el desarrollo de las consideraciones adecuadas para la completa formación de las SDWSN.

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Evaluar el impacto que tiene el ciclo de trabajo del radio en las redes de sensores inalámbricas definidas por software dentro del protocolo de descubrimiento de topología. Se medirá el impacto considerando métricas de desempeño tomando en cuenta las características de las topologías habituales en las WSN.

### **1.4.2. Objetivos específicos**

- Estudiar el estado del arte de la formación de la red en redes definidas por software en las WSN y comparar con la formación de la red con un enrutamiento de facto de las WSN.
- Seleccionar una implementación de SDN para WSN que permita generar el modelo de simulación.
- Generar un modelo de simulación validado que permita evaluar las interacciones de los nodos sensores en el contexto de IoT.

- Realizar la comparación de la evaluación con un algoritmo de enrutamiento validado.
- Definir las métricas y los escenarios de simulación para la evaluación de formación de la red.
- A partir de las evaluaciones realizadas, generar conclusiones sobre el desempeño de la formación de la red SDWSN al aplicar técnicas de ciclo de trabajo y determinar si la solución tiene soporte para estas técnicas.

## 1.5 Metodología

### 1. Investigar el estado del arte de las redes definidas por software aplicadas a las WSN

Se realizó una investigación de las SDN y la inserción de este paradigma a las redes inalámbricas de sensores, mostrando las diferentes aportaciones hechas; con el objetivo de analizarlas y observar sus características.

### 2. Seleccionar un enfoque SDWSN para realizar evaluación

Se determinó el uso de un enfoque de *Software Defined Networking solution for Wireless Sensor Networks* (SDN-WISE) para la evaluación del sistema, detallando las características de la formación de la red y seleccionando los módulos necesarios para la evaluación

### 3. Definir escenarios y métricas de evaluación de los algoritmos de enrutamiento sobre redes con ciclo de trabajo

Se definieron los escenarios que serán de utilidad considerando ciertas características que las topologías presentan, al igual se definieron las métricas necesarias para la evaluación del proceso de descubrimiento de topología llevado por la red SDWSN y la comparación con el proceso de formación de la red con RPL; con el objetivo de llevar una evaluación completa desde diferentes aspectos.

### 4. Utilizar el modelador de redes para generar un modelo de red que permita simular la interacción de los nodos sensores.



Se decidió utilizar el modelador de redes Cooja del sistema operativo de Contiki, debido a las características que presenta, para determinar la interacción de los nodos para formar la red y añadir las técnicas de ciclo de trabajo

#### **5. Validar el modelo de simulación**

Es vital utilizar herramientas que tengan la confiabilidad de que los resultados obtenidos sean correctos mediante la certeza de que su funcionamiento no esté comprometido. Por ello, se realizó la validación del simulador de ContikiOs-Cooja mediante el análisis de la capa MAC del estándar IEEE 802.15.4 utilizado en la WSN; tomando como fundamento y método de comparación un esquema de simulador validado mediante un modelo analítico y un esquema de implementación de hardware. De esta forma los resultados se pueden proyectar a escenarios más complejos obteniendo datos fidedignos.

#### **6. Evaluar el desempeño de la formación de la red.**

Se realizó un exhaustivo número de simulaciones para analizar cada uno de los escenarios propuestos con sus respectivas configuraciones de periodicidad de generación de paquetes y configuración de ciclo de trabajo. Dicho análisis convergió en una discusión comparativa entre tecnologías y propuesta de mecanismos necesarios.

#### **7. Elaborar conclusiones y documentar resultados obtenidos.**

Una vez evaluado el desempeño de la formación de la red SDWSN, se generaron conclusiones de los resultados mostrados, comparando la habilidad mostrada en la red SDWSN para formar la red cuando se presenta técnicas de ciclo de trabajo y sobre que escenarios es factible implementarla.

### **1.6 Organización de la tesis**

El capítulo 2 hace una breve introducción a las redes a evaluar. Se describen aspectos técnicos de RPL y se hace una breve introducción a las redes definidas por software y la interacción llevada a cabo para implementarse en las WSN, en dicha sección se muestra un listado de enfoques que realizan la inserción de forma teórica y práctica de SDN a las WSN que llevaron a elegir el enfoque a ser utilizado.

El capítulo 3 describe las herramientas utilizadas para realizar esta evaluación. En dicho capítulo se describe la pila de protocolos del sistema operativo Contiki utilizado para operar en los nodos sensores y las herramientas de simulación para la obtención de las métricas como el consumo de energía.

El capítulo 4 relata las configuraciones requeridas de las redes a evaluar, estos son los paquetes de control utilizados por RPL y las configuraciones de periodicidad de los paquetes de control de la red SDWSN. Dicho capítulo también describe los parámetros de desempeño utilizados para realizar la evaluación, así como una breve descripción de las topologías utilizadas para el desarrollo del análisis. Por último en el capítulo se hace la descripción de la validación del simulador Cooja y la metodología de la evaluación; en dicha sección se hace mención de la necesidad de un mecanismo externo por parte de la red SDWSN para la distribución de rutas al implementar técnicas de ciclo de trabajo.

El capítulo 5 consta del análisis realizado por las series de simulaciones llevadas a cabo para determinar el impacto que genera implementar técnicas de ciclo de trabajo a la red SDWSN y la comparación del desempeño de formación de la red con RPL.

Para finalizar el trabajo realizado en esta tesis, el capítulo 6 presenta las conclusiones obtenidas resultantes de la evaluación. De igual forma se enlista el trabajo a futuro a realizarse a continuidad.

## Capítulo 2. Procesos de formación de la WSN

---

### 2.1 Introducción

Las redes inalámbricas de sensores (WSNs por sus siglas en inglés) consisten en nodos interconectados, los cuales generalmente tienen recursos limitados como: poder de procesamiento, memoria y energía (Ndiaye et al., 2017). Por lo tanto, comúnmente se asume que deben ser de bajo consumo de energía y que tienen enlaces de comunicación inestables y con pérdidas (LLNs por sus siglas en inglés). Comúnmente, las WSN pueden estar formadas por decenas de nodos, por ejemplo, en las aplicaciones de salud (J. Ko et al., 2010). En el estado del arte existen diversos protocolos de enrutamiento adaptados para las capacidades de este tipo de redes (Shabbir y Rizwan, 2017). Uno de los más relevantes es el protocolo de enrutamiento RPL (*Routing Protocol for Low Power and Lossy Networks* por sus siglas en inglés), propuesto por el grupo de trabajo ROLL (*Routing Over Low power and Lossy networks* por sus siglas en inglés) de la IETF. De hecho, el protocolo RPL es considerado como estándar de facto para este tipo de redes (IETF, 2012).

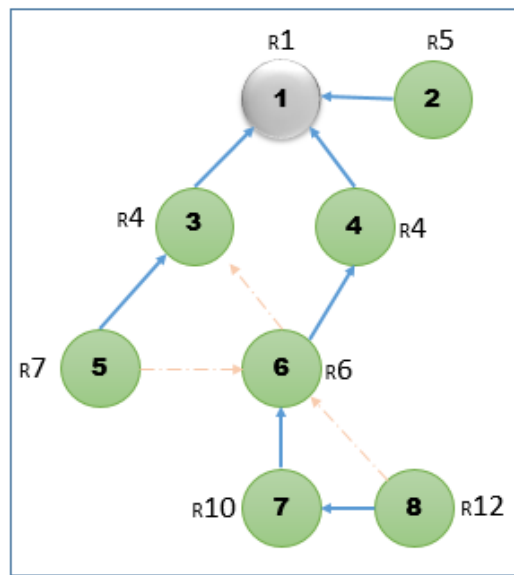
### 2.2 Redes WSN con enrutamiento RPL

RPL es un protocolo de enrutamiento proactivo para redes con direccionamiento IPv6, cuya base es ser un protocolo vector-distancia donde se calcula la dirección y la distancia de cualquier enlace de la red y la información de los cambios de la topología es informada por medio de un enrutador (Ivana, 2017). Los algoritmos proactivos generan las rutas durante el establecimiento de la red aún si no son requeridas por algún nodo. En estos protocolos periódicamente se intercambian mensajes de control para compartir su información con los nodos que están dentro de su área de cobertura (vecinos a un salto). Además, cada nodo comparte, comúnmente de forma periódica, mensajes con la información de su vecindario ( todos sus vecinos a 1 salto) (Jacquet et al., 2001). RPL forma una topología de tipo árbol llamada DAG (*Direct Acyclic Graph*), la cual es un grafo dirigido con la propiedad de que el grafo esté libre de ciclos. En la DAG todos los nodos tienen una ruta hacia un nodo raíz, formando así una DODAG (*Destination Oriented DAG*).

El nodo raíz de la DODAG es llamado DODAG root, cuya característica es no estar dirigido a otro nodo. Todas las DAG tienen al menos un nodo raíz que puede actuar como un enrutador de frontera para la

DODAG y pueden redistribuir las rutas hacia otros protocolos de enrutamiento. En general, hay tres tipos de nodos en RPL. Los primeros como se menciona en el párrafo anterior son los nodos raíz los cuales también se le conoce como *gateways* o puertas de enlace. El segundo tipo son los enrutadores, dichos nodos pueden anunciar la información de la topología de la red a sus vecinos. Por último, los nodos hoja, los cuales son nodos que no envían ningún mensaje de control y solo tienen la habilidad de unirse a la DODAG.

Una visualización de una red en RPL está mostrada en la figura 5. En dicha figura el nodo resumidero está marcado como color plateado, mientras que los demás nodos pertenecientes a la DODAG se encuentran en color verde.



**Figura 5** Representación de una DODAG en RPL.

Cada nodo que forma una red RPL tiene un nodo vecino llamado padre preferido y actúa como una puerta de acceso a otro nodo con dirección hacia el DODAG root. Una vez que se ha elegido un nodo padre, dicho nodo se convierte en un nodo hijo y almacena una tabla de posibles padres a elegir si el nodo padre seleccionado llega a ser comprometido (Ali, 2012). Los nodos padres están representados en la figura 5 como los nodos dirigidos por las flechas azules, mientras que las flechas de color anaranjado indican los posibles padres a seleccionar.

La formación de la DODAG se realiza mediante una función objetivo (OF), cuya meta es definir la manera en que la métrica de ruteo va a ser calculada. La OF especifica como algunas restricciones y funciones del enrutamiento serán tomadas en cuenta para construir la topología (Tsvetkov, 2011). Además, las funciones objetivo son usadas para calcular el rango de un nodo en la red, esto es la distancia entre el nodo y el DODAG root. En RPL el rango es el valor que define la posición relativa de los nodos con respecto a sus vecinos y al DODAG root, el valor del rango es considerado importante en la construcción y mantenimiento de la DODAG por ello cada nodo mantiene su propio rango. El valor del rango se incrementa si los nodos se posicionan lejos del nodo raíz y está basado en la relación entre padre e hijo, tal que el hijo debe tener un rango estrictamente mayor que el rango de todos sus padres de la DODAG (IEFT, 2012). El valor que adopta el rango lo define el administrador de la red y puede estar determinado con respecto al conteo de saltos o calculado como una función de la métrica de enrutamiento, como aspectos de consumo de energía o el número esperado de transmisiones (ETX). En la figura 5 dicho valor está representado a un lado de cada nodo denotado por la letra R.

Una vez que la topología de la red está completa, dicha DODAG se convierte en una instancia de la red en RPL, la cual puede estar formada por una o varias DODAGs que comparten un mismo identificador.

### **2.2.1 Mensajes de control para la formación de una red en RPL**

Para realizar la formación de la topología e intercambiar información entre los nodos, la especificación de RPL define cuatro tipos de mensajes de control definidos por el estándar de RPL como mensajes de información ICMPv6: DIO, DAO, DIS, DAO-ACK (IANA, 2011). Los cuales se describen en la tabla 1.

**Tabla 1** Descripción de mensajes de control de RPL.

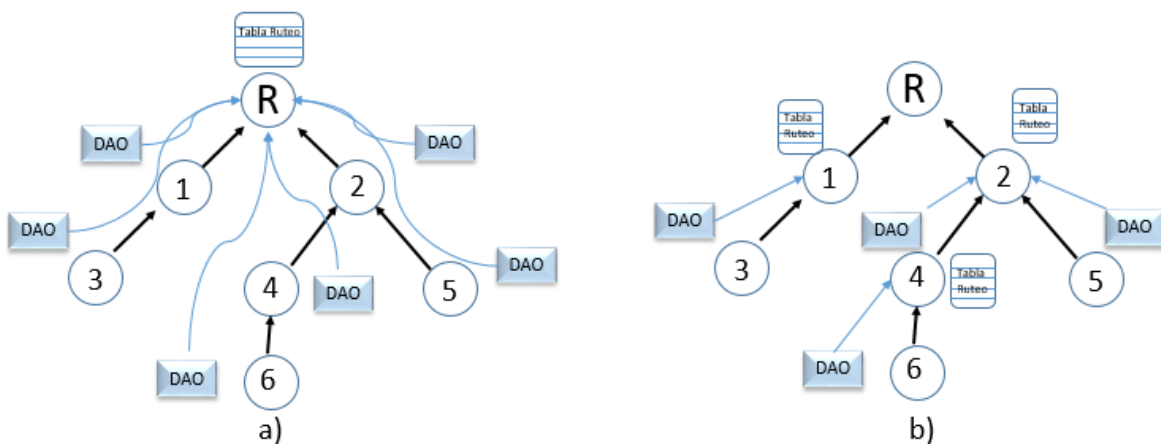
<b>Mensaje de control</b>	<b>Descripción</b>
<i>DIO</i>	<p>DODAG Information Object es la principal fuente de información de control, ya que almacena información como el valor actual del rango de los nodos, el valor de la instancia de la DODAG actual, la dirección IPv6 del nodo root, entre otros valores.</p> <p>RPL adapta la tasa de envío de mensajes de control DIO, mediante un algoritmo llamado Trickle (Internet Engineering Task Force (IETF), 2011) , donde se definen dos valores; uno para configurar el intervalo mínimo de envío, conocido como <math>I_{min}</math> y un segundo valor para definir el tiempo máximo de envío denotado por <math>I_{max}</math>. El algoritmo presenta un contador que determina la estabilidad de la red y que al expirar, dobla el valor de <math>I_{min}</math> sucesivamente hasta llegar al valor máximo <math>I_{max}</math>.</p>
<i>DAO</i>	<p>Destination Advertisement Object es utilizado para formar las rutas que serán utilizadas para enviar paquetes de datos a los nodos, es usado para propagar la información de la ruta hasta su destino a lo largo de la DODAG, así cada ruta es propagada por el nodo padre preferido para formar la red.</p>
<i>DIS</i>	<p>DODAG Information Solicitation, este paquete hace posible a un nodo que requiera ingresar a la DODAG solicitar un mensaje DIO a un nodo vecino alcanzable</p>
<i>DAO-ACK</i>	<p>Reconocimiento del DAO es enviado en respuesta por un nodo que recibe un paquete DAO permitiendo que este nodo acceda a la DODAG</p>

En RPL existen dos formas de transmitir la información, ya sea hacia arriba (*routing upward*) o hacia abajo (*routing downward*) del árbol formado una vez que se forma la DODAG, cada nodo tiene una tabla de enrutamiento hacia cada nodo de la red. Cuando llega un paquete que debe ser transmitido hacia arriba o abajo, el nodo consulta su tabla de enrutamiento para elegir el siguiente salto. Es importante mencionar

que en una red RPL, las tablas de enrutamiento tienen una entrada para cada nodo dentro de la red. Esto significa que los nodos que están cerca del nodo raíz tienen tablas de enrutamiento más grandes, lo que puede causar un problema de escalabilidad.

La especificación de RPL define dos modos de operación para el soporte del enrutamiento P2MP. Primero, el modo de *non-storing* o no almacenamiento, el cual hace uso de una fuente de enrutamiento. En este modo cada nodo tiene que propagar su lista de padres hacia el nodo raíz, posteriormente el nodo raíz calcula el camino a los destinos. Si un nodo desea transmitir información a otro nodo en la red, primero deberá enviar el paquete de información al nodo root y posteriormente el paquete será retransmitido al nodo destino. El segundo modo, es el de *storing* o almacenamiento, el cual está basado en estados. En este modo, cada participante de la red que no sea raíz u hoja tiene que mantener una tabla de enrutamiento para posibles destinos, haciendo que no sea necesario enviar la información de cada nodo hacia el nodo raíz, en lugar dicha información es enviada mediante paquetes unicast a cada padre que forma su tabla de enrutamiento hacia abajo. Cada instancia de RPL debe ser solamente de un modo, ya sea *storing* o *no storing* (Tsvetkov, 2011).

Para que el tráfico P2MP ocurra y se propague la información del enrutamiento, se hace uso de los mensajes DAO, los cuales pueden ser solamente enviados después de la formación de la topología por el intercambio de los mensajes de control DIO (Ali, 2012). Estos mensajes DAO se intercambian entre sí, tanto en el modo *storing* y *no storing*, en donde el tiempo de generación depende de la implementación de la DODAG a crear. La figura 6 muestra un ejemplo de los modos *routing* y *no routing*.



**Figura 6** Modos de operación en enrutamiento RPL a) Modo no storing de RPL b) Modo storing de RPL.

### 2.2.2 Formación de una red RPL

Cuando se usa RPL, se considera que la red está formada cuando se ha construido la DODAG. Con el objetivo de explicar mejor el proceso de formación de una DODAG (figura 7a), en la figura 7b) se presenta un ejemplo de formación de una topología de red multisalto conformada por 3 nodos sensores y un resumidero A, en un primer salto se encuentran los nodos 1.1 y 2.1, en el segundo salto el nodo 1.2. El administrador de la red configura uno o más nodos como nodos raíces de la DODAG, como es el nodo A en la figura 3. Estos nodos raíces comienzan a transmitir mensajes DIO de forma *multicast*; un nodo que no sea raíz puede solicitar un mensaje DIO en algún momento del tiempo a través de los mensajes DIS e inmediatamente la raíz de la DODAG enviará un mensaje DIO.

- 1) Los nodos cercanos recibirán el mensaje DIO de parte del nodo raíz y lo procesarán calculando su rango para seleccionar a la raíz como nodo padre, como ocurre con el nodo 1.1 y 2.1.
- 2) Los nodos que hayan recibido el mensaje DIO del resumidero ahora enviarán mensajes DIO también de forma *multicast*, de tal manera que los nodos restantes (el nodo 1.2) los seleccionen como sus padres.

Si un nodo recibe varios mensajes DIO de dos o más padres, este decidirá en base a la función objetivo al nodo padre preferido. Este proceso continuará hasta que los mensajes DIO alcancen a los nodos hoja.

Para fines de esta tesis se utilizará el modo storing de RPL debido a que es el modo por defecto que implementa Contiki OS.

- 3) Para concluir la formación de la red RPL, los nodos hijos enviarán a su nodo padre preferido un mensaje DAO, el cual decidirá ingresarlo o no en la tabla de enrutamiento respondiendo con un mensaje DAO ACK.



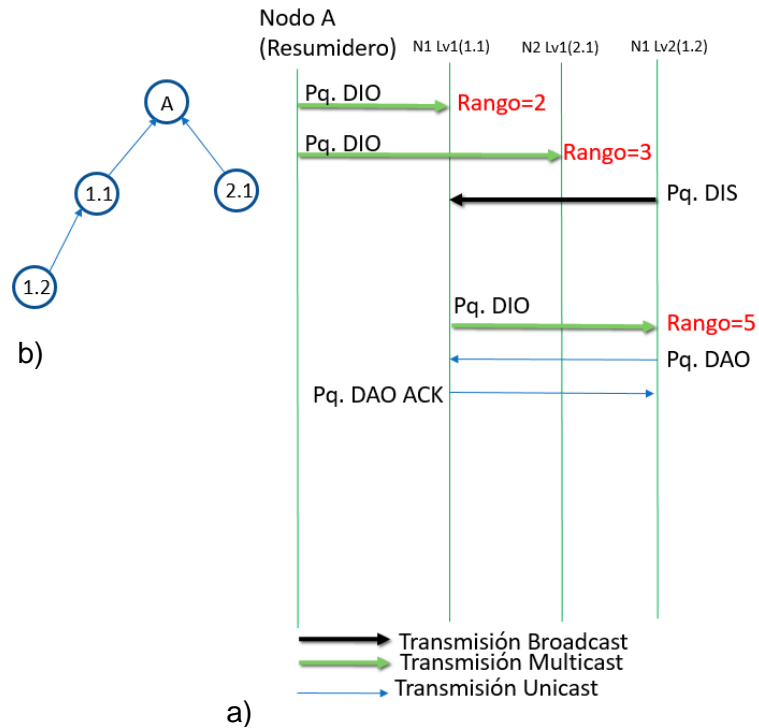


Figura 7 a) Proceso de formación de la DODAG en RPL b) DODAG multi salto.

## 2.3 Redes SDWSN

En el contexto de IoT, se espera que las WSN estén formadas por una gran cantidad de nodos, por lo que surgen varios desafíos como la administración de la red (Kobo et al., 2017) y otros aspectos inherentes que hacen que no se alcance la efectividad óptima, tales como:

- **Aspectos de energía:** los nodos sensores están restringidos en energía por el uso de baterías, a manera de que la topología de la red crece se vuelve complicado reponer dicha fuente de energía.
- **Aspectos de enrutamiento:** durante el envío de información entre nodos al resumidero pueden existir pérdidas de información, debido a la falta de sincronización de los nodos para escucharse.
- **Aspectos de seguridad:** los requerimientos de seguridad fundamentales deben ser cumplidos como autenticación de la seguridad, confidencialidad de datos, integridad en los datos, disponibilidad, y redundancia.

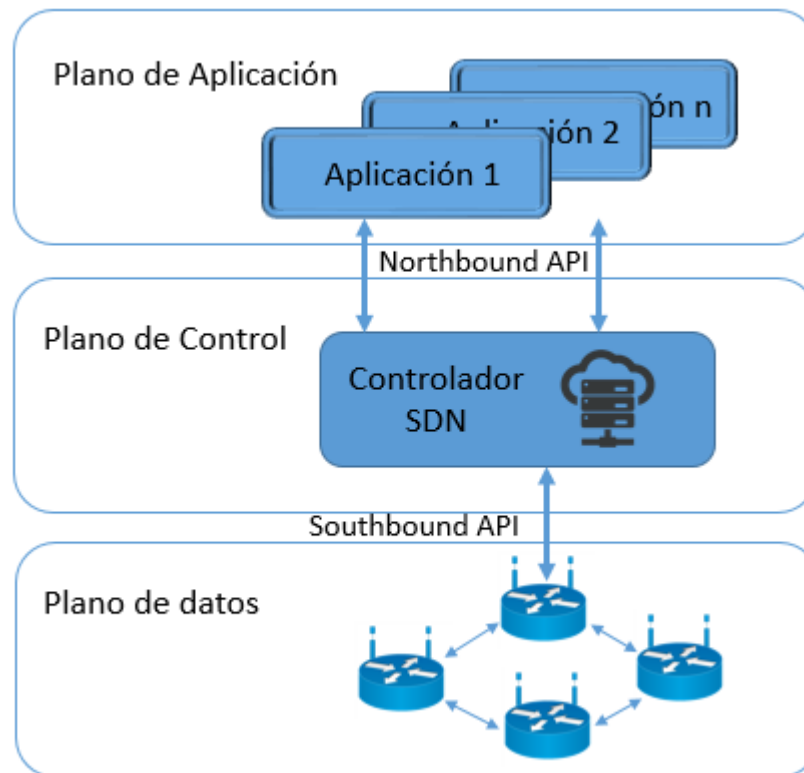
- **Aspectos de configuración:** la configuración manual de cualquier dispositivo de red es retardador especialmente cuando la red crece. Los nodos sensores necesitan responder rápido a algún cambio en la red y por lo tanto se tiene la necesidad de una configuración de administración dinámica.

El paradigma de las redes definidas por software (SDN, por sus siglas en inglés) en las WSN, asume que comúnmente en la mayoría de las WSNs los nodos sensores son estacionarios y las relaciones de conexión entre los nodos de sensores raramente cambian. Un proceso de enrutamiento de descubrimiento de topología frecuente como en los sistemas distribuidos se vuelve menos necesario. Además, el resumidero en las WSNs generalmente tiene un mayor poder de cómputo y mayores reservas de energía.

El paradigma de SDN en las WSN (SDWSN) busca aliviar los desafíos más relevantes mencionados anteriormente y fomentar la eficiencia y sostenibilidad de la WSN mediante las prestaciones que SDN ofrece (The Open Networking Foundation, 2017). Considerar las características y retos mencionados hace que aplicar un algoritmo de ruteo de control centralizado sea conveniente.

### **2.3.1 Redes SDN**

Las redes definidas por software son un paradigma de redes cuyo objetivo es simplificar la administración y configuración de la red, mejorando la eficiencia de la red mediante abstracciones de alto nivel (Kreutz et al., 2015). A diferencia de las redes tradicionales en donde el plano de control y el plano de datos están acoplados dentro del mismo hardware, el paradigma busca desacoplar la inteligencia de la red (el plano de control) del motor de envío de paquetes (el plano de datos).



**Figura 8** Arquitectura de redes definidas por software.

Esta separación provee al nodo que mantiene el plano de control una visión global de la red, para tomar decisiones sobre ella de forma centralizada. La figura 8 presenta una visualización de la separación de planos en SDN. El plano de control mantiene la inteligencia de tomar decisiones y es responsable del control y administración en base a las solicitudes que la aplicación requiera.

Al hacer la separación de planos, los dispositivos enrutador que contenían ambos planos se vuelven solo conmutadores de envío de información que obedecen las decisiones de enrutamiento hechas por el controlador mediante reglas que les enseñan a cómo manejar los paquetes recibidos. Los planos separados están interconectados por varias APIs referidos como interfaces *northbound* (la comunicación entre el plano de control y aplicación) y la interfaz *southbound* (la comunicación entre el plano de control y el plano de datos) (Jammal et al., 2014).

### 2.3.2 Redes SDN sobre WSN

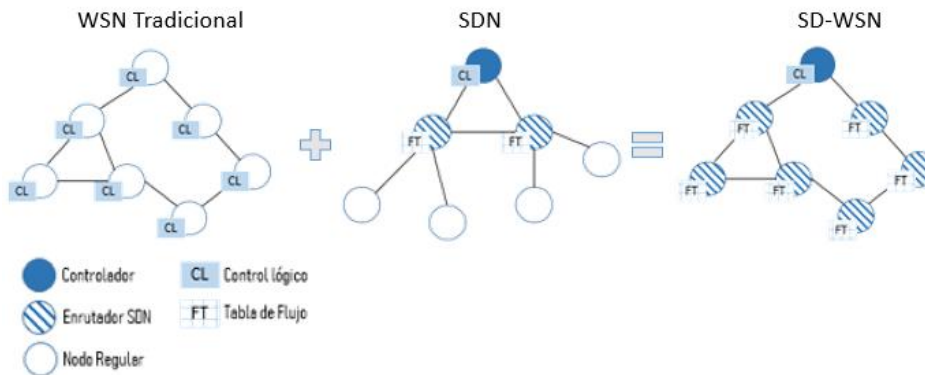
Las ventajas que presentan las WSNs son diversas por ejemplo el monitoreo de variables físicas en entornos remotos o recolección de información. Sin embargo, para poder aprovechar estas ventajas aún se deben resolver retos importantes como: aspectos de energía, administración y configuración de la red, escalabilidad, comunicación, entre otros. Aunque se ha realizado trabajo para abordar estos retos, aún no hay una solución holística que aborde todos los retos existentes.

El enfoque que brinda SDN al ser implementado sobre las WSN promete abordar a varios de los retos que presentan dicho tipo de redes, por ejemplo (Kobo et al., 2017):

- **Energía:** el paradigma SDN busca que el desacoplamiento de los planos de control y datos libere de gran parte del cómputo realizado a los nodos, ej. el cálculo de rutas para diseminar información. La mayoría de estas funciones ahora residirán en el controlador, que tiene suficientes recursos de energía. Esto busca ahorrar una cantidad considerable de energía y prolongar potencialmente la vida de los nodos en la red.
- **Administración y configuración de la red:** la administración, reconfiguración y mantenimiento de la red en las WSN es compleja. SDN busca simplificar la administración de la red centralizando el control en el controlado lógico.
- **Seguridad:** La centralización de la administración de la seguridad simplifica la implementación de los mecanismos de seguridad. Un fundamento importante de las SDN en las WSN es que el nodo sensor se vuelve un “elemento tonto” el cual solo entiende a los mensajes o comandos del controlador. Esto dificulta la operación de ser usado como un nodo malicioso.

Así, el paradigma de SDWSN se enfoca en hacer más eficiente la operación y despliegue de las WSN, de tal forma en diferentes trabajos encontrados en la literatura (Mahmud y Rahmani, 2011; Costanzo et al., 2012; Luo et al., 2012a; De Gante et al., 2014), se realiza una adaptación del paradigma de SDN para ser desplegado en WSN como prueba de concepto. Cabe destacar que el desarrollo de las SDWSN está todavía en su etapa inicial y aunque existan diferentes implementaciones de arquitecturas, todas ellas conforman los fundamentos de SDN: el desacoplamiento.

El enfoque de SDN para las redes inalámbricas de sensores implica abstraer diferentes funcionalidades y reorganizarlas a lo largo de tres planos lógicos del modelo SDN: aplicación, control y datos. Una imagen de la inserción de SDN en las WSN es observada en la figura 9 donde se muestra la combinación de ambas redes (Alves et al., 2018).

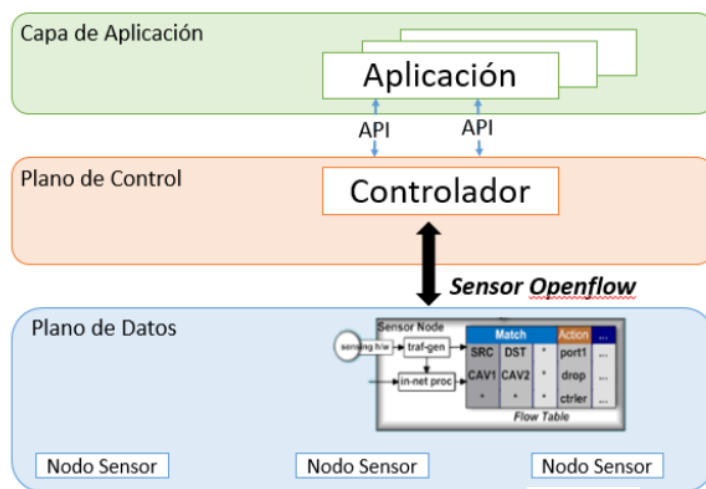


**Figura 9** Topología de SDN sobre topología WSN: Formación SDWSN.

Uno de los estudios pioneros para habilitar SDN en las WSN es el presentado en (Dely et al., 2011) en donde proponen integrar el paradigma mediante la implementación del protocolo de facto de SDN, *Open Flow* (McKeown et al., 2008), en una red inalámbrica de sensores de tipo malla. La arquitectura que ellos proponen está sugerida en dos partes llamada, 1) redes principales y 2) estaciones, en donde la red principal está compuesta por el controlador y un servidor de control y monitoreo. El trabajo presentado por (Dely et al., 2011) asume que los nodos pertenecientes a la WSN son dispositivos de cómputo normal, por lo tanto no se ajusta a los dispositivos de red que están limitados en recursos.

Posterior a ese año (Luo et al., 2012) desarrolló una investigación similar al trabajo propuesto por (Dely et al., 2011), el cual propone un marco de arquitectura para las redes definidas por software en las WSN. Esta arquitectura muestra una clara separación entre el plano de control introducido por el protocolo *Sensor Openflow*, un protocolo de comunicación personalizado e implementado entre los dos planos y diseñado específicamente para adaptar *Openflow* y hacerlo compatible con las propiedades inherentes de las WSN. El enfoque que *Sensor Openflow* brinda, es de proponer un concepto a la API *southbound* siendo este crucial para el envío de instrucciones y recepción de solicitudes entre el plano de datos y control como se observa la figura 10. La idea está en resolver los problemas inherentes en las WSN como la subutilización

de recursos, la rigidez a los cambios de política y la dificultad para administrar la red, volviendo la red tradicional WSN versátil, flexible y sencilla de administrar.

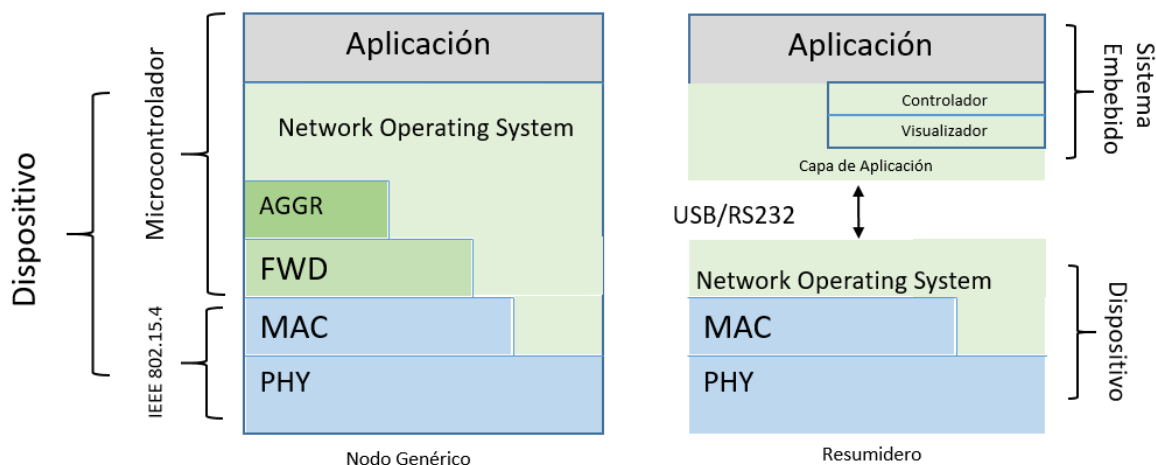


**Figura 10** Arquitectura de SDN propuesta por SensorOpenflow (Luo et al, 2012).

Sin embargo, a pesar de dar una adaptación precisa para implementar un protocolo de redes alámbricas a redes inalámbricas, se observan algunos detalles que lo vuelven difícil de implementar en un entorno real, tal que no explican como la arquitectura propuesta y el protocolo pueden ser implementados y no brindan una explicación del plano de control ni una evaluación del desempeño del sistema, dejando la aplicación meramente teórica.

Costanzo et al (2012) propone un nuevo concepto nombrado SDWN (Software Defined Wireless Networks), siendo un enfoque diferente a lo que *Sensor Openflow* propone. SDWN propone una API para la interfaz de comunicación entre un nodo sensor y el controlador. SDWN introduce por primera vez el concepto de *Flow table* (tabla de flujo). Las *Flow tables* se utilizan para almacenar las reglas de flujo que los nodos deben obedecer con base en ciertos criterios impuestos por la capa de aplicación mediante el controlador. La principal contribución que ofrece el trabajo de (Costanzo et al., 2012) es proponer la arquitectura SDWN considerando las características de una WSN basada en el estándar de comunicación IEEE 802.15.4, abriendo así la puerta a escenarios de redes inalámbricas de área personal con baja tasa de datos. La adaptación propuesta en (Costanzo et al., 2012) brinda soporte para el ciclo de trabajo, procesamiento dentro de la red y agregación de datos para reducir el consumo de energía. En comparación a los trabajos previamente mostrados, SDWN ofrece una definición flexible y de alto nivel a las reglas de

enrutamiento. La arquitectura propuesta se puede observar en la figura 11, la cual consiste en dos componentes: un nodo genérico y un nodo resumidero. El nodo genérico actúa como retransmisor utilizando las reglas de la *Flow table* siguiendo los mecanismos de la capa física y de acceso al medio de IEEE 802.15.4. El nodo resumidero es el puente entre los nodos genéricos y el controlador, por lo que está conectado a un sistema que se puede considerar sin limitaciones de procesamiento o energía, en donde reside el controlador que se encarga del plano de control que define las reglas para las *Flow tables*.



**Figura 11** Arquitectura de SDN propuesta por SDWN (Costanzo et al., 2012).

Los autores en (Costanzo et al., 2012) proveen un diseño completo y detalles a implementar para recolectar la información de la topología de la red, especificando cada uno de los diferentes formatos de paquetes, el manejo de las reglas de flujo y las acciones a implementar. Podría considerarse a este protocolo como uno de los primeros intentos para desarrollar una pila de protocolos personalizada, a ser usada en dispositivos que están limitados en recursos. El trabajo de (Costanzo et al., 2012) ha sido base para otras propuestas para redes WSN definidas por software como las presentadas en (Anadiotis et al., 2015; Galluccio et al., 2015; Fotouhi et al., 2016; Hieu et al., 2018).

SDWN es una solución muy completa que muestra resultados prometedores en el uso del paradigma de redes definidas por software en WSNs. Sin embargo, no evalúa el desempeño de características clave de las redes definidas por software como el de una administración de la red flexible. Por lo tanto, todavía es necesario desarrollar investigación que permita determinar si es posible desplegar SDN sobre WSN. En la

siguiente tabla se enlistan intentos que se han realizado en base a implementaciones y evaluaciones hechas sobre adaptar SDN sobre WSN.

**Tabla 2** Comparativa de soluciones SDN sobre WSN.

<b>Propuesta</b>	<b>Características</b>	<b>Implementación</b>	<b>Disponibilidad de aplicación</b>	<b>Desempeño de evaluación</b>
<i>FLOW-Sensor</i> (Mahmud et al., 2011)	Se enfoca en aspectos de confiabilidad	Simulación	No	Tiempo de respuesta y número de paquetes generados
<i>Sensor Openflow</i> (Luo et al., 2012b)	Propone concepto de SDWSN	No	No	No
<i>SDWN</i> (Costanzo et al., 2012)	Propone un concepto de SDWSN	No	No	No
<i>SDWSN</i> (De Gante et al., 2014)	Arquitectura propuesta para una administración inteligente de la WSN	No	No	No
(Jacobsson y Orfanidis, 2015)	Propone una arquitectura de SDN para WSN	No	No	No
<i>SDWN ONOS</i> (Anadiotis et al., 2015)	Proveen un marco SDN-IoT con controlador	Simulación Cooja y mininet	Abierta a descargar	No
<i>TinySDN</i> (Trevizan et al., 2015)	Marco de arquitectura para múltiples controladores	Simulación Cooja	Abierta	Tiempo de respuesta, aspectos de memoria
<i>SDWSN cognitivo</i> (Huang et al., 2015)	Marco de arquitectura para el control de redes virtualizadas	Simulación	No	Eficiencia en energía y QoS
<i>SDCSN</i> (Flauzac et al., 2015)	Arquitectura para implementar SDN en clusters de WSN	No	No	No
<i>SDN-WISE</i> (Galluccio et al., 2015)	Arquitectura de SDWSN basada en estados finitos	Simulación y Testbed	Abierta a descargar	RTT, eficiencia, tiempo de respuesta
<i>Multi-Task SDSN</i> (Zeng et al., 2015)	Algoritmo centralizado para optimizar la eficiencia en energía	Simulador Gurobi Optimizer	No	Tasa de sensado, eficiencia de energía
<i>SDN-ECCKN</i> (Wang et al., 2016)	Algoritmo de técnicas de agenda de sueño centralizado para ahorro de energía	Simulación Matlab	No	Tiempo de vida de la red



(Giulio, 2016)	Marco de arquitectura para implementar SDN en una red 6LoWPAN	Simulación Cooja	No	RTT, pérdida de paquetes, cantidad de bytes transmitidos y overhead
WARM (C. Silva et al., 2016)	Proveen de un marco para SDWSN	Simulación Cooja	No	Overhead y aspectos de memoria
SDN-TAP (Fotouhi et al., 2016)	Propone concepto a SDN-IoT con aplicación al cuidado de la salud	Simulación Cooja	No	Retraso, pérdida de paquetes y aspectos de tráfico
(Beyene, 2017)	Controlador en Python y evaluación de SDN-WISE	Testbed	Abierta a descarga	Tiempo de convergencia, perdida de paquetes, aspectos de movilidad
SDWSN6Lo (Fernando et al., 2018)	Arquitectura para implementar SDN a una red 6LoWPAN	Simulación	No	Aspectos de energía y memoria

Como se ha podido observar en la comparativa de las propuestas de SDN para WSN, la mayoría solo muestra un concepto teórico de algunas partes de la arquitectura. Además, pocas propuestas llevan a cabo una implementación que esté disponible para poder ser utilizada en la evaluación de nuevos protocolos o bloques de la arquitectura complementarios. De la comparativa realizada en la revisión bibliográfica de las implementaciones de SDN para WSN, se encontró que la propuesta en (Galluccio et al., 2015) llamada SDN-WISE es la más completa. SDN-WISE incluye el soporte de utilizar técnicas de ciclo de trabajo, compatibilidad con el estándar IEEE 802.15.4 para implementar en los nodos y un proceso de descubrimiento de topología para formar la red. Además, el código fuente de SDN-WISE está disponible al público y cuenta con una comunidad de soporte que incluye al desarrollador de SDN-WISE.

Dado que el objetivo de esta tesis es evaluar el impacto que tiene el ciclo de trabajo sobre el descubrimiento de topología en el despliegue de una red SDN implementada sobre WSN, se eligió a SDN-WISE como plataforma para llevar a cabo las evaluaciones necesarias.

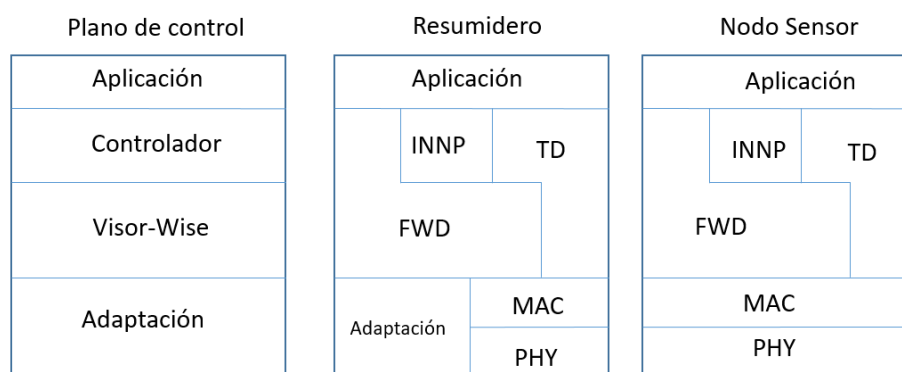
## 2.4 Redes WSN utilizando SDN-WISE

Los autores en (Galluccio et al., 2015) definen a SDN-WISE como *Software Defined Networking solution for Wireless Sensor Networks* o una solución de redes definidas por software para las redes inalámbricas de

sensores, toman como base el trabajo de (Costanzo et al., 2012) e integran mejoras como una solución en base a estados para simplificar la administración de la red. SDN-WISE busca mejorar y superar algunas limitaciones observadas en las implementaciones propuestas en la literatura como:

- Detalles del protocolo, el cual en la mayoría de los casos no se han establecido y son fundamentales para la correcta operación de la red.
- Evaluaciones de desempeño de las soluciones propuestas, que en la mayor parte de los casos no se han llevado a cabo.

Además SDN-WISE toma en cuenta los retos que tienen las WSN, permitiendo que sea eficiente en el uso de los recursos del nodo sensor, de tal forma que da soporte a utilizar técnicas de ciclo de trabajo (Wei et al., 2002) y agregación de datos (Intanagonwiwat et al., 2002) de forma nativa. SDN-WISE es una solución basada en máquinas finitas de estados, en la cual los nodos sensores realizan decisiones basados en su información local, con ello se busca reducir la señalización entre los nodos y el controlador e implementar políticas que no eran posibles en las propuestas anteriores. SDN-WISE está basado en las especificaciones de la capa física y capa de acceso al medio de IEEE 802.15.4 con una arquitectura compuesta por tres elementos: nodo resumidero, nodo sensor y el controlador, dichos componentes y sus elementos se pueden apreciar en la figura 12.



**Figura 12** Arquitectura de SDN propuesta por SDN-WISE (Galluccio et al., 2015).

En dicha arquitectura el resumidero actúa como puente entre el controlador y los nodos sensores, mediante el cual todos los paquetes de control deberán pasar a través del resumidero para alcanzar al controlador. El plano de datos está conformado por el resumidero y los nodos sensores, y es responsable del envío de los paquetes mediante la capa *Forwarding* (FWD), del descubrimiento local de la topología

mediante la capa de *Topology Discovery* (TD) y el procesamiento de los paquetes dentro de la red como la agregación de datos desarrollada por la capa *In-network Packet Processing* (INPP). Por otro lado el plano de control contiene elementos que se encargan de la administración de la red y de la formación global de la topología de la red. Además dentro del controlador se almacena la capa de *WISE-VISOR* el cual es una capa de virtualización que habilita a varios controladores operar sobre el mismo hardware. La capa de adaptación contenida tanto en el resumidero como en el controlador está encargada de dar formato a los mensajes necesarios para que dichos módulos se comuniquen. Dentro de la capa de *Forwarding* responsable del manejo de los paquetes SDN-WISE entrantes se define la estructura y los tipos de paquetes que serán enviados a lo largo de la red. El encabezado está fijo a una longitud de 10 bytes compuesto por 7 campos como se muestra en la tabla 3

**Tabla 3** Campos del encabezado del paquete de SDN-WISE.

Byte(s)	Nombre	Descripción
0	Net	Identificador de la red
1	Len	Longitud total del paquete
2-3	DST	Dirección destino
4-5	SRC	Dirección fuente
6	TYP	Tipo de paquete
7	TTL	Número de saltos restantes
8-9	NHX	Dirección del siguiente salto

#### 2.4.1 Mensajes de control para la formación de la red SDN-WISE

SDN-WISE define 8 tipo de paquetes que comenzarán con el encabezado de la tabla 3 y continuarán con la información perteneciente, dichos paquetes se enlistan en la tabla 4, de los cuales 3 son utilizados para dar formación a la red: Beacon, Report, Openpath:

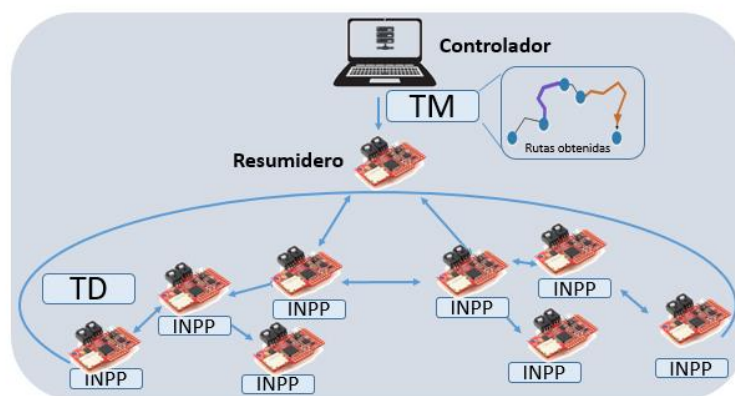
**Tabla 4** Mensajes de control para red SDWSN.

<i>Mensaje de control</i>	<i>Descripción</i>
<i>Datos</i>	Este tipo de paquete tiene contenido de información y el tamaño estará definido por la carga útil.
<i>Beacon</i>	Este paquete reporta la distancia de un nodo fuente al resumidero así como su nivel de batería. Este paquete es de tipo broadcast.
<i>Report</i>	Este paquete reporta la lista de vecinos recopilada por cada nodo teniendo un máximo de 35 vecinos para cada paquete, y es enrutada al resumidero. Este paquete es de tipo <i>unicast</i> .
<i>Request</i>	Encapsula un paquete que no ha encajado con alguna regla en la tabla de flujo, solicitando la información mediante el resumidero.
<i>Openpath</i>	Este paquete es usado para crear un camino entre dos nodos en la red, con el objetivo de reducir el número de mensajes de control enviados del plano de control
<i>Response</i>	Este paquete es una respuesta a una solicitud del paquete Request y contiene una entrada de flujo del controlador
<i>Config</i>	Es usado para leer y/o escribir algunos parámetros de configuración
<i>RegProxy</i>	Este paquete es usado para informar al plano de control sobre la existencia e información del resumidero

#### 2.4.2 Formación de una red SDN-WISE

Para el proceso de descubrimiento de topología y generación de rutas para enviar la información se llevan a cabo procesos dentro de los elementos que forman la red, como se puede apreciar en la figura 13. Dentro del controlador existe un módulo llamado *Topology Manager* (TM) que es el encargado de crear una vista global y coherente del estado actual de la red, mediante algún algoritmo de enrutamiento como Dijkstra. Para ello, se requiere recopilar información de topología local (vecinos a un salto) de los nodos sensores.

El protocolo *Topology Discovery* (TD) ejecutado por todos los nodos sensor es responsable de generar dicha información y entregarla al controlador.



**Figura 13** Módulos para formación de topología de SDWSN.

El protocolo TD mantiene actualizada la información sobre el siguiente salto hacia los controladores y sus vecinos actuales. Para este fin, el receptor de la red SDN-WISE transmite de forma periódica un paquete de descubrimiento de topología (paquete Beacon) a través del canal inalámbrico. Dicho paquete contiene la identidad del receptor que lo ha generado, un nivel de batería y la distancia actual desde el receptor, que inicialmente se establece en cero. El proceso de formación de la red se describe de la siguiente forma y se puede observar en la figura 14:

Un nodo sensor A que recibe un paquete Beacon del nodo sensor B (resumidero) realiza las siguientes operaciones:

- 1) inserta B en la lista de sus vecinos actuales junto con el RSSI actual y el nivel de la batería. Si B ya está presente en la lista de vecinos actuales, solo se actualizan los valores de RSSI y nivel de batería;
- 2) controla si ha recibido recientemente un paquete Beacon con un valor menor de la distancia actual desde el resumidero. Si este no es el caso, entonces el nodo A actualiza el valor reportado en el paquete TD al valor actual más uno y establece su siguiente salto hacia los controladores igual a B.
- 3) establece su nivel de batería en el campo correspondiente del paquete Beacon.
- 4) transmite su paquete Beacon actualizado a través del canal inalámbrico.

Así cada nodo que conforma la red recibe los paquetes Beacon de los nodos vecinos, formando así una tabla de vecindad en cada nodo. Posteriormente cuando el tiempo configurado en los paquetes Report de cada nodo es alcanzado, se transmitirá dicho paquete con dirección al resumidero mediante el nodo configurado como siguiente salto al controlador. Cada nodo que recibe este tipo de paquetes dirigidos hacia el controlador lo retransmiten al nodo establecido como su siguiente salto hasta llegar al resumidero. Una vez que el resumidero obtiene la información de la red mediante los paquetes Report enviados por cada nodo, informará al controlador cada una de las tablas de vecinos mediante una conexión serial. Dentro del controlador se realizará un algoritmo de enrutamiento tomando en cuenta la métrica establecida en las tablas de vecinos. Posterior al procesamiento de generar las rutas para cada nodo, el controlador las distribuirá secuencialmente conforme la información de cada uno de los nodos fue llegando, las rutas estarán contenidas en un paquete Openpath y entregados de forma unicast por el resumidero. Si se llega a presentar el caso de que un nodo no haya recibido la ruta, debido a que el paquete se perdió en el camino por ejemplo, dicho nodo podrá volver a solicitar la ruta cuando necesite enviar información. La solicitud de la ruta será por medio de un paquete Request dirigido al controlador, con ello el controlador volverá a enviar la ruta a dicho nodo. Con el objetivo de explicar mejor el proceso de TD, en la figura 14 b) se presenta un ejemplo que muestra la formación de una topología de red multisalto conformada por 3 nodos sensores y un resumidero, en un primer salto se encuentra los nodos 1.1 y 2.1, y en el segundo salto el nodo 1.2 (figura 14 a)).

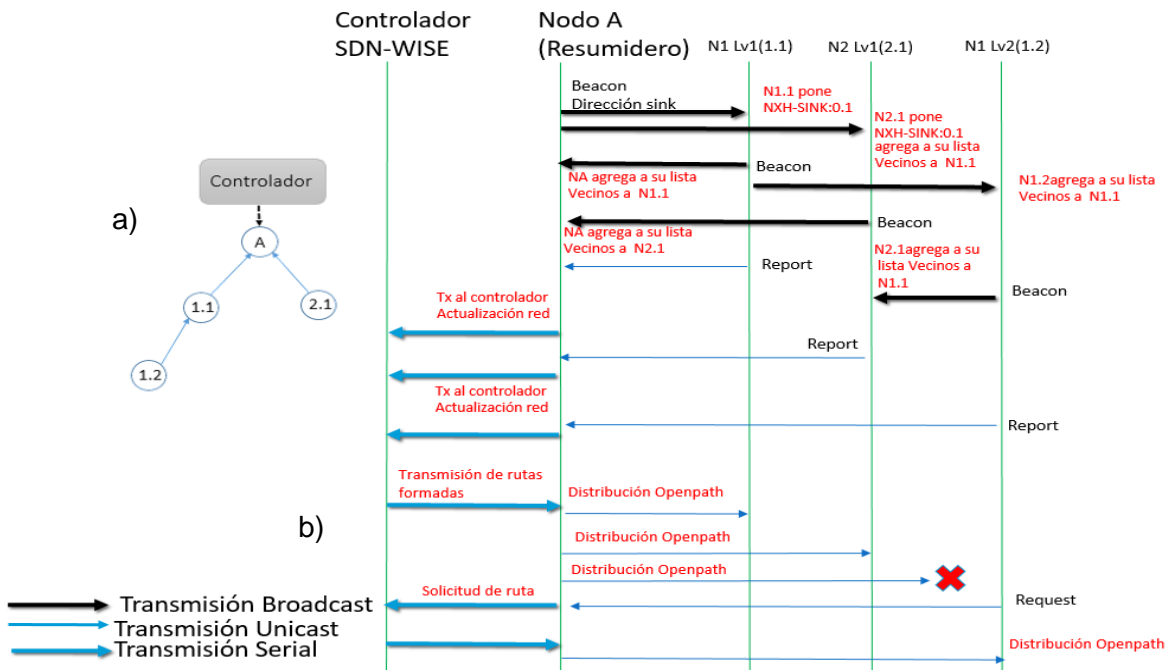


Figura 14 a) Proceso de descubrimiento de topología en red SDN-WISE, b) Topología multisalto SDN-WISE.

## Capítulo 3. Herramientas del entorno del simulador

---

### 3.1 Introducción

En este capítulo se detallan las herramientas y métodos utilizados para la generación del entorno de simulación para la evaluación del protocolo de descubrimiento de topología de los paradigmas SDWSN y 6LoWPAN. Además, se explican las características de los elementos utilizados para la obtención de las métricas definidas para la realización de esta tesis.

### 3.2 Contiki Os

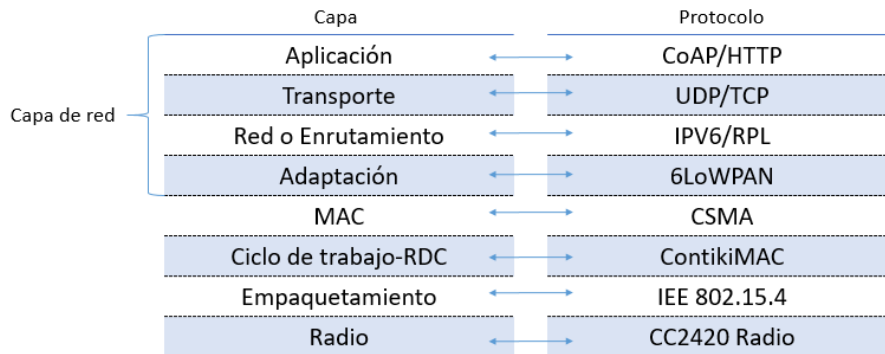
Contiki Os (Dunkels et al., 2004) es un sistema operativo (SO) multi-tarea diseñado especialmente para ambientes donde los nodos sensores están limitados en recursos, especialmente para dispositivos con una cantidad pequeña de memoria. Contiki es un SO basado en eventos, esto permite utilizar una programación multi-hilos aplicados a procesos mediante proto-hilos o *protothreads* que permiten el manejo eficiente de uso de memoria (Adam Dunkels et al., 2006). Contiki Os es de código abierto, desarrollado en lenguaje C donde los programas realizados se consideran procesos, es compatible con diferentes estándares de comunicación como, IPv4 e IPv6 y opera sobre diferentes plataformas de hardware como el CC2650 sensortag de Texas Instruments (Texas Instruments, 2015) o el mote Z1 Remote de Zolertia (Zolertia, 2010).

### 3.3 Contiki Os – Pila de protocolos

Contiki Os provee una pila de protocolos de red que permite la comunicación con otros nodos llamada *netstack*, basada en el modelo tradicional de comunicación OSI (*Open Systems Interconnection*). El netstack de Contiki Os se muestra en la figura 15, está diseñado bajo el principio de separación en cinco capas:

- Capa de red (*NETSTACK\_NETWORK*)

- Capa de acceso al medio (*NETSTACK\_MAC*)
- Capa del ciclo de trabajo del radio (*NETSTACK\_RDC*)
- Capa de empaquetamiento (*NETSTACK\_FRAMER*)
- Capa del radio (*NETSTACK\_RADIO*)



**Figura 15** Pila de protocolos de ContikiOs.

### 3.3.1. Contiki Os- Capa de red

La capa de red de Contiki Os toma en cuenta las funcionalidades de proveer direcciones, enrutamiento y estructura de los paquetes. Engloba cuatro sub-capas, una breve explicación de cada una de ellas es mostrada a continuación:

- **Sub-capas de aplicación:** contiene los protocolos de intercambio de mensajes como (Restful, CoAP, HTTP y MQTT) que complementan a las tareas proporcionadas por la capa de transporte para que los datos de información sean desplegados para su función.
- **Sub-capas de transporte:** contiene las librerías utilizadas por los protocolos de transporte UDP y TCP, esto lo realiza mediante el módulo uIPv6
- **Sub-capas de enrutamiento:** el protocolo por defecto es ContikiRPL que es una versión de RPL basada en la especificación del grupo de trabajo ROLL de la IETF.



- **Sub-capa de adaptación:** se encarga de la compresión y fragmentación de los encabezados IPv6 para ser utilizados sobre 6LoWPAN.

### 3.3.2. Contiki Os - Capa de acceso al medio

Contiki Os utiliza como base el algoritmo de acceso múltiple por sensado de portadora y prevención de colisiones no ranurado (*Unslotted CSMA/CA*) del protocolo IEEE 802.15.4. (Ko et al., 2006) como capa MAC. La capa MAC y la capa de RDC trabajan en conjunto para realizar las retransmisiones de un paquete colisionado, y agendar cada transmisión proveniente de capas superiores. El funcionamiento del algoritmo de CSMA/CA no ranurado está mostrado en la figura 16 y se describe de la siguiente forma:

Cada nodo antes de iniciar su transmisión esperará un tiempo aleatorio dentro de una ventana de contención formada por el valor *BE* (*back off exponent*) para posteriormente aplicar técnicas de sensado del medio con el fin de determinar el estado del canal dentro de su rango de cobertura. Si en dicho intento el canal se encuentra ocupado, la ventana de contención se aumentará y el nodo esperará durante un nuevo periodo de tiempo aleatorio para volver a sensar el canal e intentar realizar una nueva transmisión de ese paquete. Este proceso será realizado hasta que el número de intentos de transmisión haya excedido el valor definido denotado por *macMaxCSMABackoff*, al suceder esto el paquete será descartado. Si en alguno de los intentos el canal se encuentra libre, el paquete será transmitido.

El sensado del canal es realizado mediante la evaluación del canal libre (*Clear Channel Assesment CCA*) y puede realizarse de tres formas (IEEE Computer Society, 2003):

- El medio se determinará ocupado si el indicador de potencia de la señal recibida (RSSI) es más alto que cierto umbral predefinido.
- El medio se determinará ocupado si se detecta al menos una señal con características de modulación y esparcimiento.

- Se asume que el medio está ocupado si ambas condiciones anteriores son detectadas.

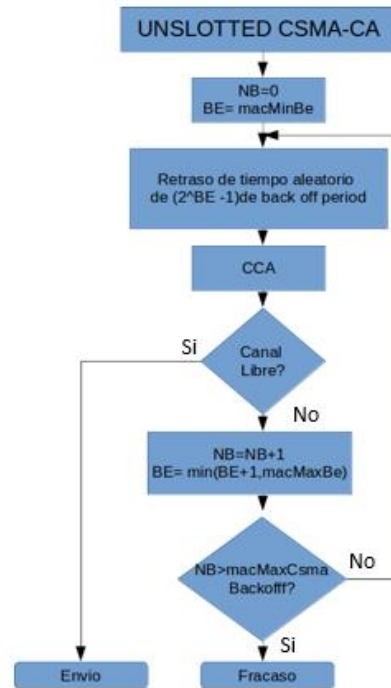


Figura 16 Algoritmo de CSMA-CA no ranurado.

### 3.3.3 Contiki Os- Capa de RDC

El ciclo de trabajo del transceptor de radio es una estrategia propuesta para las WSN que permite reducir el consumo de energía inherente en las WSN (Ahmed, 2017). En Contiki OS se agrega una capa de ciclo de trabajo del radio en su pila de protocolos llamada RDC. Dicha capa controla el proceso de transmisión de cada nodo al configurar periódicamente la conmutación del radio transceptor entre un periodo corto activo (al escuchar el canal) y periodos extensos inactivos al mantener el radio “dormido”. El ciclo de trabajo se define como la tasa entre la duración del periodo activo y el intervalo entre dos periodos cortos sucesivos de sensado del canal. Al implementar estos mecanismos se busca ahorrar energía y evitar fenómenos como *Idle Listening* al tener el transceptor apagado la mayor parte del tiempo. Se asume que entre más corto sea el ciclo de trabajo, menor será el consumo de energía.

ContikiMac (Adam Dunkels, 2011) es el mecanismo por defecto en Contiki Os para administrar la operatividad del radio gracias a la mejora en eficiencia del consumo de energía que provee con respecto

los protocolos existentes (Michel y Quoitin, 2014) como B-MAC (Polastre et al., 2004), X-MAC (Buettner et al., 2006) y BoX-MAC (Moss y Levis, 2008).

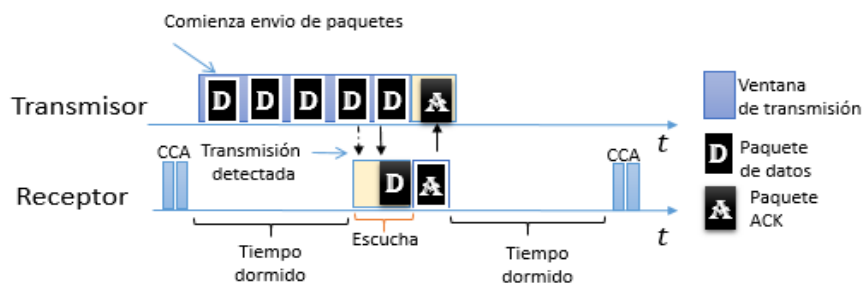
Al implementar el protocolo ContikiMac periódicamente el transceptor del radio se enciende por una cantidad  $t_{RDC}$  milisegundos para sensor el canal a la espera de escuchar posibles transmisiones de nodos vecinos (*Clear Channel Assessment CCA*). A este periodo se le denomina periodo *wake-up*, el resto del tiempo el nodo se mantendrá apagado. Este intervalo de conmutación de estados se define como frecuencia de ciclo de trabajo ( $f_{RDC}$ ), y se calcula por la ecuación (1)

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

ContikiMac no utiliza mensajes de señalización y no agrega encabezados adicionales en el paquete. Así los paquetes que se envían de la capa de RDC son los paquetes recibidos por la capa MAC. Si una transmisión es detectada durante un periodo de *wake-up*, el receptor se mantendrá encendido para recibir el paquete. Cuando el paquete haya sido recibido exitosamente, el receptor enviará un paquete de reconocimiento ACK al nodo transmisor para indicar que no hubo problemas en la transmisión.

Para efectuar una transmisión se realizarán diferentes procesos dependiendo del tipo de paquete:

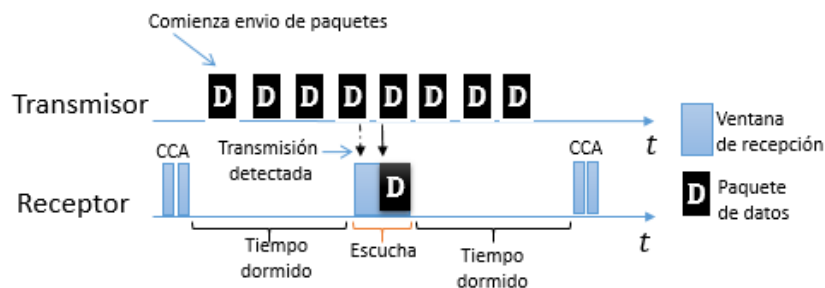
- Si el paquete es de tipo unicast, el radio transmisor se mantendrá encendido durante la transmisión y el paquete se enviará  $n$  número de veces. Esto se realizará hasta que el receptor confirme de recibido el paquete mediante un paquete ACK, como se observa en la figura 17



**Figura 17** Transmisión de paquetes unicast con ContikiMac.

- Si el paquete es de tipo broadcast, el radio transmisor estará conmutando el encendido del radio a lo largo de la duración del paquete. El paquete se enviara  $n$  número de veces fijo, esta cantidad

de paquetes estará en función del periodo  $t_{RDC}$ . Para este tipo de paquetes el receptor no envía un paquete ACK para indicar que la recepción fue exitosa. Este proceso se puede observar en la figura 18.



**Figura 18** Transmisión de paquetes broadcast con ContikiMac.

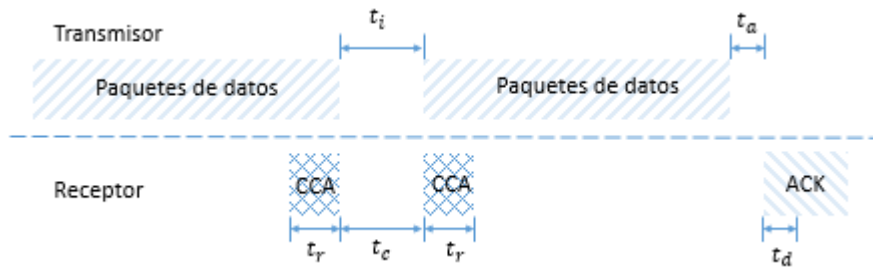
La tabla 5 muestra la cantidad de paquetes extras generados por ContikiMac y la duración de enviar esa cantidad de paquetes (*Strobe time*), para un paquete *broadcast* de control del paradigma de SDWSN y el enrutamiento de RPL.

**Tabla 5** Cantidad de paquetes generados por ContikiMac.

Frecuencia $f_{RDC}$	Paquete Beacon –SDN-WISE	Strobe time	Paquete DIO-RPL	Strobe time
8 Hz	49 pqts	126.4 ms	30 pqts	129.4 ms
16 Hz	25 pqts	64.8 ms	15 pqts	64.6 ms
32 Hz	13 pqts	33.4 ms	8 pqts	34.2 ms
64 Hz	7 pqts	17.84 ms	5 pqts	21.2 ms

En ContikiMac el nodo sensor realiza dos CCAs consecutivos para determinar en base a la potencia de la señal recibida (RSSI) si existe una transmisión entrante. Si la respuesta de los CCAS es exitosa, esto es, el canal está libre de transmisiones, el radio inmediatamente irá a dormir. Si la respuesta es fallida, quiere decir que el canal está ocupado, el radio se mantendrá despierto y mandará llamar un procedimiento denominado *fast sleep optimization* para intentar determinar si la señal recibida es debido a ruido del canal o si es un paquete entrante. En caso de que determine que la señal es ruido, mandara a dormir al radio.

La distribución de tiempos para la fase de *wake-up* es importante para hacer las transmisiones de paquetes funcionales. En la figura 19 se muestra un ejemplo de la distribución de tiempos y los requerimientos específicos para el correcto sensado del canal y transmisión de paquetes.



**Figura 19** Esquema de Distribución de tiempo para transmisión de paquetes.

Donde los requerimientos mostrados en la figura son los siguientes:

- $t_i$ : el intervalo entre cada paquete de transmisión.
- $t_r$ : el tiempo requerido para obtener un RSSI estable, necesitado para indicar un CCA estable.
- $t_c$ : el intervalo entre cada CCA.
- $t_a$ : el tiempo entre recibir un paquete y enviar un paquete de reconocimiento.
- $t_d$ : el tiempo requerido para detectar exitosamente un ack del receptor.

Con base en esto algunas consideraciones deben ser tomadas en cuenta:

- $t_i < t_c$  el intervalo entre cada transmisión de paquetes debe ser menor que el intervalo entre cada CCA, para asegurar que tanto el primer o el segundo CCA sea capaz de observar una transmisión.
- El tiempo de transmisión del paquete debe ser mayor a la suma del intervalo entre cada CCA y la duración de dos CCA. Esto es ec. (2)

$$t_s > t_r + t_c + t_r \quad (2)$$

- El intervalo entre dos transmisiones denotado por  $t_i$  debe ser mayor que el tiempo necesitado para enviar y recibir un ACK, esto es ec. (3)

$$t_i > t_a + t_d \quad (3)$$

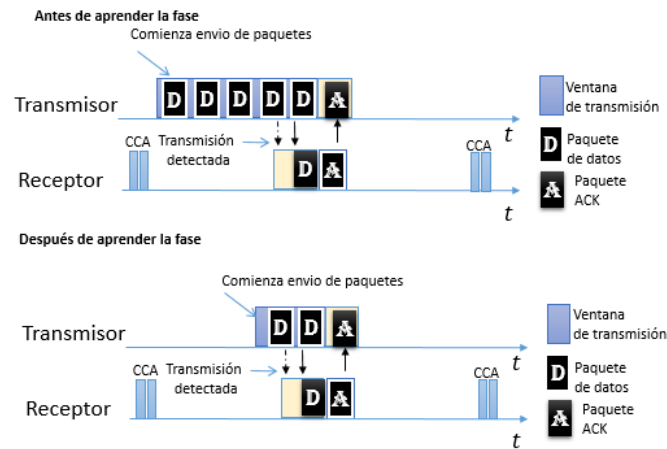
Estas consideraciones son resumidas en la ecuación (4)

$$t_a + t_d < t_i < t_c < t_c + 2t_r < t_s \quad (4)$$

Las razones para que exista una transmisión fallida son las siguientes:

- La transmisión fue un paquete de tipo unicast y no se recibió un paquete ACK después del intervalo de tiempo  $t_d$ .
- No se recibió el paquete ACK debido a que dos o más nodos intentaron enviar o retransmitir un paquete en el mismo periodo de contención ocupando al mismo tiempo el canal causando una colisión entre los paquetes.
- Debido a un paquete descartado por la capa MAC al llegar al límite permitido de intentos por acceder al canal o debido a superar el límite del tamaño de la cola de la capa MAC para agendar una transmisión.

Como un mecanismo adicional implementado en ContikiMac, se utiliza el mecanismo de bloqueo de fase, esto es cuando un nodo recibe un ACK aprende la agenda de tiempo de encendido de dicho destino. El transmisor posteriormente hace uso de este conocimiento durante los siguientes envíos de paquetes a ese vecino. Como se muestra en la figura 20, una vez aprendida la agenda puede empezar a realizar las transmisiones de los paquetes justo antes de que el receptor despierte. De esta forma el mecanismo decreta el tiempo y la energía gastada para realizar una transmisión.



**Figura 20** Transmisión de bloqueo de fase por parte de ContikiMAC.

Otro mecanismo de la capa de RDC implementado en Contiki Os es el de nullrdc, tal como su nombre lo indica proporciona un control de ciclo de trabajo nulo, esto quiere decir que nunca conmuta al transeptor manteniéndolo siempre encendido.

En la evaluación de este trabajo de tesis se utilizará el protocolo nullrdc como una referencia base para para observar el impacto que tiene agregar técnicas de ciclo de trabajo (usando el protocolo RDC) al proceso de formación de la red.

### 3.3.4 Contiki Os- Capa de Empaquetamiento

La capa de empaquetamiento que brinda Contiki Os es una colección de funciones auxiliares que son llamadas para crear y darle formato a un paquete con datos para ser transmitido, al igual de hacer un análisis de los datos cuando se recibe un paquete. Algunos ejemplos que se pueden encontrar en esta capa son: 802154\_framer que toma en cuenta las características de los paquetes del estándar IEEE 802.15.4 y contikimac\_framer que además de tomar en cuenta las características del estándar IEEE 802.15.4 hace adaptaciones requeridas cuando se utiliza Contikimac como técnica de ciclo de trabajo.

### 3.3.5 Contiki Os- Capa de Radio

Por último, la capa de radio se encuentra en el nivel más bajo del *netstack* de ContikiOs, controla el radio del sensor mediante un manejador y es responsable en enviar y recibir la información mediante enlaces inalámbricos del medio. En esta capa la información llega en forma de bytes o paquetes enteros mediante un manejador de interrupciones. Los datos entrantes se leen uno a uno y son colocados en un buffer en donde se realiza un mecanismo de encuesta al proceso. Este mecanismo de encuesta hace que el proceso sea dirigido hacia un evento especial a las capas superiores. Está basado en la capa física del estándar de IEEE 802.15.4

## 3.4 Simulador de redes Cooja

Cooja (Österlind et al., 2006) es el simulador de redes inalámbricas de sensores multi-nivel desarrollado en lenguaje JAVA que utiliza el código fuente de Contiki Os. Permite la simulación de diferentes capas de comunicación, desde la capa física hasta la capa de aplicación, como también la emulación de un conjunto de hardware de nodos sensores. Los periféricos que se encuentran en los nodos de hardware simulados son llamados interfaces, y permiten al simulador detectar eventos disparados, como el tráfico de red entrante. Todas las interacciones entre la simulación y los nodos simulados son llevadas a cabo mediante *plugins*, como ejemplo el control de la simulación que permite iniciar o pausar una simulación. Una visualización del entorno de simulación se puede observar en la figura 21, en ella se muestra los *plugins* de *Timeline* (línea de tiempo) que despliega la duración de cada paquete transmitido y recibido por los nodos, el *Mote output* que sirve como consola para mostrar las interacciones de cada mote, y el *Network* que es la ventana del entorno mapeado en con una cuadrícula donde se ubican los nodos.

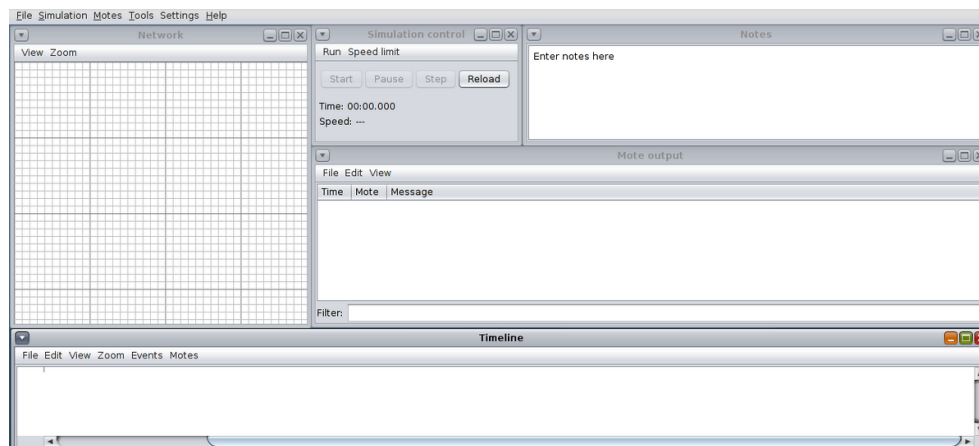


Figura 21 Entorno de simulación Cooja.



El simulador Cooja permite a través de sus características realizar simulaciones con resultados muy cercanos a los que ocurren utilizando un hardware físico, y permite un despliegue de los experimentos simulados directamente sobre los nodos reales, dando al usuario la facilidad de agregar funcionalidades personalizadas para simulaciones específicas. Una de las herramientas útiles del simulador para este trabajo de tesis es el *plugin* de *radio messages*, figura 22, dicha interfaz emula las características de tener un capturador de paquetes *sniffer*. Esto permite visualizar y analizar los paquetes intercambiados en el medio inalámbrico con las características del protocolo IEEE 802.15.4 y 6LoWPAN

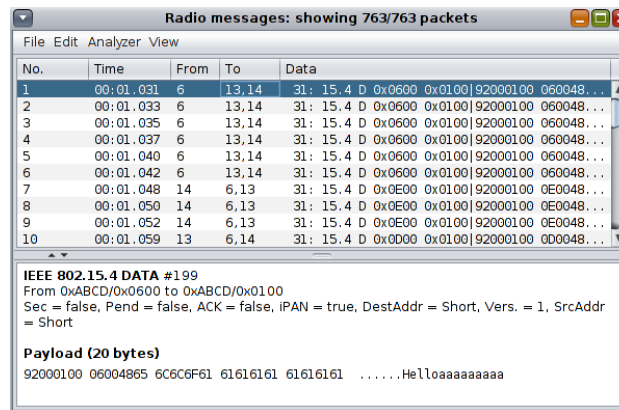


Figura 22 Herramienta del simulador : Radio message.

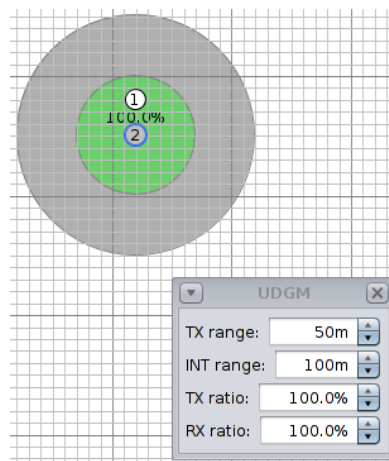
Además Cooja permite utilizar cuatro modelos de propagación para simular los efectos del canal inalámbrico. Cada modelo implementado es responsable de escuchar a un número registrado de transmisores y calcular que radio debe escuchar la información. Para lograr esto, cada modelo está diseñado para introducir bits erróneos dependiendo de la distancia, o simular obstáculos de tal forma que bloqueen o corrompan la información entre diferentes radios, así cada modelo de propagación afectará el comportamiento de éxito o fracaso de la señal transmitida (Österlind, 2006). Los modelos implementados en Contiki Os son los siguientes (Stehlík, 2011):

- **Unit Disk Graph Medium (UDGM) con pérdidas constantes:** Es un modelo simple donde el rango de transmisión está modelado como un disco ideal en donde los nodos fuera del disco no reciben los paquetes transmitidos.

- **Unit Disk Graph Medium (UDGM) con pérdidas de distancia:** Esta versión modela el rango de transmisión igual que la versión con pérdidas constantes. Sin embargo, añade algunas configuraciones:
  1. Las interferencias son tomadas en cuenta, si el paquete es interferido es considerado como perdido en base a una colisión.
  2. La tasa de éxito del de la transmisión y recepción es definido en base a un valor de probabilidad.

Dichas características pueden ser observadas en la figura 23.

- **Directed Graph Radio Medium (DGRM):** El propósito de este modelo es especificar la tasa de éxito de transmisión de forma asimétrica por enlace y definir los retardos de propagación de cada enlace.
- **Multipath Ray-tracer Medium (MRM):** Este modelo contrasta con los modelos implementados en otros simuladores de red como NS2 (Kevin Fall, 2011) u OMNET++ (Varga y Hornig, 2008), debido a que no está basado en mediciones empíricas sino en un enfoque analítico. La potencia recibida es calculada usando las formulas del modelo de Friis, y los obstáculos en el entorno son considerados como atenuadores. El modelo también permite configurar parámetros del radio como la potencia transmitida, la ganancia de la antena y la longitud de los paquetes.



**Figura 23** Modelo UDGM: rango de cobertura (área verde), rango de interferencia (área gris) y tasas de transmisión y recepción.

En este trabajo de tesis se decidió utilizar el modelo de UDGM con pérdidas de distancia debido a la configuración del rango de transmisión e interferencia dentro del área cobertura del nodo, al igual que es un modelo base en varios trabajos relacionados observados en la literatura (De Gante et al., 2014; Hieu et al., 2018). Cooja también permite incrementar la complejidad del análisis de simulación y el tiempo de corrida (*runtime*) en cada experimento. Esta extensión del *runtime* de las simulaciones es necesaria cuando se corren múltiples simulaciones, de tal forma que se puedan obtener resultados estadísticamente significativos al evaluar el mismo escenario con diferentes semillas. Cooja permite para estos casos la simulación autónoma de múltiples repeticiones del experimento.

### **3.5 Módulo de medición de energía.**

Una de las métricas principales de esta tesis es la comparación del consumo de energía empleado para la formación de la red sobre diferentes escenarios y parámetros de configuración. Dentro de Contiki Os se ha desarrollado una herramienta que permite estimar el consumo de energía de un nodo al realizar cierta tarea llamada Powertrace (Adam Dunkels et al., 2011), la cual toma en cuenta ciertos módulos como el consumo del transceptor o el consumo de procesamiento por parte del CPU.

Powertrace es un programa desarrollado en lenguaje C que puede ser adaptado como un *plugin* o una función al simulador COOJA y permite estimar con un 94 % de precisión el consumo de energía en los nodos al ejecutar cierta aplicación. Powertrace utiliza un rastreo de estado de energía para estimar el consumo de energía del sistema, utilizando estampas de tiempo cuando un componente entra y sale de un estado. Estos estados de energía son periódicamente impresos en entidades denominadas capsulas de energía. Los valores mostrados en estas capsulas representan la energía consumida para un ciclo específico por el transceptor y el microcontrolador durante diferentes estados, como transmitir, recibir o mantenerse en modo espera (*idle*), dicho valor se mide en ticks de reloj (Ahmed, 2017). Los ticks de reloj por segundo de un sistema están definidos como la velocidad de reloj de operación del procesador. Los modos de operación a medir están definidos en la tabla 6.

**Tabla 6** Modos de operación Powertrace.

<b>Modo de operación</b>	<b>Descripción</b>
<i>Low Power Mode (LPM)</i>	Usado para desplegar el tiempo en el que el dispositivo se encuentra en un estado pasivo a la espera de un posible evento
<i>CPU</i>	Usado para desplegar el tiempo empleado para realizar los cálculos y procesamiento de datos
<i>Transmisión del radio</i>	Usado para desplegar el tiempo empleado para las transmisiones realizadas por el nodo
<i>Recepción del radio</i>	Usado para desplegar el tiempo empleado por el radio al recibir paquetes.

Al habilitar la herramienta Powertrace en cada mote por cierto periodo de tiempo (configurado por el usuario) se desplegará un mensaje como el siguiente:

2502 P 194.255 14 11505 579545 0 2273 0 2273 689 32052 0 141 0 141

Donde el significado de cada campo se encuentra descrito en la tabla 7.

**Tabla 7** Análisis de trama Powertrace.

<b>Parámetro y valor</b>	<b>Descripción</b>
<b>Tiempo=2502</b>	Valor de tiempo transcurrido
<b>Dirección= 194.255</b>	Dirección del nodo
<b>Seq#= 14</b>	Número de secuencia
<b>all_cpu=11505</b>	Consumo de energía acumulado del CPU
<b>all_lpm=579545</b>	Consumo de energía acumulado para el modo Low Power
<b>all_transmit=0</b>	Consumo de energía acumulado para modo transmisión
<b>all_listen=2273</b>	Consumo de energía acumulado para modo recepción
<b>all_idle_transmit=0</b>	Consumo de energía acumulado en modo transmisión idle
<b>all_idle_listen=2273</b>	Consumo de energía acumulado en modo recepción idle
<b>cpu=689</b>	Consumo de energía del CPU dentro de un ciclo
<b>lpm=32052</b>	Consumo de energía del módulo LPM dentro de un ciclo
<b>transmit=0</b>	Consumo de energía en modo transmisor dentro de un ciclo
<b>listen=141</b>	Consumo de energía en modo receptor dentro de un ciclo
<b>idle_transmit=0</b>	Consumo de energía en modo transmisión idle dentro de un ciclo
<b>idle_listen=141</b>	Consumo de energía en modo recepción idle dentro de un ciclo.

El consumo de energía es calculado usando los valores típicos nominales de voltaje y corriente del mote que se está utilizando en cada modo de operación, mediante la ecuación (5)

$$E = \frac{((\tau_{tx} * I_{tx}) + (\tau_{rx} * I_{rx}) + (\tau_{cpu} * I_{cpu}) + (\tau_{lpm} * I_{lpm})) * V_{nom}}{RTimer\_Second} \quad (5)$$

Donde:

E es la energía consumida en el nodo expresada en Joule

$\tau_{tx}$  es el valor de tiempo utilizado para el proceso de transmisión.

$\tau_{rx}$  es el valor de tiempo utilizado para el proceso de recepción

$\tau_{cpu}$  es el valor de tiempo utilizado en el estado del microcontrolador esta encendido pero el radio apagado

$\tau_{lpm}$  es el valor de tiempo utilizado en el estado de Low power mode

$I_n$  es el valor de corriente definido en la hoja de datos del radio para cada estado, donde  $n$  es el modo de operación ( $tx, rx, cpu, lpm$ ).

$V_{nom}$  es el voltaje nominal del radio

$RTimer\_Second$  es el valor en ticks de reloj del microcontrolador por un segundo equivalente a 32768 ticks

## Capítulo 4. Configuraciones de evaluación

---

### 4.1 Introducción

Este capítulo describe el marco de evaluación y las configuraciones empleadas en la comparativa del algoritmo de descubrimiento de topología para la formación de la red de una red SDWSN y el estándar de facto, una red basada en RPL. Esta evaluación nos permitirá responder preguntas sobre ¿Qué protocolo tiene un menor esfuerzo en formar la red? , ¿Sobre qué escenario es factible utilizar algún protocolo?, ¿Cuál protocolo es más eficiente y cuáles son los costos-compromisos observados para que sea eficiente?

### 4.2 Configuraciones de la red RPL

Para realizar el análisis de la formación de una red RPL se utilizó la implementación que viene por defecto en Contiki OS, denominada ContikiRPL. ContikiRPL es una implementación de código abierto de RPL en lenguaje C, que opera sobre enlaces inalámbricos de baja energía y con pérdidas (Tsiftes et al., 2010), implementa el protocolo de RPL en su versión número 18 de las especificaciones hechas por parte del grupo ROLL (Winter et al., 2011) y presenta dos funciones objetivo programadas que pueden ser utilizadas por el usuario; la función objetivo cero “OF0” (Thubert, 2012) y la función objetivo por mínimo rango con histéresis “MRHOF” (Ganawali y Levis, 2012). Por defecto ContikiRPL opera en un enrutamiento con modo storing.

En este trabajo de tesis se utilizará la función objetivo MRHOF, debido a que es la función que viene por defecto en la implementación de ContikiRPL. La función objetivo MRHOF está basada en formar las rutas con el costo más pequeño, en este caso el camino con menor rango, por defecto utiliza la métrica de conteo de paquetes de transmisión esperada (ETX), esto permite al sistema encontrar los caminos estables con el mínimo número de transmisiones esperadas requeridas para transmitir un paquete exitosamente desde los nodos hacia el nodo raíz en la instancia DAG. La elección de esta función objetivo no causa afectaciones a la evaluación del proceso de descubrimiento de topología debido a que es utilizada en un proceso posterior, por lo que se considera que no se pierde generalidad al elegir la función objetivo MRHOF (Kechiche et al., 2017). La implementación de ContikiRPL hace una separación de diferentes módulos:

- El módulo lógico del protocolo, administra la información de la DODAG, mantiene un conjunto de candidatos a ser padres con su respectiva información y valida los mensajes RPL en un nivel lógico de acuerdo a las especificaciones de RPL.
- El módulo de construcción y análisis de los mensajes, realiza una traducción entre los mensajes ICMPv6 de RPL y las estructuras de datos propias de ContikiRPL
- El módulo de función objetivo, implementa una API en base a la función objetivo definido.

Para realizar el análisis de costo de las rutas y hacer las actualizaciones del enrutamiento a la red, ContikiRPL hace uso de un módulo externo que tiene información de los nodos vecinos. Dentro de esta actualización del entorno de la DODAG, ContikiRPL recalcula el costo del camino al resumidero mediante el enlace actualizado, y checa, si es necesario, seleccionar a un nuevo padre preferido. El costo del enlace refleja la métrica ETX por nodo, la cual es calculada usando una función de las medias móviles pesada exponencialmente sobre el número de transmisiones de capa MAC (J. Ko et al., 2011). La métrica ETX es usada para calcular el rango para la función objetivo MRHOF. Al integrar el algoritmo de RPL al netstack de Contiki Os se hicieron modificaciones a variables de la versión 18 del protocolo RPL, estas modificaciones son las siguientes:

Para el funcionamiento de RPL al utilizar la capa de RDC y su protocolo, Contiki OS modifica los valores de configuración del tiempo de generación de los mensajes de control para la formación y mantenimiento de la DODAG, estos cambios son reflejados en las siguientes variables:

- **Intervalo de envío de mensajes DIO** Esta variables es representada como  $2^{\text{DEFAULT\_DIO\_INTERVAL\_MIN}}$  ms. De acuerdo a la versión 18 el valor por defecto está definido por la variable `DEFAULT_DIO_INTERVAL_MIN` cuyo valor es 3 lo que significa que se comenzará enviando paquetes DIO cada 8 ms. Este valor es considerado demasiado bajo cuando se utiliza técnicas de ciclo de trabajo con intervalos de *wake-up* de cientos de milisegundos, como lo es en su configuración de 8 Hz al estar encendiendo el radio cada 125 ms para sensar el canal (Garcia-Loya, 2018). ContikiRPL configura esta variable por defecto a 12 teniendo un intervalo de envío de paquetes DIO a 4.096 s
- **Tiempo máximo de aumento de los mensajes DIO:** se encuentra definido por la variable `DEFAULT_DIO_INTERVAL_DOUBLING` el cual el estándar sugiere un valor máximo de 20 teniendo

así un valor de tiempo máximo en función de  $2^{\text{DEFAULT DIO INTERVAL DOUBLINGS} + \text{DEFAULT DIO INTERVAL MIN}}$ , esto es  $2^{20+3} = 2.33 \text{ horas}$ , sin embargo este valor es inadecuado debido a que al realizar el cambio en la variable del intervalo de mensajes DIO se volvería complejo dar mantenimiento a la red por la cantidad de mensajes generados. ContikiRPL sugiere un valor de 8 teniendo un intervalo máximo de generación de paquetes DIO de  $2^{12+8} = 17.47 \text{ minutos}$ .

Una evaluación de dichas modificaciones hechas por ContikiOS al protocolo RPL es realizado en (García-Loya, 2018) donde muestran mejoras en términos de PDR, retraso de envío paquetes y número de paquetes retransmitidos al aplicar técnicas de ciclo de trabajo cuando se implementan las modificaciones hechas por ContikiOs en comparación de las especificaciones hechas en el estándar. (García-Loya, 2018) concluye que el estándar RPL no está adecuado para ser utilizado con técnicas de ciclo de trabajo, con lo cual las modificaciones hechas por ContikiOS se encuentran validadas para usarse con técnicas de ciclo de trabajo. Un resumen de las configuraciones utilizadas para la evaluación de la formación de la red utilizando el protocolo de RPL está mostrado en la tabla 8.

**Tabla 8** Parámetros de configuración para el protocolo de RPL.

<b>Parámetro</b>	<b>Valor</b>
<i>Función objetivo</i>	MRHOF
<i>Modo de operación</i>	Modo Storing
<i>Métrica</i>	ETX
<i>RPL_DIO_INTERVAL_MIN</i>	12
<i>RPL_DIO_INTERVAL_DOUBLINGS</i>	8
<i>RPL_DIO_REDUNDANCY</i>	10
<i>RPL_DIS_INTERVAL</i>	60 segundos
<i>RPL_WITH_DAO_ACK</i>	habilitado
<i>RPL_DEFAULT_INSTANCE</i>	0x1e
<b>Tipo de paquete RPL</b>	<b>Tamaño (bytes)</b>
<i>DIS</i>	2
<i>DIO</i>	64
<i>DAO</i>	46
<i>DAO-ACK</i>	4
<i>ACK</i>	5
<b>Netstack</b>	<b>Protocolo</b>
<i>NETSTACK_CONF_NETWORK</i>	sicslowpan_driver
<i>NETSTACK_CONF_MAC</i>	csma_driver
<i>NETSTACK_CONF_RDC</i>	contikimac_driver y nullrdc_driver
<i>NETSTACK_CONF_FRAMER</i>	framer_802154
<i>NETSTACK_CONF_RADIO</i>	cc2420_driver



### 4.3 Configuraciones de la red SDWSN

Debido a que no hay un estándar definido para la utilización del paradigma de redes definidas por software SDN sobre las WSN teniendo solo algunas propuestas teóricas y unas cuantas implementaciones prácticas realizadas como se muestra en el capítulo 2.3.2, se decidió elegir para la evaluación de la red SDWSN, la propuesta realizada en (Galluccio et al., 2015), SDN-WISE. La siguiente lista muestra a detalle los motivos y características por los cuales se eligió esta plataforma:

- Es una implementación de código abierto que puede ser descargada desde un repositorio en github desde la siguiente liga: <https://github.com/sdnwiselab/sdn-wise-contiki.git>
- Es una implementación que tiene un desarrollo hecho en Contiki OS, siendo compatible con el simulador de redes Cooja, adquiriendo la posibilidad de ser implementado sobre múltiples plataformas que el sistema soporta.
- Tiene soporte por los desarrolladores, brindando información en foros sobre la utilización del sistema que puede servir como documentación.
- Es una implementación que ha sido tomada como fundamento para análisis sobre las características de redes SDWSN (Buratti et al., 2016; Giulio, 2016; Beyene, 2017; Hieu et al., 2018; Tsapardakis et al., 2018).
- La implementación ha sido desarrollada en base a un histórico de soluciones propuestas para insertar y adaptar el paradigma de SDN sobre las WSN, como se observa en el capítulo 2.3.2.

Con ello dicha implementación presume ofrecer las siguientes características:

- El objetivo de SDN-WISE es presentar una solución de que facilite la administración de una red brindando la apertura a nuevas aplicaciones en un entorno seguro.
- Este enfoque intenta dar solución a los requerimientos y limitaciones que se observan en las WSN, mediante la habilitación de módulos para dar soporte a redes con recursos limitados.

- La propuesta hace uso de un controlador lógicamente centralizado y una definición flexible de reglas de flujo, con el objetivo de dar robustez y flexibilidad a la red.
- Habilitan el uso de técnicas de ciclo de trabajo y agregación de datos y brindan un enfoque de generación de tablas de flujo basado en el estado de cada nodo de la red.

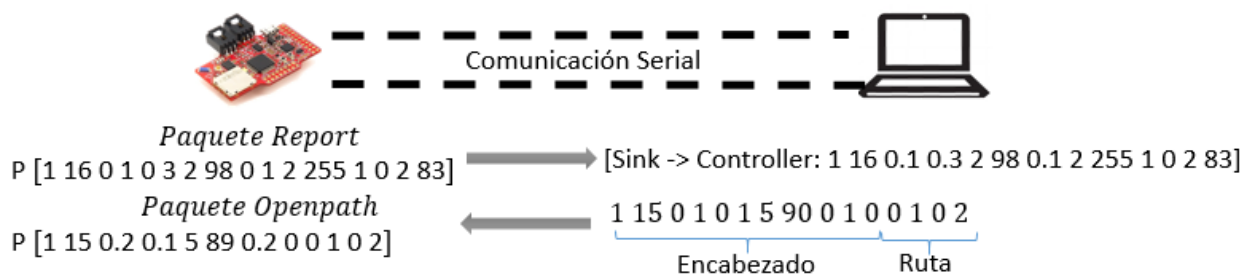
La implementación de Contiki OS no presenta un controlador definido y tomando en cuenta los objetivos de este trabajo de tesis para realizar una evaluación justa que considere los módulos necesarios presentados en los aspectos técnicos de la solución SDWSN, se decidió habilitar un controlador simple que obtenga las rutas en base a la información recolectada por el protocolo de descubrimiento de topología. El controlador que se utilizó, es encontrado en la evaluación realizada en (Hieu et al., 2018). El cual es un desarrollo que se encuentra de forma pública y puede ser descargado de la siguiente liga <https://github.com/hieunq95/sdn-wise-trickle.git>

El controlador es un módulo desarrollado en lenguaje JAVA, implementado en el simulador Cooja como un plugin, tiene como objetivo implementar el algoritmo de enrutamiento Dijkstra<sup>1</sup>, que es un algoritmo para determinar el camino más corto de un vértice origen a un vértice destino en base a una métrica dada. La métrica que utiliza el controlador es el valor del nivel del enlace entre cada nodo por medio del indicador de potencia de la señal recibida "RSSI" El controlador opera sobre una computadora y se comunica con el nodo resumidero mediante una capa de adaptación. La capa de adaptación está basada en una comunicación serial entre el controlador y el resumidero y toma como base la aplicación de Contiki OS "Collect-View" que permite obtener información de cada nodo sensor mediante protocolos de comunicación, como "USB-Serial". Una vez obtenido cada paquete Report de los nodos, la capa de adaptación en el resumidero da formato al paquete empezando con un delimitador para que el controlador pueda hacer un análisis del paquete. Posterior al proceso del enrutamiento Dijkstra llevado a cabo en el controlador, se enviará al resumidero cada una de las rutas formadas mediante un paquete Openpath que estará formado por un encabezado para que el resumidero pueda identificar dicho paquete recibido y la ruta contenida dirigida hacia el n-esimo nodo de la red.

La figura 24 muestra la traducción de los paquetes Report y Openpath intercambiados entre el resumidero y el controlador para que puedan ser leídos y utilizados dentro de cada sistema:

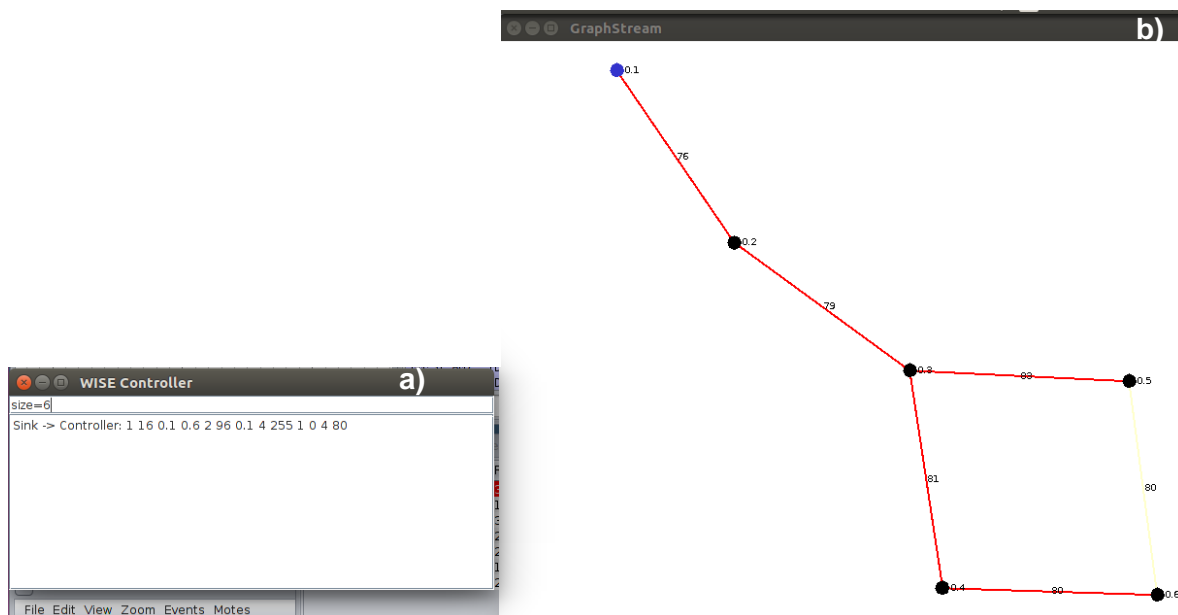
---

<sup>1</sup> Para mayor información del algoritmo puede consultar las paginas 366-369 de(Tanenbaum y Wetherall, 2011)



**Figura 24** Comunicación entre el resumidero y el controlador en red SD-WISE.

La figura 25 a) muestra una imagen de la consola del controlador, así como la ventana del gráfico de la red, figura 25b), generadas a partir del plugin “WISE Controller”.



**Figura 25** Plugin controlador SDN-WISE. a) Consola del controlador b) Ventana de grafica de la red.

El flujo de proceso de un paquete tomando en cuenta el enrutamiento realizado por el controlador de SDN-WISE es el siguiente:

Debido a que no existe un estándar definido para las implementaciones SDWSN tampoco existe un manejador para los paquetes formados de este tipo, por lo que la parte de red del *netstack* de ContikiOs estará conformado por el manejador Rime. El protocolo Rime permite tener una comunicación entre primitivas con las demás capas pertenecientes del *netstack*, no tiene un protocolo de enrutamiento

definido debido a que toma en consideración la información proveniente de las aplicaciones que están en las capas superiores a él. En este caso, hará caso de la información proveniente de las rutas obtenidas del controlador en la tabla de flujo implementada, indicando a la capa inferior hacia que nodo transmitir la información enviando los paquetes salientes a la capa MAC. Al intercambiar los mensajes de control para el proceso de formación de la SDWSN, existe la restricción que la periodicidad de los paquetes Beacon debe ser menor a la periodicidad de los paquetes Report  $\tau_{Report} > \tau_{Beacon}$ . Esto debido a que los paquetes Report son encargados de informar al resumidero sobre la información de vecindad perteneciente a cada nodo y para ello las tablas de vecindad en cada nodo deben ser formadas con antelación. Con esta restricción se deja a que la convergencia de la red estará realizada después del periodo  $\tau_{Report}$  a elegir. Este requerimiento nos dirige a saber que combinación es adecuada para la formación de la topología de la red. Por ello esta evaluación propondrá algunas configuraciones de periodicidad de estos paquetes para evaluar cuál de ellas es la óptima y el impacto que se genera al aplicar técnicas de ciclo de trabajo. Las configuraciones utilizadas para evaluar la red SDWSN se encuentran en la tabla 9.

**Tabla 9** Configuraciones utilizadas para la evaluación de la red SDWSN.

<b>Parámetro</b>	<b>Valor</b>	
<i>Métrica</i>	RSSI	
<i>Configuraciones de envío de paquetes de control</i>	$\tau_{Report} = 20 \text{ seg}$	$\tau_{Beacon} = 4, \text{seg}$
		$\tau_{Beacon} = 2, \text{seg}$
		$\tau_{Beacon} = 1, \text{seg}$
	$\tau_{Report} = 10 \text{ seg}$	$\tau_{Beacon} = 4, \text{seg}$
		$\tau_{Beacon} = 2, \text{seg}$
		$\tau_{Beacon} = 1, \text{seg}$
	$\tau_{Report} = 5 \text{ seg}$	$\tau_{Beacon} = 4, \text{seg}$
		$\tau_{Beacon} = 2, \text{seg}$
		$\tau_{Beacon} = 1, \text{seg}$
<b>Tipo de paquete en SDN-WISE</b>	<b>Tamaño (bytes)</b>	
<i>Beacon</i>	10 + 2	
<i>Report</i>	10 + 2 + (3 *# vecinos)	
<i>Openpath</i>	10 + (2* # de nodos en el camino)	
<i>Request</i>	10 + Carga útil	
<b>Netstack</b>	<b>Protocolo</b>	
<i>NETSTACK_CONF_NETWORK</i>	rime_driver	
<i>NETSTACK_CONF_MAC</i>	csma_driver	
<i>NETSTACK_CONF_RDC</i>	contikimac_driver y nullrdc_driver	
<i>NETSTACK_CONF_FRAMER</i>	contikimac_framer	
<i>NETSTACK_CONF_RADIO</i>	cc2420_driver	

## 4.4 Métricas de evaluación de desempeño

Para llevar cabo la comparativa de SDWSN y la red basada en RPL se cuantificó el desempeño de ambos enfoques considerando las métricas de consumo de energía utilizado para formar la red y el tiempo empleado en formar la red. Además se medirá la cantidad de paquetes de control intercambiados en cada una de las redes para formar la red.

### 4.4.1 Tiempo de Convergencia

La primer métrica desempeño de interés es el tiempo de convergencia, los nodos sensores en una WSN necesitan formar la topología de tal manera que puedan comunicarse, por ello el tiempo de configuración de la red es una métrica crucial para ser evaluada. En este trabajo de tesis, el tiempo de convergencia  $\tau_c$ , como se observa en la ecuación 6 se define como la diferencia de tiempo empleado para la distribución de las rutas formadas en cada proceso de formación de las redes a evaluar al punto en que cada nodo pueda ser operable para enviar paquetes de datos hacia el nodo resumidero denotado por  $\tau_f$ , y el valor de tiempo de inicio de la red denotado por  $\tau_i$

$$\tau_c = \tau_f - \tau_i \quad (6)$$

Para la red RPL, el tiempo de convergencia está definido al punto en que el último nodo hoja de la red hace el proceso de intercambiar un paquete DAO con su nodo padre, y este le responda con un paquete DAO-ACK. Para la red SDWSN, el tiempo de convergencia está definido al punto en que todos los nodos que pertenecen a la red aprenden la ruta diseminada por el controlador mediante los paquetes Openpath.

### 4.4.2 Consumo de Energía

El consumo de energía es una métrica que nos indica la cantidad de esfuerzo empleado para formar la red durante el tiempo de convergencia o formación de la red empleada. Este consumo está evaluado en Joules, se medirá a partir del envío del primer paquete de control transmitido al iniciar la red hasta el punto de

paro definido por la métrica de tiempo de convergencia de la red. Para medir el consumo de energía se hace uso de la herramienta PowerTrace habilitada en Contiki Os y su funcionamiento se encuentra explicado en el capítulo 3.5. Se medirá el consumo de energía promedio de los nodos sensores que emplean ciclo de trabajo, durante los estados de transmisión y recepción de paquetes en el proceso de formación de la red, dicho consumo queda denotado por la ecuación 7

$$E = \frac{((\tau_{tx} * I_{tx}) + (\tau_{rx} * I_{rx})) * V_{nom}}{RTimer\_Second} \quad (7)$$

Donde:

- $\tau_{tx}$  es la cantidad de tiempo utilizado por el proceso de transmisión en ticks de reloj
- $\tau_{rx}$  es la cantidad de tiempo utilizado por el proceso de recepción en ticks de reloj
- $I_{tx}$  es la cantidad de corriente nominal empleada al transmitir, denotada en amperes
- $I_{rx}$  es la cantidad de corriente nominal empleada al recibir, denotada en amperes
- $V_{nom}$  es la cantidad de voltaje nominal de alimentación del dispositivo, denotado en volts.
- $RTimer\_Second$  es la cantidad de tiempo total de utilización del sistema, denotada en ticks de reloj.

#### 4.4.3 Overhead de la red.

El *overhead* de comunicación se conoce como el número total de paquetes de control de un algoritmo de enrutamiento transmitidos de un nodo a otro, estos paquetes de control incluyen la generación de tablas de ruteo y la preparación para diseminar los paquetes de datos de un nodo sensor en la red. Para medir el *overhead* en la red RPL se tomará en cuenta la suma de todos los paquete de control intercambiados por los nodos que forman la red en función al tiempo de convergencia, esto es la suma de todos los paquetes

DIS, DIO, DAO y DAO-ACK que cada nodo intercambi6 para formar la red, como se denota en la ecuaci6n 8

$$OH_{RPL} = \sum_{i=1}^N (Dio_i + Dis_i + Dao_i + Daoack_i) \quad (8)$$

Donde

- $OH_{RPL}$  cantidad de paquetes de control de la red RPL
- $Dio_i$  n6mero total de paquetes Dio enviados por el i-esimo nodo durante toda la simulaci6n.
- $Dis_i$  n6mero total de paquetes Dis enviados por el i-esimo nodo durante toda la simulaci6n.
- $Dao_i$  n6mero total de paquetes Dao enviados por el i-esimo nodo durante toda la simulaci6n.
- $Daoack_i$  n6mero total de paquetes Dao-ack enviados por el i-esimo nodo durante toda la simulaci6n.
- $N$  cantidad de nodos sensores en la red

Para medir el *overhead* en la red SDWSN se tomar6 en cuenta la suma de los paquetes de control transmitidos en base al n6mero de nodos pertenecientes a la red en funci6n del tiempo de convergencia. Esto es la suma de todos los paquetes Beacon, Report, Openpath y Request generados por los nodos sensores y el resumidero conectado al controlador de SDN-WISE pertenecientes a la red, como se denota en la ecuaci6n 9

$$OH_{SDWSN} = \sum_{i=1}^N (Beacon_i + Report_i + Opath_i + Request_i) \quad (9)$$

Donde

- $OH_{SDWSN}$  cantidad de paquetes de control de la red SDWSN
- $Beacon_i$  n6mero total de paquetes Beacon enviados por el i-esimo nodo durante toda la simulaci6n

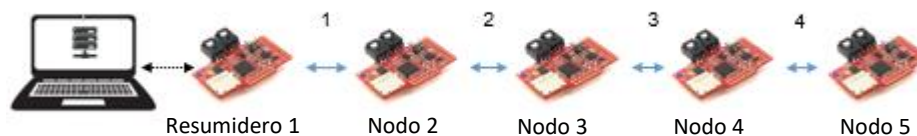
- $Report_i$  número total de paquetes Report enviados por el i-esimo nodo durante toda la simulación
- $Opath_i$  número total de paquetes Openpath enviados por el i-esimo nodo durante toda la simulación
- $Request_i$  número total de paquetes Request enviados por el i-esimo nodo durante toda la simulación
- $N$  cantidad de nodos sensores en la red

## 4.5 Escenarios de evaluación

Además de las métricas observadas en la sección anterior, la evaluación del impacto del ciclo de trabajo en la formación de la red, toma en cuenta aspectos principales en la topología que se pueden encontrar, como la cantidad de usuarios que conforman las WSN así como la distribución y localización de estos usuarios mediante el número de saltos, con el objetivo de observar cómo estas características pueden afectar a la formación de la red y evaluar qué tecnología presenta mejor desempeño sobre el escenario. Para ello se presenta los escenarios a evaluar propuestos en este trabajo.

### 4.5.1 Escenario #1 Topología multisalto

Como se muestra en la figura 26 una topología multisalto conformada por cinco nodos sensores distribuidos linealmente, de tal forma que cada nodo represente un salto para llegar al resumidero el cual estará conectado una computadora.



**Figura 26** Escenario de topología de múltiples saltos



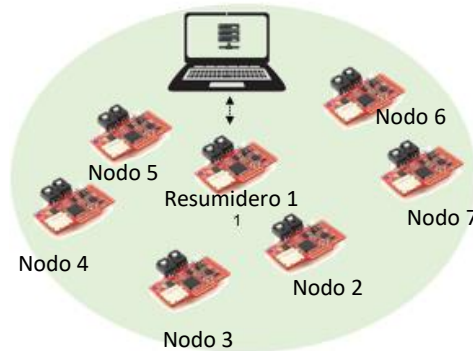
La tabla 10 indica la vecindad de los nodos dentro del rango de cobertura e interferencia para esta topología.

**Tabla 10** Distribución de topología multi saltos.

<i>Topología multi-salto</i>	Rango de transmisión	Rango de interferencia
<i>Nodo fuente</i>		Nodos vecinos dentro del rango
<i>Resumidero 1</i>	2	2,3
<i>Nodo 2</i>	1,3	1,3,4
<i>Nodo 3</i>	2,4	1,2,4,5
<i>Nodo 4</i>	3,5	2,3,5
<i>Nodo 5</i>	4	3,4

#### 4.5.2 Escenario #2 Topología de tipo malla a un solo salto

Este escenario mostrado en la figura 27, presenta una topología malla que contiene 6 nodos sensores y un resumidero distribuidos de tal forma que todos los nodos están dentro del rango de cobertura de lo demás nodos, teniendo que cada nodo esté a un salto del resumidero. Esta topología analizará el caso de tener múltiples usuarios en la red dentro de un mismo rango de cobertura.



**Figura 27** Escenario de topología de múltiples usuarios.

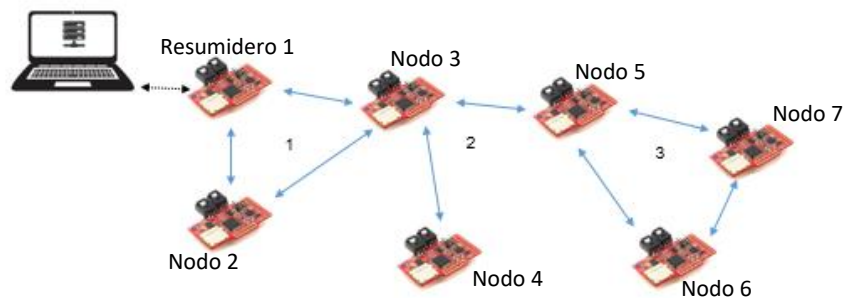
La tabla 11 indica la vecindad de los nodos dentro del rango de cobertura e interferencia para esta topología.

**Tabla 11** Distribución de topología un solo salto.

<i>Topología multi-usuario</i>	Rango de transmisión	Rango de interferencia
<i>Nodo fuente</i>	Nodos vecinos dentro del rango	
<i>Resumidero 1</i>	2,3,4,5,6,7	2,3,4,5,6,7
<i>Nodo 2</i>	1,3,4,5,6,7	1,3,4,5,6,7
<i>Nodo 3</i>	1,2,4,5,6,7	1,2,4,5,6,7
<i>Nodo 4</i>	1,2,3,5,6,7	1,2,3,5,6,7
<i>Nodo 5</i>	1,2,3,4,6,7	1,2,3,4,6,7
<i>Nodo 6</i>	1,2,3,4,5,7	1,2,3,4,5,7
<i>Nodo 7</i>	1,2,3,4,5,6	1,2,3,4,5,6

#### 4.5.3 Escenario #3 Topología combinada múltiples usuarios – múltiples saltos

Como se observa en la figura 28 se presenta una representación de un escenario aleatorio donde se distribuyen los nodos de cierta forma que existan usuarios dentro del mismo rango de cobertura y se presenten saltos para llegar al resumidero. Esta topología consta de 6 nodos sensores y un resumidero localizados a lo largo de 3 saltos.

**Figura 28** Escenario de topología de múltiples usuarios y múltiples saltos.

La tabla 12 indica la vecindad de los nodos dentro del rango de cobertura e interferencia para esta topología.

**Tabla 12** Distribución de topología multi usuario y muti salto.

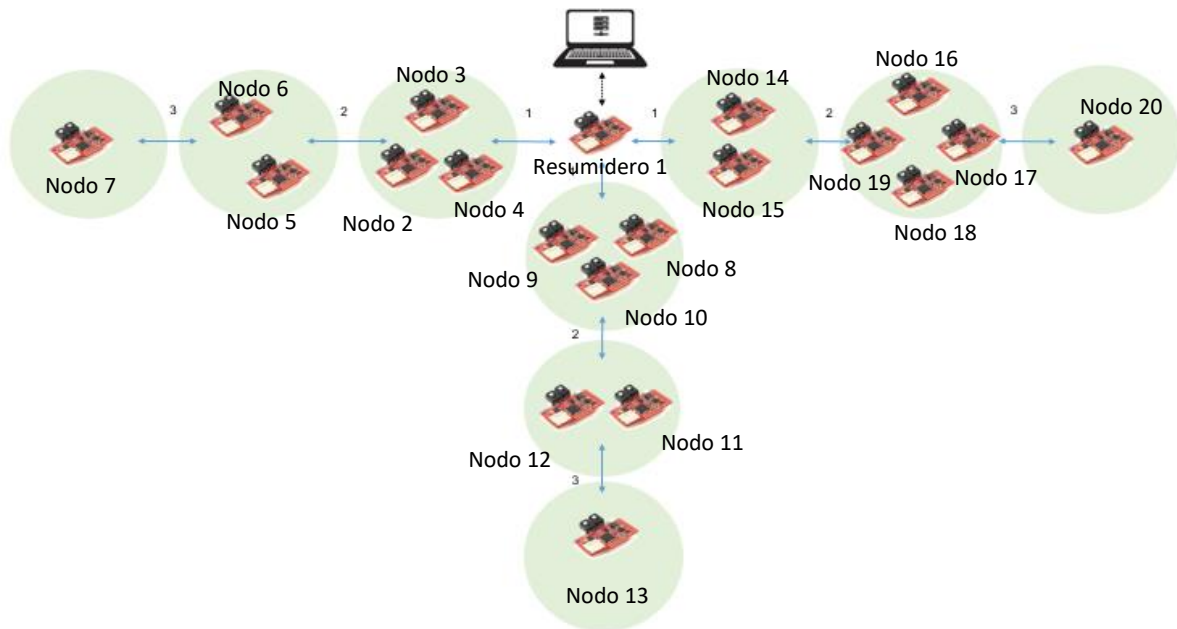
<i>Topología multi-salto, multi-usuario</i>	Rango de transmisión	Rango de interferencia
<i>Nodo fuente</i>	Nodos vecinos dentro del rango	
<i>Resumidero 1</i>	2,3	2,3,4,5
<i>Nodo 2</i>	1,3	1,3,4,5
<i>Nodo 3</i>	1,2,4,5	1,2,4,5,6,7
<i>Nodo 4</i>	3	1,2,3,5,6,7
<i>Nodo 5</i>	3,6,7	1,2,3,4,6,7
<i>Nodo 6</i>	5,7	1,3,4,5,7
<i>Nodo 7</i>	5,6	3,4,5,7

#### 4.5.4 Escenario #4 Topología basada en una aplicación

Este escenario está propuesto para albergar una red que consta de una cantidad de 19 usuarios y un resumidero distribuidos a lo largo de 6 saltos, de tal forma que presente características de múltiples usuarios y múltiples saltos, donde la distribución muestra una topología general encontrada en un escenario de monitoreo de adulto mayor, como el que se encuentra en un albergue de cuidado de ancianos. Para generar dicha topología se realizó un análisis de los diferentes centros de cuidado del adulto mayor en la ciudad de Ensenada Baja California, México para observar características inherentes: como la cantidad de ancianos promedio que albergan, la localización de cada anciano dentro del edificio, el número de ancianos promedio por cuarto, y la localización de la central de emergencias que cada centro presenta, para así genera la topología mostrada en la figura 29.

Esta topología servirá para evaluar el comportamiento de la formación de la red al tener una cantidad considerable de usuarios en una red que tiene una aplicación específica, y el desempeño que brinda al encontrar los usuarios distribuidos a lo largo de una cantidad considerable de saltos.

La tabla 13 indica la vecindad de los nodos dentro del rango de cobertura e interferencia para esta topología.



**Figura 29** Escenario de topología de aplicación de monitoreo y cuidado de adulto mayor.

**Tabla 13** Distribución de topología aplicación.

<b>Topología aplicación</b>	<b>Rango de transmisión</b>	<b>Rango de interferencia</b>
<i>Nodo fuente</i>	Nodos vecinos dentro del rango	
<i>Resumidero 1</i>	2,3,4,8,9,10,14,15	2,3,4,5,6,8,9,10,11,12,14,15,16,17,18,19
<i>Nodo 2</i>	1,3,4,5,6	1,3,4,5,6,7,8,9,10,11,12,14,15
<i>Nodo 3</i>	1,2,4,5,6	1,2,4,5,6,7,8,9,10,11,12,14,15
<i>Nodo 4</i>	1,2,3,5,6	1,2,3,5,6,7,8,9,10,11,12,14,15
<i>Nodo 5</i>	2,3,4,6,7	1,2,3,4,6,7,8,9,10
<i>Nodo 6</i>	2,3,4,5,7	1,2,3,4,5,7,8,9,10
<i>Nodo 7</i>	5,6	2,3,4,5,6,7
<i>Nodo 8</i>	1,9,10,11,12	1,2,3,4,5,6,9,10,11,12,13,14,15,16,17,18,19
<i>Nodo 9</i>	1,8,10,11,12	1,2,3,4,5,6,8,10,11,12,13,14,15,16,17,18,19
<i>Nodo 10</i>	1,8,9,11,12	1,2,3,4,5,6,8,9,11,12,13,14,15,16,17,18,19
<i>Nodo 11</i>	8,9,10,12,13	2,3,4,8,9,10,12,13,14,15
<i>Nodo 12</i>	8,9,10,11,13	2,3,4,8,9,10,11,13,14,15
<i>Nodo 13</i>	11,12	8,9,10,11,13
<i>Nodo 14</i>	1,15,16,17,18,19	1,2,3,4,8,9,10,11,12,15,16,17,18,19,20
<i>Nodo 15</i>	1,14,16,17,18,19	1,2,3,4,8,9,10,11,12,14,16,17,18,19,20
<i>Nodo 16</i>	14,15,17,18,19,20	1,8,9,10,14,15,17,18,19,20
<i>Nodo 17</i>	14,15,16,18,19,20	1,8,9,10,14,15,16,18,19,20
<i>Nodo 18</i>	14,15,16,17,19,20	1,8,9,10,14,15,16,17,19,20
<i>Nodo 19</i>	14,15,16,17,18,20	1,8,9,10,14,15,16,17,18,20
<i>Nodo 20</i>	16,17,18,19	14,15,16,17,18,19

## 4.6 Validación del simulador

Con el objetivo de determinar la fiabilidad de los resultados obtenidos por el simulador Cooja, es necesario realizar una validación en base a los mecanismos utilizados para la transmisión de información de acuerdo con el estándar IEEE 802.15.4. Por ello en este trabajo de tesis el simulador Cooja es validado por medio de una implementación de hardware y comparada con los resultados de (Reyes-Cruz, 2017), como se observa en la figura 30

En (Reyes-Cruz, 2017) se realizó la validación de la capa IEEE 802.15.4 del simulador OMNET++ en el framework enfocado a redes inalámbricas y móviles “MiXiM” (Kodali y Malothu, 2016). Esto mediante la comparativa del modelo de simulación de la capa MAC del protocolo IEEE 802.15.4 con el modelo analítico de (Casillas-Gaxiola, 2012) y una cama de pruebas experimental.



**Figura 30** Enfoque de validación del simulador Cooja.

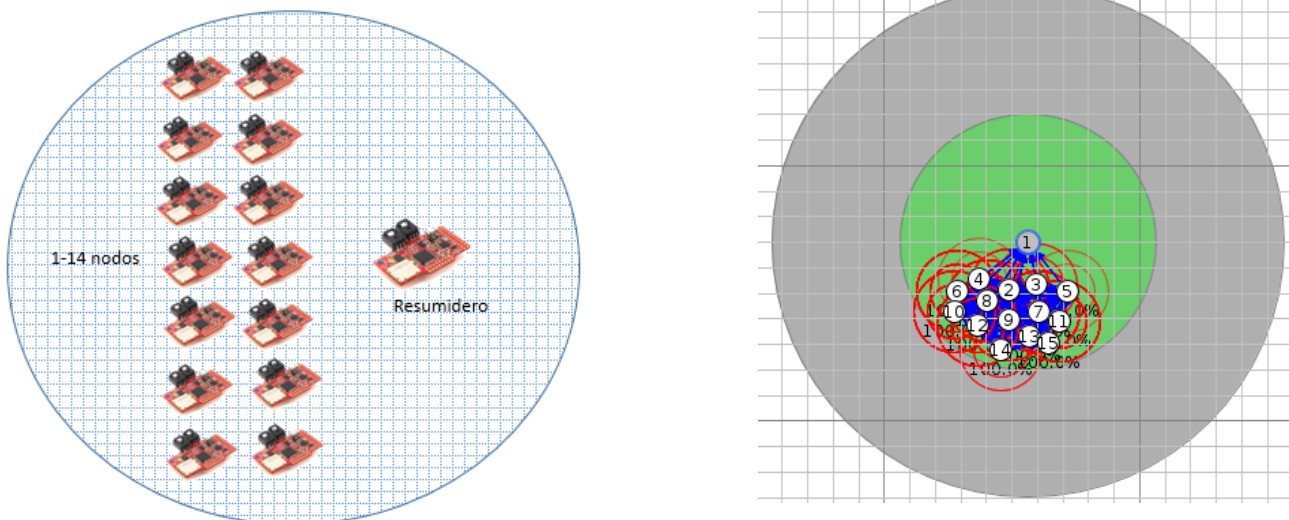
### 4.6.1 Escenario de simulación de validación

La tabla 14 muestra las configuraciones hechas a los parámetros del mecanismo de CSMA a implementarse en el simulador COOJA y OMNET++.

**Tabla 14** Parámetros de configuración del mecanismo de CSMA no ranurado.

Parámetro	Valor	Descripción
<i>aUnitBackoffPeriod</i>	20 símbolos	El número de símbolos que forman el periodo de tiempo básico usado por el algoritmo de CSMA/CA
<i>macMaxFrameRetries</i>	3	El número máximo de reintentos permitidos después de una transmisión fallida
<i>macMaxCSMABackoffs</i>	4	El número máximo de valor desistimiento que el algoritmo de CSMA /CA intenta antes de declarar una transmisión fallida en el canal
<i>macMinBE</i>	3	El valor mínimo del exponente de back off del algoritmo CSMA/CA
<i>macMaxBE</i>	5	El valor máximo del exponente de back off
<i>aMaxPHYPacketSize</i>	127 bytes	Máximo tamaño de PSDU que la capa física puede aceptar

El escenario de simulación que se utilizó para validar la capa MAC del modelo de Contiki OS dentro del simulador COOJA se muestra en la figura 31. Se presenta una red de topología tipo estrella en donde el número de nodos transmisores variará de 1 a 14 nodos todos dentro de su rango de cobertura y un nodo resumidero. La evaluación consiste en realizar un total de 10 simulaciones por cada escenario de cantidad de nodos transmisores, dando un total de 140 simulaciones con tiempo de simulación de 600 segundos cada una.



**Figura 31** Escenario para validación por medio de simulador.

Cada uno de los escenarios toma las siguientes consideraciones:

- Todos los nodos se escuchan entre ellos, esto es, cuando un nodo está transmitiendo cualquier otro nodo que evalúe el canal detectará que está ocupado, evitando así que ocurra el fenómeno del nodo oculto.
- No existen pérdidas de propagación debido a que los nodos se encuentran lo suficientemente cerca del nodo resumidero, ni tampoco de interferencia ya que no existe alguna red externa generando interferencia. Se toma en consideración el modelo de propagación UDGM que considera el modelo de Friis.
- La movilidad de los nodos es nula, cada nodo se encuentra estático para que no exista alguna alteración extra.

Los parámetros definidos para el desarrollo de la simulación están contenidos en la tabla 15.

**Tabla 15** Parámetros de configuración para validación por simulador.

<b>Parámetro</b>	<b>Valor</b>
<b>Tiempo de simulación</b>	600s
<b># de simulaciones por nodo</b>	10
<b># Total de simulaciones</b>	140
<b>Mote utilizado</b>	Z1 Zolertia
<b>Modelo de propagación</b>	UDGM
<b>Capa de red</b>	Rime Protocol
<b>Capa Mac</b>	CSMA/CA-modo sin Beacon habilitado
<b>Capa RDC</b>	nullrdc
<b>Capa Física</b>	IEEE 802.15.4
<b>Tamaño de Paquete (con encabezado MAC)</b>	31 y 115 Bytes
<b>Tasas de transmisión</b>	12 kbps

La métrica considerada para la evaluación es la tasa de paquetes perdidos como se observa en la ecuación 10, definida como el complemento del número de paquetes recibidos en el resumidero entre el número total de paquetes generados por los nodos transmisores.

$$P_l = 1 - \left( \frac{\sum_{j=1}^Q P_{r_j}}{\sum_{i=1}^N P_{t_i}} \right) \quad (10)$$

Donde:

- $P_l$  tasa de paquetes perdidos.
- $P_{r_j}$  paquetes recibidos por el j-esimo resumidero.
- $P_{t_i}$  paquetes transmitidos por i-esimo nodo.
- $Q$  Número de resumideros en la red.
- $N$  Número de nodos transmisores en la red.

El perfil de tráfico utilizado en la comparativa de los resultados de (Reyes-Cruz, 2017) y Cooja se muestran en la tabla 16.

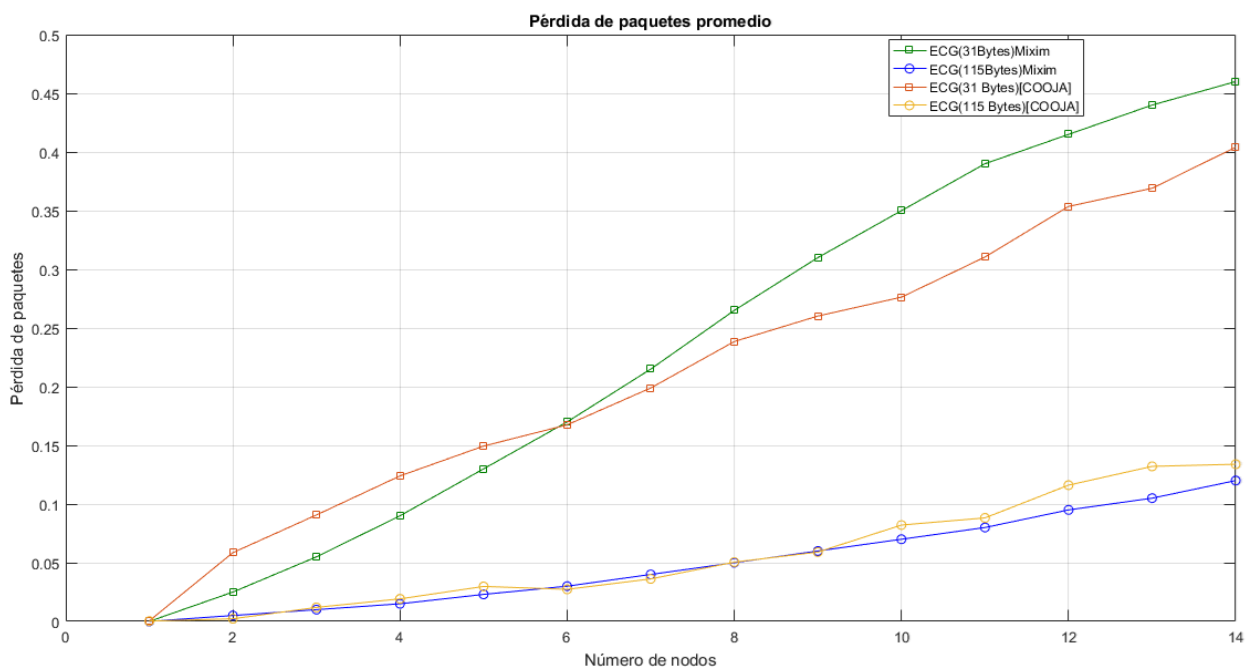
**Tabla 16** Análisis de empaquetamiento para validación del simulador.

<b>Tasa de 12kbps</b>	<b>Tamaño de paquete 31 bytes</b>	<b>Tamaño de paquete de 115 bytes</b>
<i>Payload</i>	18 bytes	102 bytes
<i>Tasa de envío de paquetes</i>	83.33 paquetes/segundo	14.70 paquetes/segundo
<i>Encabezado MAC</i>	13 bytes	13 bytes
<i>Sobre encabezado MAC</i>	8.66 kbps ~ 72 %	1.52 kbps~12 %
<i>Tráfico total por nodo</i>	20.66 kbps	13.524 kbps

Con los datos mostrados en la tabla 16 y los resultados mostrados en la figura 32 se puede apreciar que el método de empaquetado grande presenta mejores resultados al tener una pérdida de paquetes de alrededor del 14 por ciento cuando se presenta una distribución de nodos grande, como es el caso de tener 14 usuarios en la red, en comparación al método de empaquetado pequeño que presenta alrededor de un 40 por ciento de pérdidas para el mismo caso.



El método de empaquetado pequeño presenta mayor contención del canal por la cantidad de paquetes intercambiados para suplir la tasa demandada, conforme vaya aumentando el número de usuarios en la red la contención será mayor resultando en mayor probabilidad de encontrar el medio ocupado al evaluarlo, por consecuencia se presenta mayor cantidad de paquetes descartados por rebasar el límite de intentos establecido por la capa MAC, como se aprecia para el caso del empaquetamiento de 31 Bytes en la figura 32.



**Figura 32** Comparativa de pérdida de paquetes promedio de entre simulador Omnet++ y Cooja.

Conforme a la comparación de los resultados entre los simuladores, se puede apreciar que ambos casos presentan la misma tendencia, conforme aumenta el número de nodos en la red la tasa de pérdida de paquetes es mayor debido a la contención del canal generado. Para un empaquetamiento grande (115 Bytes) la tasa de pérdida de paquetes es muy similar variando un poco al aumentar el número de usuarios. Así mismo para un empaquetamiento pequeño (31Bytes) la relación entre pérdidas de paquetes varía alrededor de 2.7 por ciento en promedio de las muestras observadas, variación que puede considerarse poco significativa y que pudiera afinarse al aumentar el número de simulaciones realizadas.

## 4.6.2 Comparativa con una cama de pruebas

Se realizó también una comparativa con el simulador Cooja y una implementación de cama de pruebas. Para el simulador se utilizarán los nodos de Zolertia Z1, cuyos valores de operación están mostrados en la tabla 17. Estos nodos se eligieron por tener características más similares a los nodos disponibles en el laboratorio, los Sensor Tag CC2650. Es importante mencionar que existen diferencias entre los Zolertia Z1 y los Sensor Tags CC2650 en términos de consumo de energía y de capacidad del microprocesador. No obstante, la implementación en los Sensor Tag CC2650 se utilizó para validar la capa física y de acceso al medio del simulador, las cuales son compartidas por ambos tipos de nodos. Por lo que se considera que es válido el uso de los nodos Zolertia Z1 para la evaluación.

### 4.6.2.1 CC2560STK SensorTag.

La tarjeta de desarrollo CC2650 SensorTag, figura 33a), desarrollado por Texas Instruments, es un dispositivo IoT que incluye 10 sensores micro-electro-mecánicos de baja energía entre los que destacan una unidad inercial compuesta por acelerómetro, giróscopo y magnetómetro, además de un sensor de temperatura del objeto y ambiente, sensor de humedad y sensor de luminiscencia. La tarjeta de desarrollo está basada en el microcontrolador ARM Cortex M3 dentro de un circuito integrado CC2650 haciendo al dispositivo multi-estándar en la banda de 2.4 GHz al utilizar tres tecnologías de comunicación, Zigbee, 6LoWPAN y Bluetooth Low Energy. Este dispositivo es alimentado por una batería tipo CR2032 de 3 voltios. EL CC2650 SensorTag tiene soporte para ser utilizado con el sistema operativo de Contiki, al ser accesible para cargar las aplicaciones de comunicación desarrolladas por el usuario, como por ejemplo la utilización del estándar de IEEE 802.15.4, en una programación desarrollada en lenguaje C.

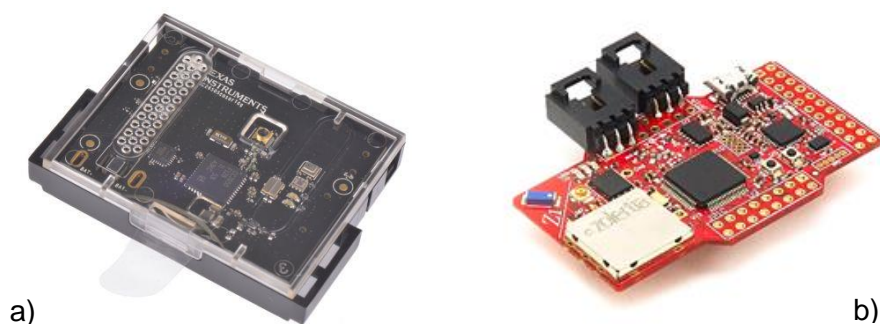
### 4.6.2.2 Zolertia Z1

El módulo Z1 de Zolertia figura 33 b) es una plataforma de desarrollo con propósito general para WSN, compatible con la familia de nodos Tmote con mejoras en el desempeño del nodo mediante la flexibilidad de agregar una combinación de fuentes de energía, sensores y conectores. Z1 presenta soporte para emplear los sistemas operativos de código abierto usados por la comunidad, como Tiny OS y Contiki Os.

La pila de protocolos de red permite incluir el estándar 6LoWPAN y su modo inalámbrico de baja energía cumple con las especificaciones de los protocolos IEEE 802.15.4 y Zigbee. La arquitectura principal del Z1 está basada en la familia de microcontroladores y radio transceptores de Texas Instruments: MSP430 y CC2420 respectivamente, esto lo hace compatible con diferentes nodos basados en la misma arquitectura. Debido a que el simulador Cooja no tiene compatibilidad de utilizar la tarjeta de desarrollo SensorTag CC2650 se decidió utilizar para el simulador el mote Z1 de Zolertia, que presenta características similares a las del SensorTag y que el simulador Cooja puede ofrecer. Una comparación de los recursos de las tarjetas se muestra en la tabla 17.

**Tabla 17** Características de tarjetas de desarrollo para evaluación.

<b>Característica</b>		<b>CC2650 SensorTag</b>	<b>Z1 Zolertia</b>
<i>Habilitación en Cooja</i>		No	SI
<i>Módulo de comunicación</i>		2.4 GHz IEEE 802.15.4 , 6LoWPAN, Zigbee	2.4 GHz IEEE 802.15.4 , 6LoWPAN, Zigbee
<i>Radio</i>		CC2620	CC2420
<i>MCU</i>		ARM Cortex M3	MSP430f2617
<i>RAM y ROM</i>		RAM=20Kb ROM=128kb	RAM= 8kb ROM=92kb
<i>Número de sensores</i>		10	2
<b>Condiciones típicas</b>	<b>Modo de operación</b>		
MCU encendido	Radio Rx	5.9 mA	18.8 mA
MCU encendido	Radio Tx	6.1 mA	17.4 mA
MCU <i>idle</i>	Radio apagado	550 $\mu$ A	0.1 $\mu$ A
MCU <i>standby</i>	Radio apagado	1 $\mu$ A	0.5 $\mu$ A
Voltaje		3 v	3 v

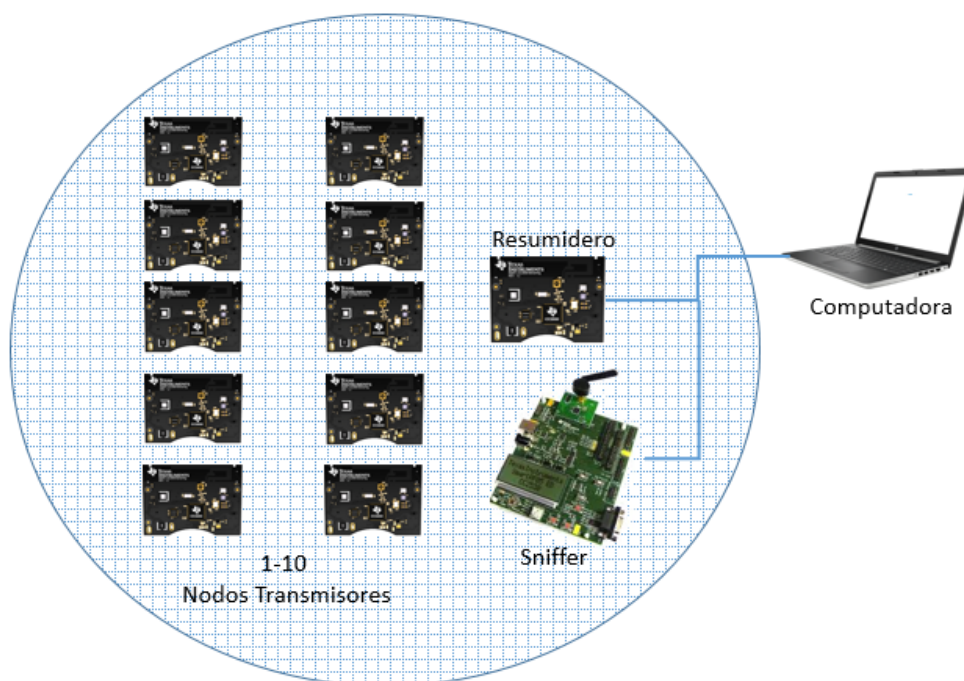


**Figura 33** Dispositivos utilizados para validación a) Sensortag CC2650 b) Z1 Zolertia.

#### 4.6.2.3 Cama de pruebas y configuraciones para validación por implementación.

El esquema utilizado para realizar las pruebas se observa en la figura 34. Se presenta una cama de pruebas compuesta por 10 nodos transmisores SensorTag y un resumidero, donde cada configuración variará la cantidad de nodos transmisores en la red de 1 a 10 nodos. El resumidero se encuentra conectado mediante un puerto serie a una computadora para obtener las métricas de paquetes recibidos. Para observar los paquetes enviados al canal radio, se utilizó la tarjeta de desarrollo SmartRF05 de Texas Instruments como *sniffer* mediante el programa “Packet Sniffer” desarrollado por dicha empresa.

Debido a que se observó en la sección anterior un buen desempeño a la configuración de tasa de paquetes baja, se optó por implementar el perfil de tráfico presentado en (Mendívil-Gastélum, 2017) dentro de sus pruebas de cobertura. Dicho trabajo toma en cuenta el sistema de monitoreo y localización de adultos mayores con detección de caídas. El empaquetado que presenta conforma un paquete de tamaño 30 bytes de información enviado a una tasa de 1 Hz durante un tiempo de prueba de 5 minutos, como se puede apreciar una tasa de paquete de datos pequeña. Se realizaron 10 pruebas por cada variación de configuración del número de nodos transmisores teniendo un total de 100 experimentos.



**Figura 34** Escenario de validación por implementación.

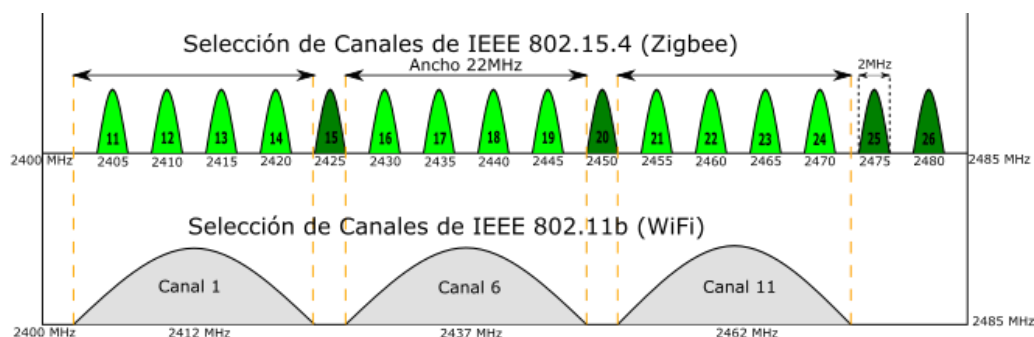
Las configuraciones hechas en Contiki Os son las mismas que se utilizarán para el simulador y son mostradas en la tabla 18:

**Tabla 18** Parámetros de configuración para evaluación de implementación.

Parámetro	Valor	
Número de nodos	1-10	
Duración de prueba	300 segundos	
Tamaño de carga útil	30 bytes	
Tasa de trasmisión	1 paquete por segundo	
Mote utilizado en simulador	Z1 Zolertia	
Mote utilizado en implementación	Sensortag CC2650	
Velocidad de los nodos	Estático	
Modelo de Propagación	Modelo de Friis / Pérdidas por propagación	
Protocolo PHY	IEEE 802.15.4	
Protocolo MAC	CSMA/CA sin beacon habilitado	
Protocolo de Ruteo	Rime	
Frecuencia de transmisión	2.4GHz	
Canal de transmisión	25 [2.475 GHz]	
Parámetros de configuración CSMA	macMaxFrameRetries	3
	MacMaxCSMABackoffs	4
	macMinBE	3
	macMaxBE	5
	aMaxPHYPacketSize	127 bytes

Durante la evaluación de la implementación se realizaron algunas observaciones a considerar:

- En un ambiente no controlado, los dispositivos SensorTag son susceptibles a la interferencia que se genera en el rango de operación de otras tecnologías, como WiFi, debido a que operan en la banda ISM de 2.4Gz, como se observa en la figura 35. Esto ocasiona que se generen colisiones causadas por transmisiones externas a la cama de pruebas propuesto.

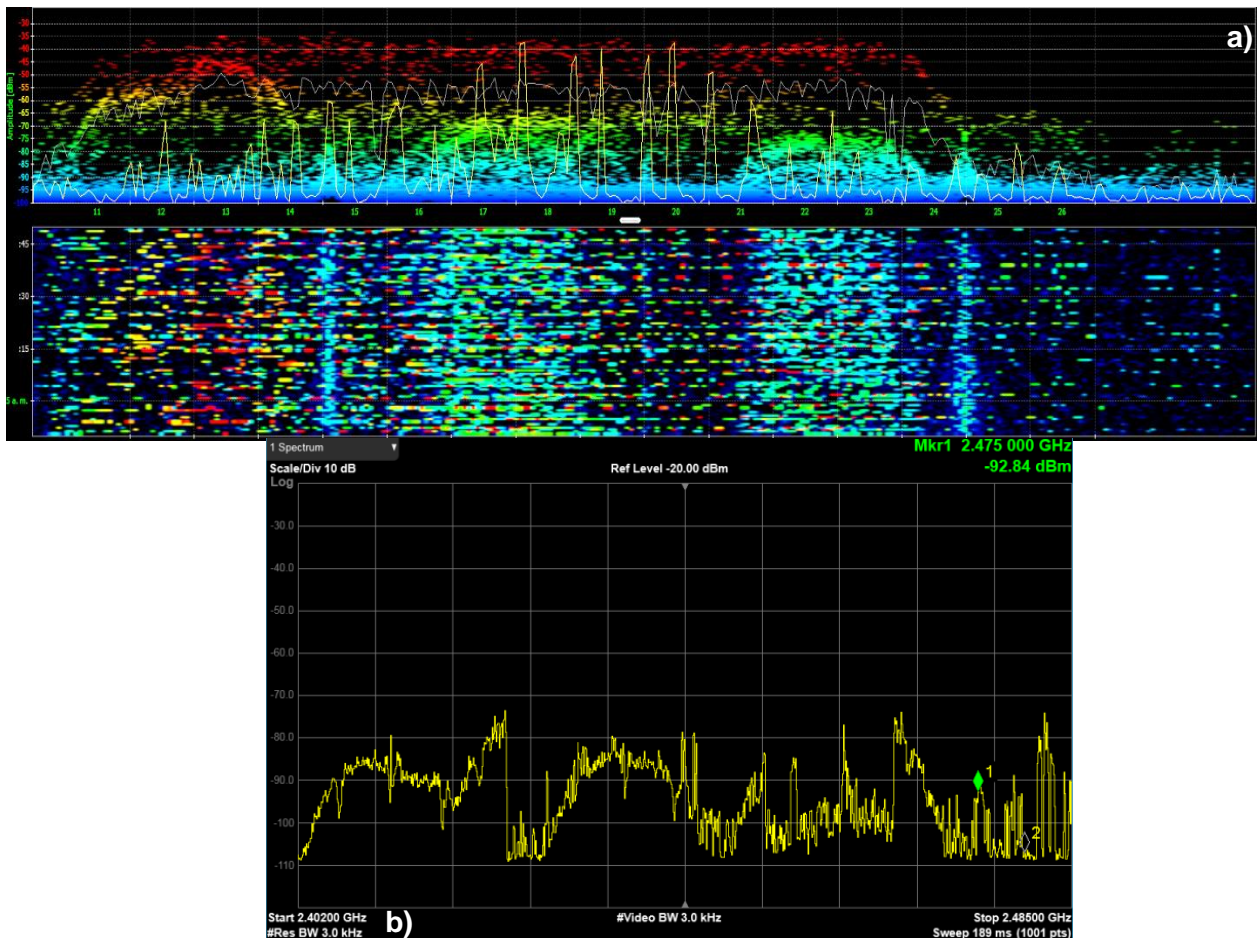


**Figura 35** Distribución de canales para tecnología IEEE 802.15.4 e IEEE 802.11.

A raíz de esto se hizo un análisis del medio para observar cómo se encontraba el canal. Dicho análisis fue llevado a cabo mediante dos equipos para corroborar las condiciones del canal.

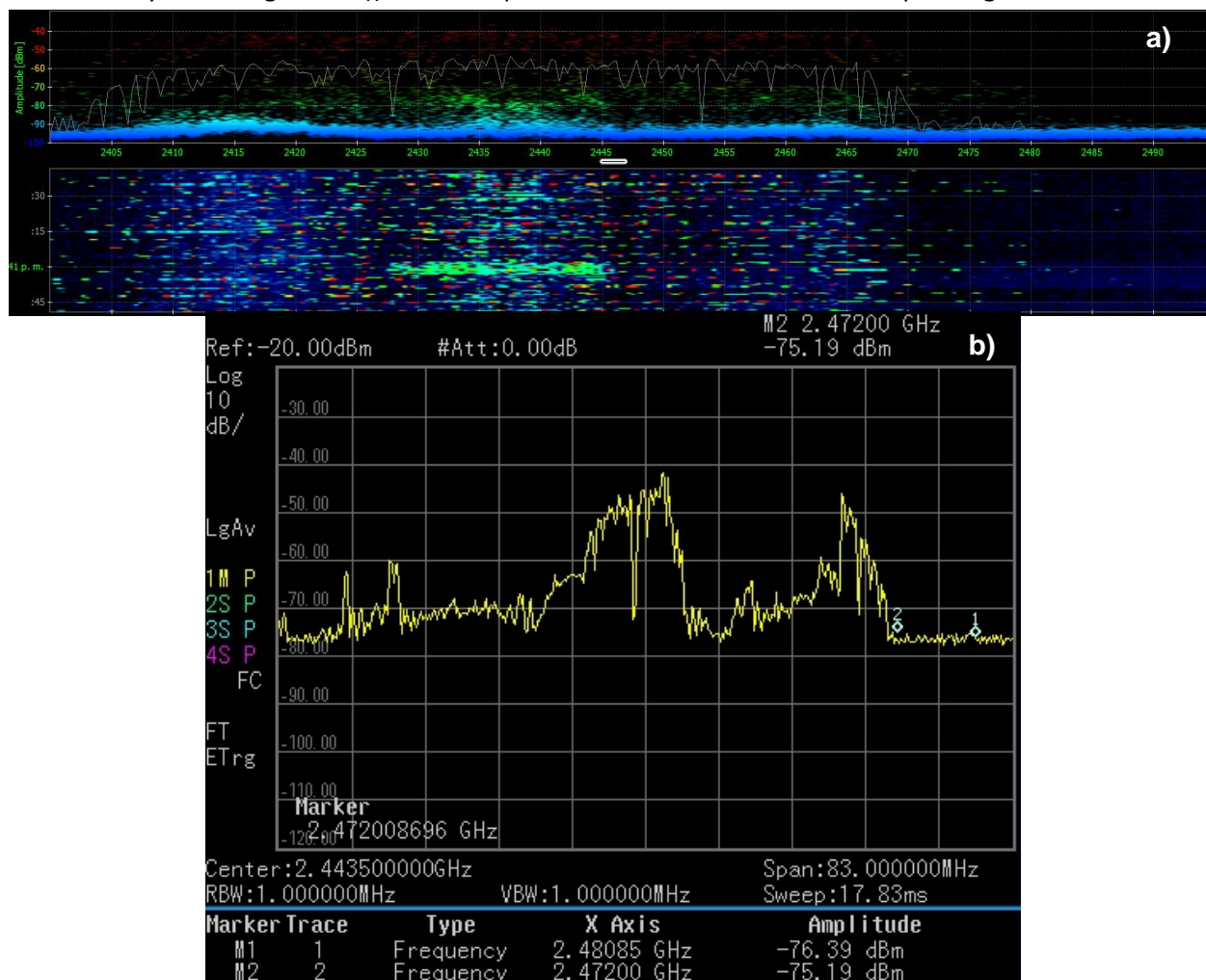
- Un analizador de espectros de la marca Keysight modelo N9040B con un rango de frecuencias de operación de 3 a 50 GHz, situado con una frecuencia de inicio a 2.402 GHz y una frecuencia de paro de 2.485 GHz
- Un analizador de espectros portátil con interfaz gráfica por computadora denominado Wi Spy de la marca Metageek que permite observar el medio inalámbrico dentro del rango de frecuencias de operación de WiFi y Zigbee (2402 -2480 MHz)

Como se observa en la figura 36 se pueden apreciar los lóbulos de operación de los canales principales de la tecnología WiFi, resultando que el medio se encuentre inundado por transmisiones a lo largo de los 16 canales que opera IEEE 802.15.4, incluso se llegan a observar algunas transmisiones en los canales 25 y 26 (frecuencia 2.475 y 2.480 respectivamente) donde los canales de WiFi no llegan a transmitir, pudiendo ser transmisiones de otra red IEEE 802.15.4 o Bluetooth.



**Figura 36** Espectro del rango de frecuencias de 2.402 a 2.485 GHz con interferencia en el canal a) Espectro obtenido mediante la herramienta WiSpy b) Espectro obtenido mediante analizador de espectros.

A raíz de esto se buscó tener un ambiente lo más controlado posible, para evitar interferencias significativas que pudieran afectar la evaluación justa con el simulador, ya que el simulador no presenta la característica de tener interferencias por dispositivos con el protocolo IEEE 802.11. Las características del medio encontrado para realizar las pruebas de validación del simulador se encuentran en la figura 37 donde se observa que el canal de operación efectivo para trabajar es el canal 25 (mostrado entre el marcador 2 y 1 de la figura 37b)) debido a que las transmisiones externas son poco significativas.

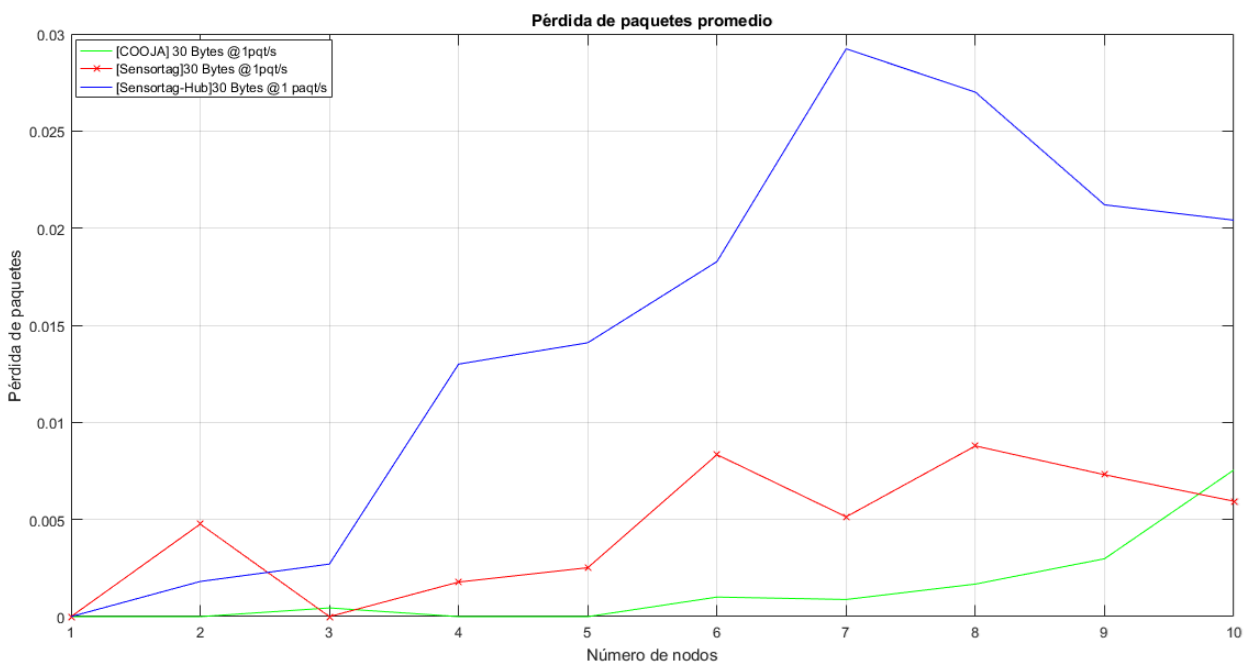


**Figura 37** Espectro del rango de frecuencias de 2.402 a 2.485 GHz sin interferencia en el canal 25 a) Espectro obtenido mediante la herramienta WiSpy b) Espectro obtenido mediante analizador de espectros.

Otra característica importante a considerar es el nivel de corriente para alimentar a los dispositivos Sensor Tags, la elección de una fuente de alimentación para estos dispositivos no debe tomarse de forma trivial, debido a que el nivel de energía con el que se alimente al nodo trae repercusiones en el desempeño del dispositivo para transmitir al no tener la suficiente carga eléctrica para operar el radio.

Se realizaron pruebas las cuales los diez nodos utilizados para la validación fueron suministrados de energía mediante un dispositivo de múltiples puertos USB tipo HUB modelo USB-HUB 10C2 de la marca Plugable. Dicho dispositivo tiene un suministro de corriente total en sus diez puertos de salida de  $900mA$ . Este valor de corriente no fue suficiente para dar operatividad a la cantidad de nodos conectados al dispositivo causando una disminución de corriente al operar al mismo tiempo la cantidad de nodos causando un decaimiento del desempeño de envío de paquetes.

Los resultados en la figura 38 muestran el desempeño de la pérdida de paquetes cuando se tiene un ambiente donde existe interferencia por otras tecnologías en el medio denotado por la línea de color rojo, la línea de color azul muestra los resultados al suministrar energía a los SensorTag con el dispositivo HUB, estos resultados se comparan con los resultados mostrados en la simulación denotado por la línea de color verde.



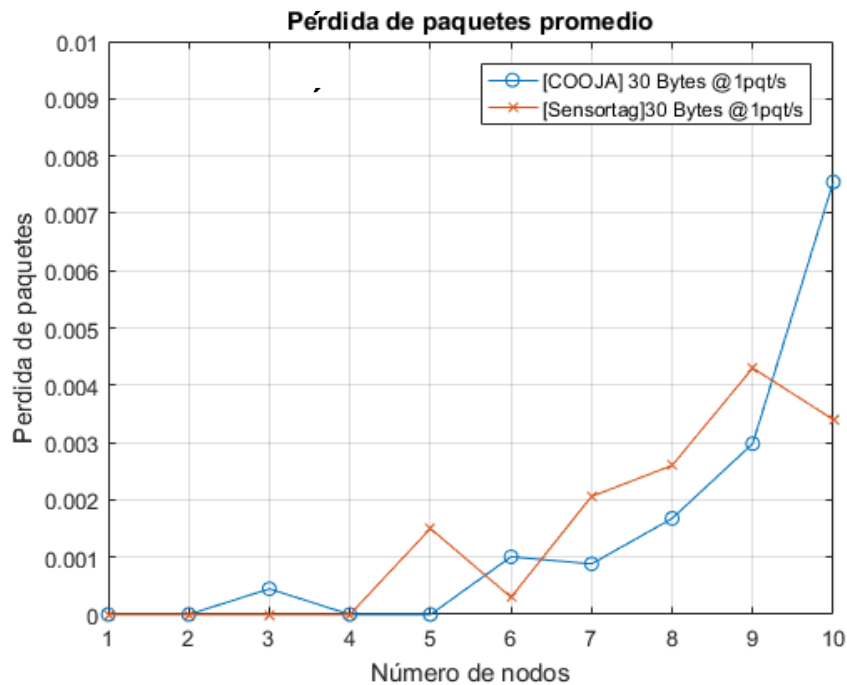
**Figura 38** Pérdida de paquetes en validación por implementación con escenarios de interferencia y administración de energía.

Se puede apreciar que el no tener en mente un control de la cantidad de energía administrada a los nodos sensores puede llegar a costar hasta un 3% en la tasa de paquetes perdidos esto es una diferencia de alrededor de 0.95% de los resultados obtenidos con el simulador. También el tener un ambiente con otras



redes operando en el mismo rango de frecuencias trae consigo hasta un 0.8% de pérdidas teniendo una diferencia con el simulador de alrededor de un 0.25%.

Los resultados de simulación muestran que se esperaba tener un máximo de un 0.0088% de pérdidas de paquetes en un escenario donde la tasa de transmisión es baja y una red poblada de alrededor a 10 nodos conteniendo por el medio. Tomando en cuenta las consideraciones observadas anteriormente y para realizar una validación lo más justa posible con el simulador, la figura 39 muestra los resultados de la validación del simulador por medio de una implementación dentro de un ambiente lo mayormente posible controlado con el uso de baterías nuevas y un canal libre de interferencia de otros dispositivos como es el canal 25. Tener un ambiente controlado no es una tarea trivial por las condiciones del medio. En esta figura se muestran los resultados de pérdida de paquetes promedio conforme la cantidad de usuarios incrementa en un entorno de simulación comparado con el de implementación. La tendencia en la figura 39 muestra que los resultados obtenidos en la cama de pruebas y el simulador son muy similares teniendo una variación máxima entre ambos del 0.082%.



**Figura 39** Pérdida de paquetes en validación por implementación en un escenario lo mayormente posible controlado.

## 4.7 Metodología de evaluación

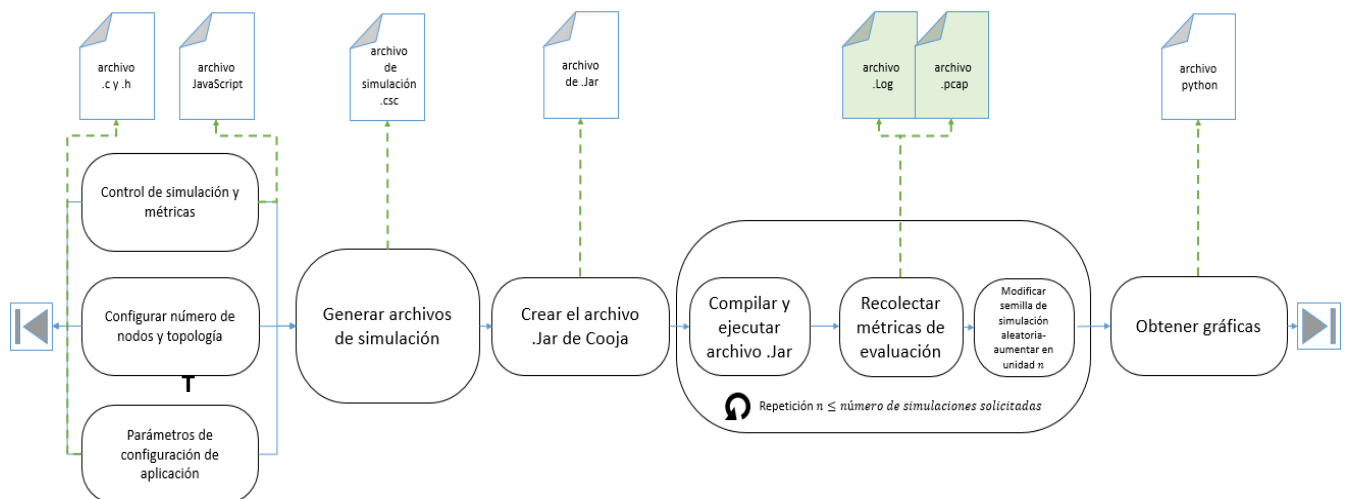
Las simulaciones en esta evaluación se basan en la capa de acceso al medio del estándar IEEE 802.15.4, particularmente el mecanismo CSMA/CA no ranurado y sin beacon habilitado

La figura 40 muestra el proceso de cómo fueron llevadas a cabo las simulaciones, comenzando con el proceso de asignar cada uno de los parámetros que se requieren para la configuración del escenario a evaluar.

En cada iteración se cambió la semilla y se generan dos tipos de archivos marcados en verde en la figura 40, estos archivos contienen la información para obtener las métricas evaluadas:

- Un archivo de texto con extensión .log contiene la información del tiempo de convergencia y consumo de energía de cada nodo.
- Un archivo con extensión .pcap almacena las capturas de tráfico de red como la información de cada paquete transmitido durante el desarrollo de la simulación, este archivo indica la cantidad de paquetes total que se utilizó para formar la red.

Por último, se generó código fuente en lenguaje Python para obtener el valor medio de las métricas de cada nodo sensor en la red. Las configuraciones generales para realizar la evaluación dentro del simulador están expuestas en la tabla 19.



**Figura 40** Modelo de ejecución de simulación y análisis.

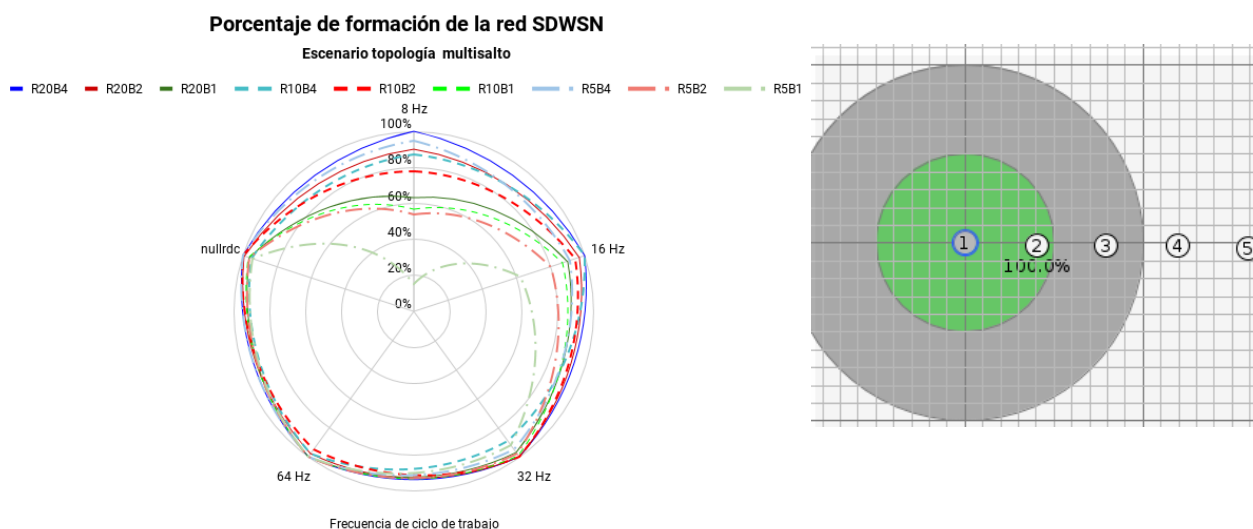
**Tabla 20** Parámetros de configuración de evaluación de redes SDWSN y RPL.

<b>Parámetros</b>	<b>Valor</b>		
Tiempo total de simulación	Variable		
Simulador	Cooja		
Rango de Transmisión	50 m		
Rango de Interferencia	100 m		
Modelo de propagación	UDGM Pérdidas por distancia		
Capa Física	IEEE 802.15.4		
Capa Acceso al Medio	CSMA		
Capa RDC en nodos sensores	ContikiMac y null-rdc		
Frecuencia de ciclo de trabajo	8,16,32,64 Hz		
Capa RDC en resumidero(nodo sensor #1)	null-rdc		
Modelo de enrutamiento	RPL (modo Storing)	SDN-WISE	
Configuraciones de envío de paquetes de control	$T_{DIO\ min} = 4\ seg$ $T_{DIO\ max} = 17\ min$	$T_{Report} = 20\ seg$	$T_{Beacon} = 4, seg$
			$T_{Beacon} = 2, seg$
			$T_{Beacon} = 1, seg$
		$T_{Report} = 10\ seg$	$T_{Beacon} = 4, seg$
			$T_{Beacon} = 2, seg$
			$T_{Beacon} = 1, seg$
		$T_{Report} = 5\ seg$	$T_{Beacon} = 4, seg$
			$T_{Beacon} = 2, seg$
			$T_{Beacon} = 1, seg$
# de Simulaciones	100		
Distancia por nodo	20m-40m		
Cantidad de nodos resumidero	1		
Cantidad de nodos sensores	4-6-19		
Movilidad	Estática		

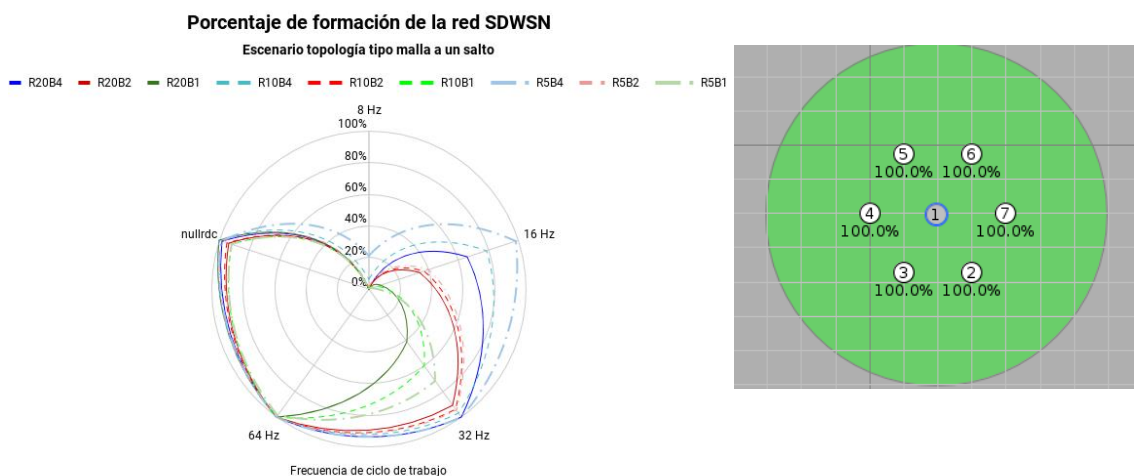
El mecanismo de distribución de rutas por parte del resumidero no contempla enviar un paquete de reconocimiento ACK del paquete Openpath. Como se mencionó en la sección 4.4.1 en este trabajo se considera que la red SDWSN converge cuando cada nodo en la red obtiene el Openpath correspondiente. Sin embargo al enviar el paquete Openpath es posible que llegue a perderse debido a colisiones. Además, al agregar técnicas de ciclo de trabajo del radio como ContikiMac se genera una transmisión repetitiva del paquete dependiendo de la frecuencia  $f_{RDC}$  (ver sección 3.3.3). Esto genera mayor contención en el canal resultando en una mayor probabilidad de pérdida del paquete Openpath. Si el paquete Openpath se pierde durante el proceso de entrega de rutas, el nodo destino no puede formar su tabla de flujo, por lo tanto se considera que la red no converge. A continuación se presenta un análisis del número de veces en las que la red llega a converger en los escenarios propuestos en la sección 4.5.

#### 4.7.1 Efecto de ciclo de trabajo sobre formación de red en SDWSN

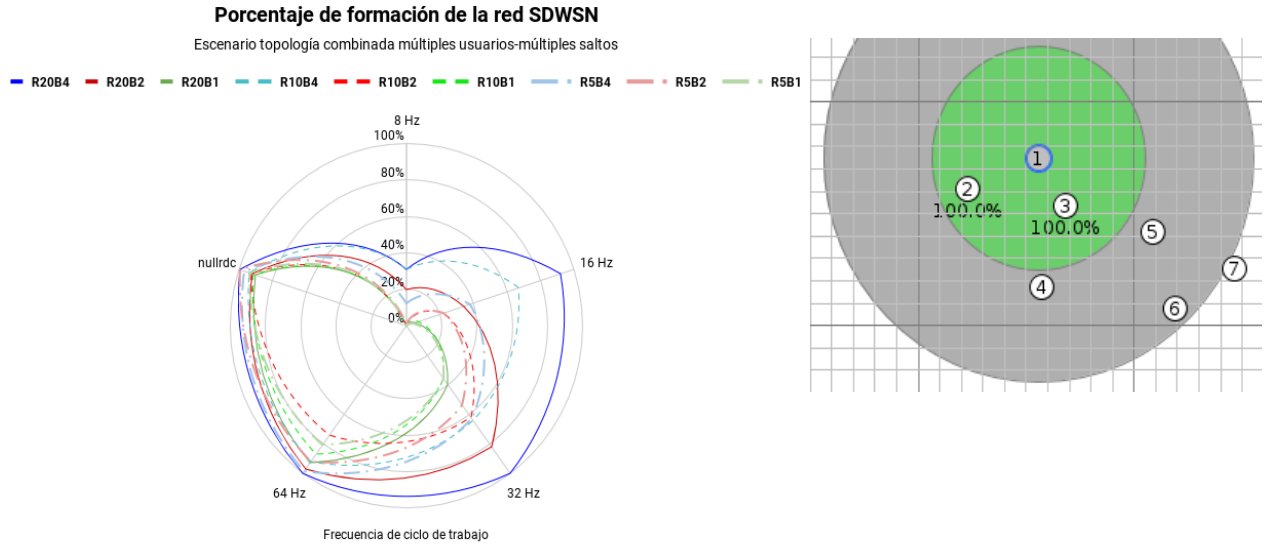
Este análisis consideró las características de configuración propuestas en la tabla 19, se realizaron un total de 100 experimentos (con semillas pseudoalatorias distintas) por cada configuración de periodicidad de paquete Beacon y Report. Esto resulta en un total de 1800 simulaciones realizadas tomando en cuenta las 4 topologías propuestas a analizar. Para diferenciar entre las configuraciones de periodicidad de los paquetes Report y Beacon, se propone la nomenclatura  $R\alpha B\gamma$ , donde  $R$  indica la configuración de periodicidad del paquete Report mediante el valor tiempo  $\alpha$  y  $B$  indica la configuración de periodicidad del paquete Beacon mediante el valor de tiempo  $\gamma$ , p.ej.  $R10B4$  (periodo de paquete Report a 10 segundos, periodo de paquete Beacon a 4 segundos). Las figuras siguientes muestran la distribución de éxitos de convergencia para cada una de las configuraciones propuestas en la tabla 19.



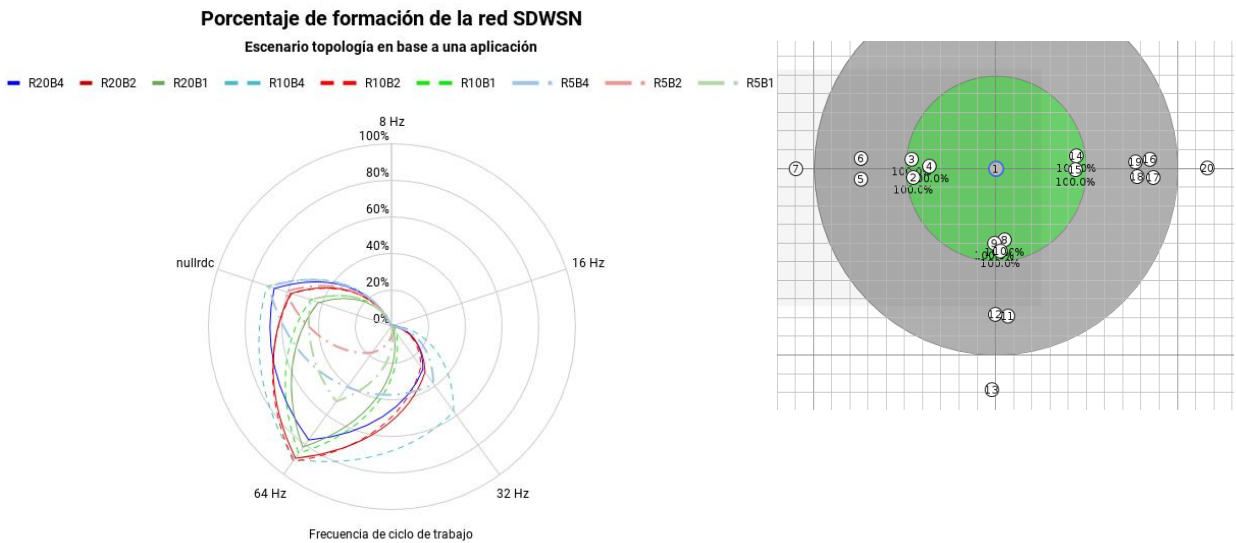
**Figura 41** Distribución de convergencia de la red SDN-WISE para escenario multisalto.



**Figura 42** Distribución de convergencia de la red SDN-WISE para escenario con escenario de múltiples usuarios.



**Figura 43** Distribución de convergencia de la red SDN-WISE para escenario de múltiples saltos-múltiples usuarios.



**Figura 44** Distribución de convergencia de la red SDN-WISE para escenario de aplicación de monitoreo adulto mayor (múltiples usuarios).

Las siguientes conclusiones se obtienen de las figuras de la distribución de convergencia de la red (fig 41-44):

- Tener el radio encendido la mayor parte del tiempo trae mayor probabilidad de que la red converja sin problemas considerando la periodicidad de los paquetes de control. El desempeño de

formación de la red disminuye conforme la frecuencia de conmutación de encendido del radio disminuye, p.ej. a 8 Hz.

- Una mayor cantidad de paquetes de control (Beacon y Report) disminuye el número éxitos en la convergencia de la SDWSN. Esto se puede asociar con una mayor contención en el acceso al medio. Una frecuencia alta de envío de paquetes Beacon y Report puede hacer que la red se forme en menor tiempo, pero cuando se aplica un ciclo de trabajo de baja frecuencia provoca que la red no converja.
- La cantidad y la distribución de nodos en saltos de la red impacta en la disseminación correcta de las rutas, se observa que al tener una mayor cantidad de saltos la probabilidad de que el paquete se pierda en el camino es mayor, como se observa en la figura 46 donde la topología de red de aplicación está conformada por 6 saltos.
- Fenómenos como colisiones causadas por nodo oculto, interferencia causada entre transmisiones repetitivas de otros paquetes de control y saturación en la cola para colocar la transmisión a un nodo vecino en la capa MAC , son causas observadas por las cuales un paquete Openpath llega a perderse o ser descartado en el camino, evitando que alcance a su destino

A partir de las conclusiones observadas se hace notar que es necesario agregar un mecanismo de reconocimiento ACK a la red SDN-WISE para ayudar a que la red converja cuando se haya perdido un paquete Openpath por algún motivo.

#### **4.7.2 Mecanismo de re-distribución de paquete Openpath**

Para realizar una comparación justa con la tecnología de RPL del costo de formar enteramente la red se propone diseñar un mecanismo complementario al protocolo de SDN-WISE que considere las observaciones realizadas en el análisis de distribución de éxito de formación de la red. Mediante las consideraciones observadas del análisis de la sección anterior, se diseñó un mecanismo simple, observado en el diagrama de flujos de la figura 45.

Este mecanismo se encarga de redistribuir correctamente la ruta, cuando sea requerido. Esto es que no afecte a las configuraciones que tienen un mayor porcentaje de probabilidad de que la red converja en su totalidad y provea un apoyo para las configuraciones en donde se observa que el paquete Openpath esté perdido en el camino.

La operación del mecanismo es el siguiente: Inicia en los nodos sensores, a partir de que transmiten su paquete Report al resumidero. Entre el resumidero y el controlador comienza el proceso llamado *Topology Management*, el tiempo que tarda en realizar este proceso fue medido y tarda en promedio 4 segundos, a partir de que el primer nodo entregó su paquete Report, para generar las rutas y distribuirlas. Se consideró un tiempo estimado de 5 segundos de espera en cada nodo después de enviar su paquete Report al término de este tiempo observará si ha aprendido una ruta, si aprendió la ruta se almacenará en la tabla de flujo e indicará que está listo para transmitir información. Si después de este periodo el nodo no recibió un paquete Openpath, enviará una solicitud hacia el controlador mediante un paquete Request, para que vuelva a distribuir la ruta. Esta solicitud se realizará de manera periódica hasta que el nodo haya recibido la ruta solicitada. Para evitar mayor contención en el canal y el uso excesivo de paquetes de control para obtener una ruta, se determinó un periodo de 15 segundos en el cual si el nodo no ha recibido la ruta la volverá a solicitar, así sucesivamente hasta que logre obtenerla. Este periodo fue propuesto tomando en cuenta varias pruebas con diferentes valores para determinar el tiempo mínimo efectivo para transmitir la solicitud de ruta al no haber recibido el paquete Openpath previamente. Una vez que el nodo haya recibido la ruta solicitada, la almacenará en su tabla de flujos para ser utilizada cuando se requiera, e indicará que estará listo para transmitir información. A su vez cuando el último nodo hoja en la red haya aprendido e indicado que está listo para transmitir información haciendo que todos los nodos en la red hayan indicado que se encuentran operables para transmitir se dará por convergida la red. El simulador Cooja establece por defecto un tiempo de paro *Timeout* para las simulaciones automáticas de 20 minutos. Si el periodo *Timeout* se cumple antes de que la red llegue a considerarse convergida, se determinará que la red no convergió y se dará por concluida la simulación.

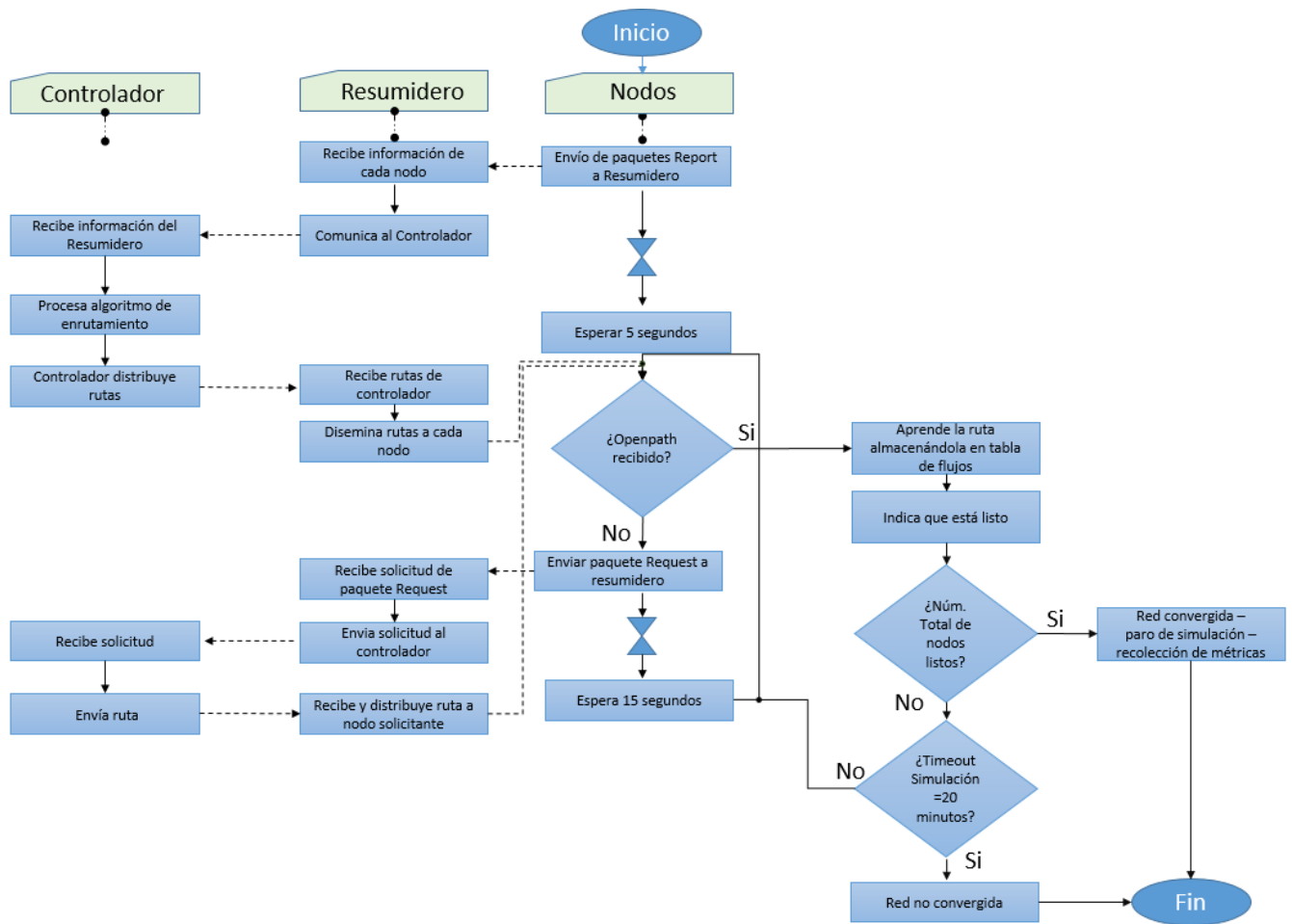


Figura 45 Algoritmo de redistribución de paquetes Openpath perdidos.



## Capítulo 5. Análisis del impacto del ciclo de trabajo en la formación de la red

---

### 5.1 Introducción

El ciclo de trabajo en las WSN ha sido usado ampliamente para reducir el consumo de energía en los nodos y prolongar su tiempo de vida en la red. Sin embargo aplicarlo puede traer un costo en términos del caudal eficaz o un mayor retraso en la transmisión (Charfi y Bouyahi, 2012; Carrano et al., 2014). En este capítulo se presentan los resultados obtenidos de la comparativa de los algoritmos de descubrimiento de topología de una red SDWSN y una red basada en RPL cuando se considera el ciclo de trabajo en el proceso de formación de una red. Se presenta un análisis de cada uno de los escenarios propuestos (ver Sección 4.5) para determinar cuáles son las condiciones y consideraciones de desempeño de formación de la red de cada paradigma.

Al revisar la literatura no se encontró ningún punto de referencia estandarizado de configuración de la red para evaluar el desempeño de formar una red con enrutamiento realizado por RPL o el enrutamiento realizado por la solución SDWSN, además de considerar el impacto del ciclo de trabajo durante este proceso.

### 5.2 Análisis del impacto del ciclo de trabajo en formación de la red SDWSN en comparación con la red RPL

Como previamente se ha indicado el objetivo de esta tesis es evaluar el proceso de formación de una red inalámbrica de sensores definida por software al agregar una técnica de ciclo de trabajo al radio y se hará una comparación con el proceso de formación del enrutamiento de facto de las WSN i.e. RPL. Los resultados de esta comparación están en términos del consumo de energía del radio transceptor, la cantidad de paquetes de control generados y el tiempo empleado para formar la red al punto que sea totalmente operable. Para evitar que la red SDWSN no converja se implementa el mecanismo mostrado en la sección 4.7.2 y poder así realizar una comparación con el enrutamiento RPL. Dicho mecanismo apoyará a que converja la red cuando el paquete Openpath (como se observa a frecuencias  $f_{RDC}$  bajas en la sección 4.7.1) es perdido solicitando periódicamente la ruta al controlador.

Las configuraciones de periodicidad de los paquetes de control se eligieron tomando en cuenta los trabajos previos de evaluación de esta tecnología. Las evaluaciones realizadas consideran diferentes periodos de paquetes Beacon y Report sin detallar cual configuración es la óptima para el proceso de formación de la red. Por ejemplo (Tsapardakis et al., 2018) utiliza un periodo de paquete Report de 20 segundos, en (Aregawi-Beyene, 2017) utilizan periodos de 20 segundos para los paquetes Report y de 5 segundos para los paquetes Beacon. Por su parte (Hieu et al., 2018) utiliza una configuración de periodo de paquete Report de 10 segundos y de Beacon de 5 segundos y (Buratti et al., 2016) utiliza una periodicidad de paquetes Report de 20 segundos con paquetes Beacon periódicos a 10 segundos.

Las combinaciones en las configuraciones de periodicidad de paquetes de control a evaluar son clasificadas en periodos de tiempo *largo*, *medio* y *corto* como se describe en la tabla 20:

**Tabla 21** Configuración de periodicidad de paquetes de control para red SDWSN.

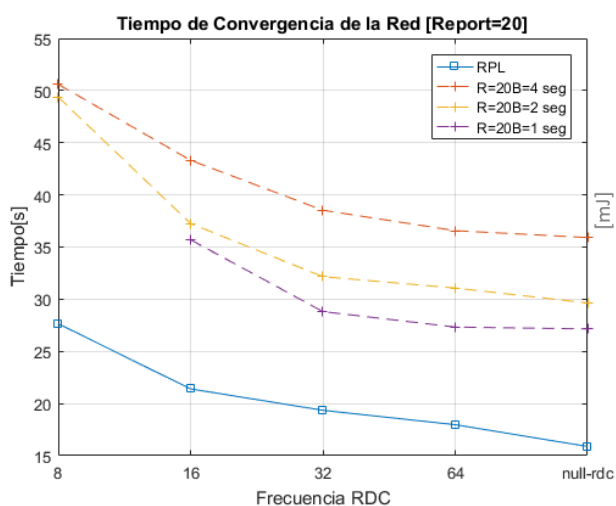
<i>Modo de configuración de periodicidad</i>	Periodo de paquete Report	Periodo de paquete Beacon	Nomenclatura $R\alpha B\gamma$
<i>Largo</i>	20 segundos	4 segundos	R20B4
		2 segundos	R20B2
		1 segundo	R20B1
<i>Medio</i>	10 segundos	4 segundos	R10B4
		2 segundos	R10B2
		1 segundo	R10B1
<i>Corto</i>	5 segundos	4 segundos	R5B4
		2 segundos	R5B2
		1 segundo	R5B1

Se comienza con una periodicidad de paquete Beacon de 4 segundos y de ahí se disminuye su valor, para realizar la comparativa con la periodicidad de paquetes DIO que utiliza RPL para formar la de red. Cada configuración de periodicidad de los paquetes de control de la red SDWSN es comparada con la configuración de los paquetes de control de RPL.

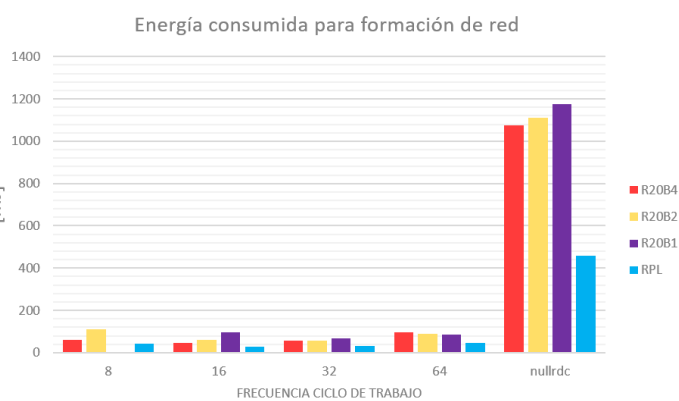
Las figuras 46 a la-80 muestran los resultados obtenidos en cada uno de los escenarios propuestos, posterior a ello se hace un análisis breve del comportamiento observado.

### 5.2.1 Análisis de resultados comparativos para escenario #1 Topología multisalto

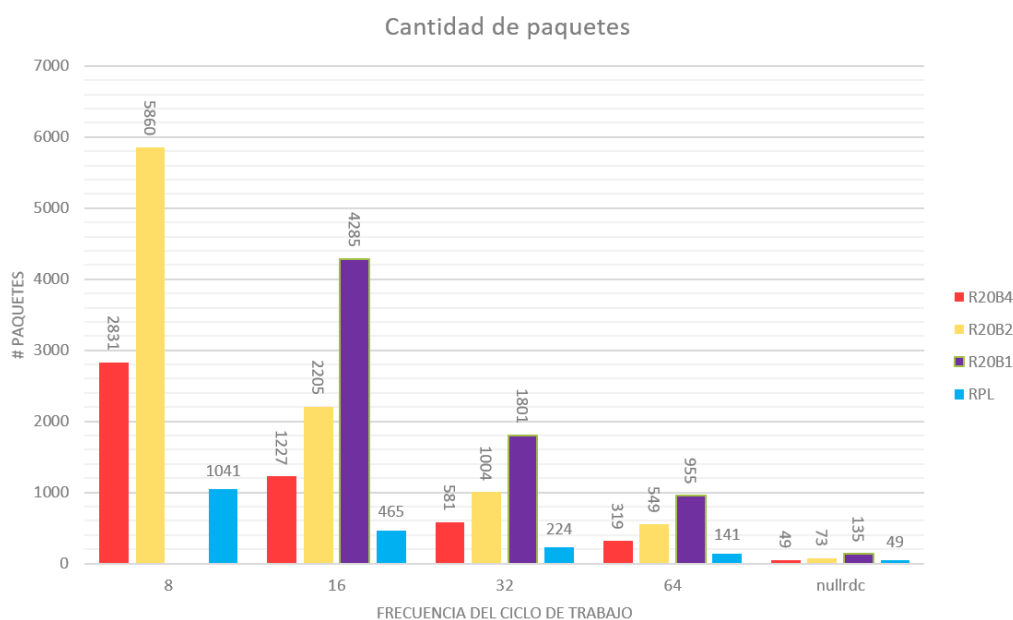
Los resultados mostrados en este apartado se obtienen utilizando la topología multisalto encontrada en la sección 4.5.1 y las configuraciones de periodicidad de paquetes de control de la tabla 20. Las figuras (46 - 48) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *largo*.



**Figura 46** Comparación del tiempo de convergencia. Topología multisalto, configuración R20By.



**Figura 47** Comparación energía consumida. Topología multisalto, configuración R20By.



**Figura 48** Comparación cantidad de paquetes de control. Topología multisalto, configuración R20By.

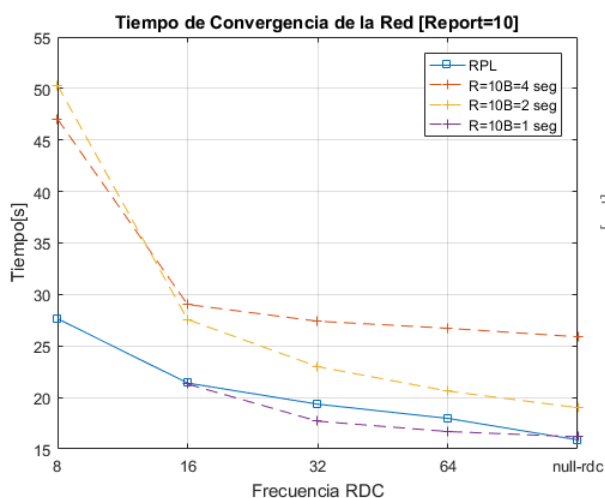
Se puede observar en las tres figuras, que conforme el ciclo de trabajo aumenta, el desempeño de ambos enfoques mejora. Sin embargo, se obtienen mejores métricas cuando se utiliza RPL para formar la red. Particularmente, el tiempo de convergencia de la red SDWSN mostrado en la figura 46, es hasta 2.26 veces mayor en comparación con el tiempo de formación de una WSN usando RPL. En la tabla 21 se muestra la diferencia de tiempo en segundos para formar la red en comparación con RPL. El impacto de utilizar una frecuencia  $f_{RDC}$  de 8 Hz a la configuración R20B1 hace que la red no converja incluso considerando el mecanismo de reconocimiento ACK para el paquete Openpath. Esto se puede explicar por la cantidad de tráfico generado al mandar cada segundo los paquetes Beacon, aunado de la cantidad de paquetes extras generados por utilizar esta frecuencia.

**Tabla 22** Diferencia de tiempo de convergencia de la red (Topología #1) entre configuración R20B $y$  y configuración de periodicidad en RPL.

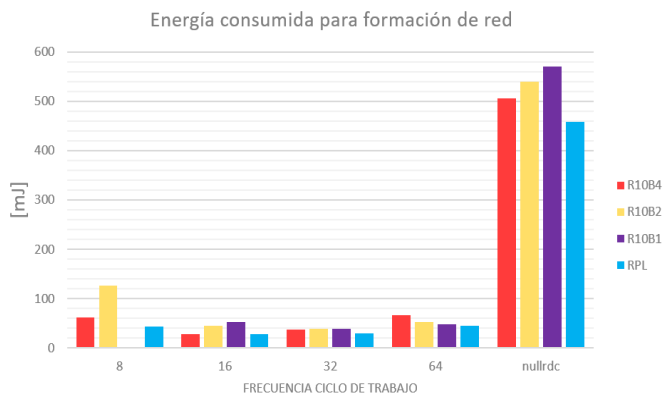
$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4	R20B2	R20B1
8Hz	22.954 seg	21.733 seg	----
16Hz	21.915 seg	15.857 seg	14.299 seg
32Hz	19.18 seg	12.81 seg	9.46 seg
64Hz	18.6 seg	13.102 seg	9.36 seg
nullrdc	20.025 seg	13.755 seg	11.265 seg

Formar una red SDWSN de este tipo y con estas configuraciones al no aplicar ciclo de trabajo trae consigo un gasto de hasta 2.4 veces más en el consumo de energía para formar la red en comparación a RPL. Como se observa en la figura 47 entre mayor frecuencia de envío de paquetes Beacon el consumo de energía es mayor. Aplicar técnicas de ciclo de trabajo a ambas redes trae consigo una disminución del consumo de energía del 93 % para formar la red eliminando el estado de *idle listening* en todas la frecuencias  $f_{RDC}$ . La comparación del consumo de energía para formar la red entre las configuraciones SDWSN y la configuración de RPL es de alrededor del 51%, siendo RPL el que consume menor energía para formar la red. La cantidad de paquetes de control (*overhead*) generados para formar la red SDWSN presenta la tendencia, de entre mayor frecuencia de generación de paquetes Beacon mayor cantidad total de *overhead*. Agregar ciclo de trabajo al radio impacta en la generación de paquetes de control, la cual está regida por la transmisión repetitiva del paquete. Al formar la red SDWSN con la configuración de periodicidad de paquetes *largo* con configuraciones de ciclo de trabajo la cantidad de overhead generado puede llegar a ser de hasta 9 veces mayor en comparación de la formación con RPL, siendo en todas las configuraciones R20B $y$  un overhead mayor a RPL, como se observa en la figura 48

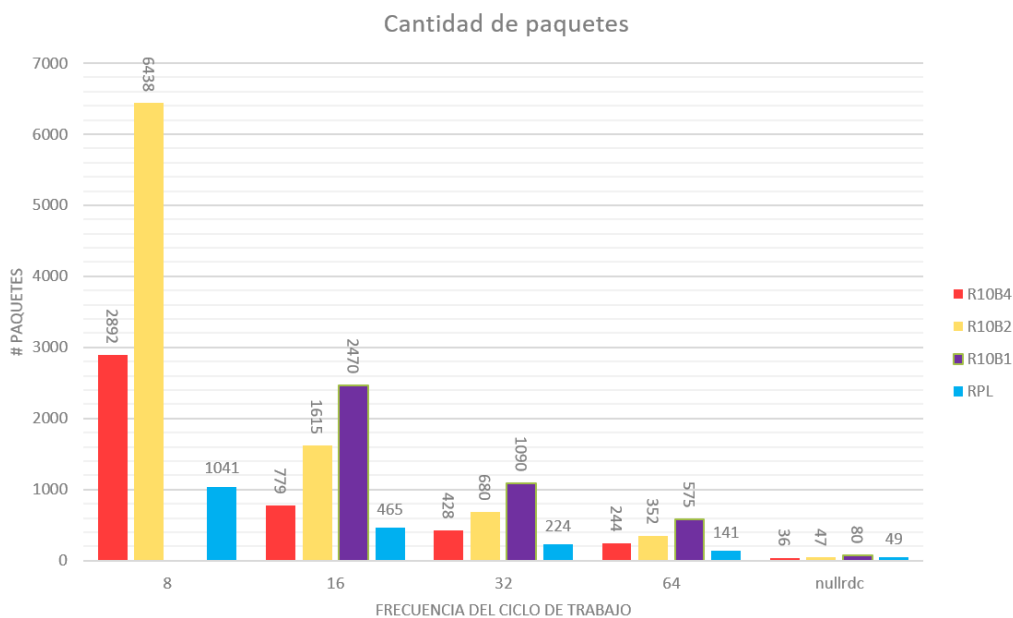
Las figuras (49 - 51) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *medio*.



**Figura 49** Comparación del tiempo de convergencia. Topología multisalto, configuración R10By.



**Figura 50** Comparación energía consumida. Topología multisalto, configuración R10By.



**Figura 51** Comparación cantidad de paquetes de control. Topología multisalto, configuración R10By.

Reducir el período de envío del paquete Report reduce el tiempo de convergencia de la red como se muestra en la figura 49. La configuración R10By muestra tener mejores resultados en comparación de RPL para formar la red en términos de tiempo de convergencia, esto es al configurar la periodicidad de los paquetes Beacon a 1 segundo a una  $f_{RDC}$  de 32 y 64 Hz se tiene menor tiempo de convergencia de la red, como se observa en la columna R10B1 de la tabla 22 la diferencia de tiempo de convergencia entre la red

SDWSN y RPL es negativa, lo cual indica que el tiempo de convergencia está a favor de la red SDWSN. Sin embargo como se mostró en el análisis de la configuración R20B $\gamma$  el impacto de utilizar una frecuencia  $f_{RDC}$  de 8 Hz en conjunto de una periodicidad de paquete Beacon a 1 segundo continua haciendo que la red no converja en su totalidad incluso considerando el mecanismo de reconocimiento ACK.

**Tabla 23** Diferencia de tiempo de convergencia de la red (Topología #1) entre configuración R10B $\gamma$  y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4	R10B2	R10B1
8Hz	19.393 seg	22.662 seg	----
16Hz	7.627 seg	6.167 seg	-0.101 seg
32Hz	8.054 seg	3.64 seg	-1.653 seg
64Hz	8.75 seg	2.65 seg	-1.28 seg
nullrdc	10.005 seg	3.125 seg	0.295 seg

La figura 50 muestra el consumo de energía empleado para formar la red SDWSN y RPL en comparación de la formación de la red cuando se aplica ciclo de trabajo al radio, este consumo de energía es medido al sumar la cantidad de energía empleada en los procesos de transmisión y recepción. Al aplicar técnicas de ciclo de trabajo se disminuye el estado de *idle listening* en el radio transceptor, conforme disminuye la frecuencia del ciclo de trabajo la cantidad de energía consumida en el proceso de recepción disminuye y la cantidad de energía consumida para el proceso de transmisión aumenta por la cantidad de paquetes de control generados. A lo largo de las configuraciones de periodicidad de paquetes Beacon el consumo de energía para el estado de recepción varía siendo mayor a ciertas frecuencias  $f_{RDC}$  y menor en otras, haciendo que el consumo de energía total se vea influenciado por esta cantidad de energía, como es el caso para la configuración R10B4 a una  $f_{RDC}$  de 64 Hz se presente un consumo de energía mayor en comparación a la energía consumida a una  $f_{RDC}$  de 16 Hz.

Converger la red en un menor tiempo trae beneficios en el consumo de energía como se observa en la figura 50. La disminución del consumo de energía se observa a simple vista al utilizar la configuración de nullrdc, el cual presenta una reducción del 51.9% en comparación de la configuración R20B $\gamma$ . La tabla 23 muestra la diferencia del porcentaje de energía presentado en la configuración de periodicidad de paquetes *medio* (R10B $\gamma$ ). En contraste con la configuración de periodicidad de paquetes *largo* (R20B $\gamma$ ). Sin embargo, a pesar de que la cantidad de energía reducida con la configuración de periodicidad de paquetes de control *medio*, el enrutamiento RPL forma la red con un consumo de energía menor al

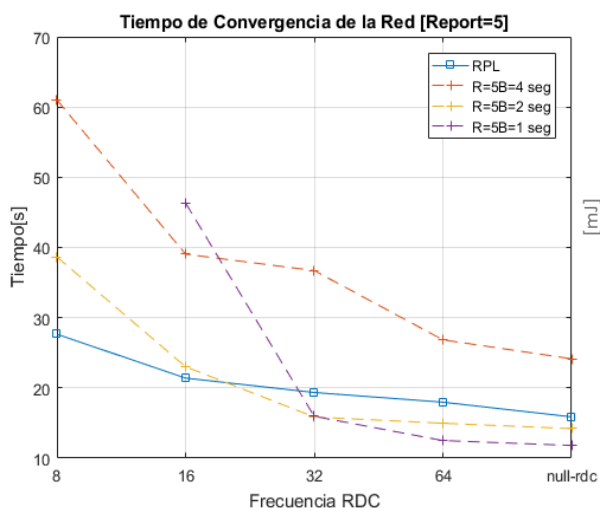
observado con el paradigma SDWSN. La reducción del consumo de energía al aplicar técnicas de ciclo de trabajo es significativa, teniendo una reducción de alrededor del 89% en ambas redes.

**Tabla 24** Diferencia de porcentaje de consumo de energía de la red (Topología #1) entre configuración R20B $\gamma$  y R10B $\gamma$ .

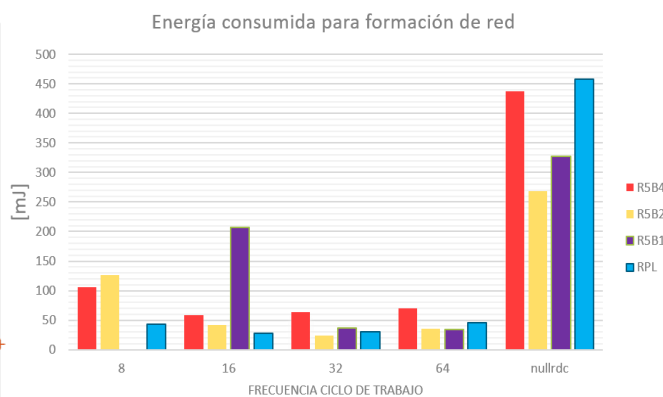
$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4-R10B4	R20B2-R10B2	R20B1-R10B1
8Hz	-4.20%	-13.04%	-----
16Hz	38.61%	25.37%	44.29%
32Hz	32.77%	32.61%	42.70%
64Hz	29.35%	39.17%	44.04%
Nullrdc	53.04%	51.40%	51.41%

Se observa en la tabla 23 que utilizar una  $f_{RDC}$  de 8Hz en combinación con una configuración de periodicidad de paquetes de control *medio* (casillas de color amarillo), presenta un incremento en lugar de traer una reducción en el consumo de energía en comparación con la configuración de periodicidad de paquetes de control *largo*.

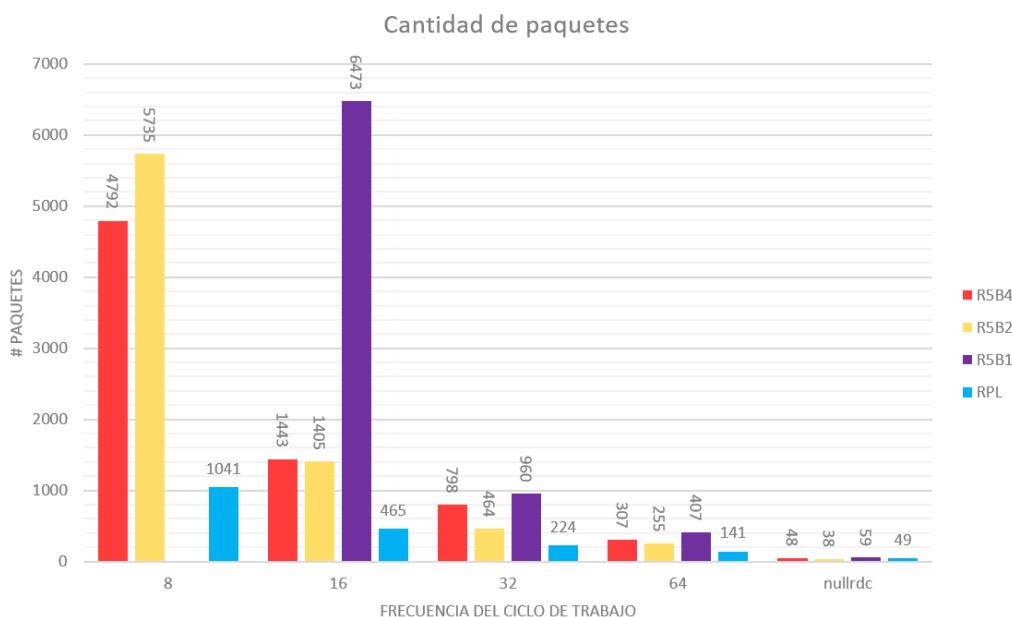
Como se observa en las columnas R10B4 y R10B2 de la tabla 22 del tiempo de convergencia de la red, para los casos donde se tiene una  $f_{RDC}$  de 8 Hz la diferencia del tiempo de convergencia de la red es mayor en comparación al de las demás configuraciones, esto repercute en la cantidad de energía consumida ya que al tomar más tiempo en formar la red, la cantidad de paquetes de control aumenta como se observa en la figura 51. Al disminuir el periodo de envío de paquetes Report la cantidad de paquetes para formar la red SDWSN al utilizar la configuración nullrdc disminuye como se observa en los periodos R10B4 y R10B2, presentando menor overhead en comparación de la formación con RPL. Al igual que en la configuración R20B $\gamma$  se observa el impacto sobre el overhead al agregar ciclo de trabajo, donde a menor frecuencia  $f_{RDC}$  mayor cantidad de overhead para formar la red. La cantidad de overhead generado para formar la red SDWSN al agregar ciclo de trabajo para la configuración de periodicidad paquetes de control *medio* puede llegar a ser de hasta 5 veces mayor en comparación de la formación con RPL, presentando en todas las configuraciones R10B $\gamma$  un overhead mayor a RPL. Las figuras (52 - 54) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *corto*.



**Figura 52** Comparación del tiempo de convergencia. Topología multisalto, configuración R5By.



**Figura 53** Comparación energía consumida. Topología multisalto, configuración R5By.



**Figura 54** Comparación cantidad de paquetes de control. Topología multisalto, configuración R5By.

Los resultados obtenidos en el tiempo de convergencia siguen siendo competitivos para RPL, donde las configuraciones R5B2 y R5B1 convergen en un menor tiempo en comparación de la configuración utilizada en RPL, esto se observa en la figura 52 para frecuencias  $f_{RDC}$  altas (32 y 64Hz) y nullrdc. Ambas configuraciones (R5B2 y R5B1) muestran un buen desempeño en términos de consumo de energía teniendo una disminución a RPL del 35% del consumo presentado. Dichas observaciones se pueden apreciar en la tabla 24 en la columna R5B2 y R5B1 donde se aprecia la diferencia de tiempo convergencia con valor negativo (indica una ventaja de tiempo) para la red SDWSN en comparación con RPL.



Para frecuencias  $f_{RDC}$  bajas (8, 16 Hz) se observa un incremento en el tiempo de convergencia esto es debido a la combinación de tener mayor flujo de paquetes de control en el medio como se observa en la figura 56, donde la cantidad de paquetes intercambiados aumenta significativamente para estas configuraciones, esto causa que la cantidad de colisiones y pérdida de paquetes se incremente, haciendo que se tome un mayor tiempo en distribuir correctamente todas las rutas en la red. Nuevamente se observa que la periodicidad de paquetes Beacon a 1 segundo a una frecuencia  $f_{RDC}$  de 8 Hz la red no converge totalmente incluso con el mecanismo de reconocimiento ACK implementado.

**Tabla 25** Diferencia de tiempo de convergencia de la red (Topología #1) entre configuración R5B $\gamma$  y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R5B4	R5B2	R5B1
8Hz	33.363 seg	11.003 seg	----
16Hz	17.647 seg	1.617 seg	24.937 seg
32Hz	17.36 seg	-3.49 seg	-3.39 seg
64Hz	8.88 seg	-3 seg	-5.46 seg
nullrdc	8.265 seg	-1.675 seg	-4.075 seg

Los resultados mostrados en la tabla 24 muestran una diferencia de tiempo ligeramente mayor en comparación de la configuración de periodicidad anterior (R10B $\gamma$ ), esto se debe a que la diferencia de tiempo presentada en frecuencias  $f_{RDC}$  bajas es aumentada por presentar una frecuencia mayor en la pérdida del paquete Openpath, haciendo que el mecanismo de reconocimiento ACK implementado opere por converger la red al solicitar un mayor número de veces la ruta. La diferencia de uso del mecanismo se puede apreciar en la figura 52, donde a frecuencias  $f_{RDC}$  altas el tiempo de convergencia de la red se muestra estable, en comparación de utilizar frecuencias  $f_{RDC}$  bajas, donde el tiempo de la red crece, generando mayor impacto al utilizar una menor periodicidad de paquetes Beacon, p.ej. R5B1. Se espera que al disminuir la frecuencia de ciclo de trabajo se provea una reducción al consumo de energía sin embargo, como se observa en la tabla 25 al combinar la frecuencia baja de ciclo de trabajo en combinación con un modo de periodicidad *corto* puede incrementar el consumo de energía para formar la red. Esto es observado en la figura 53 donde la frecuencia  $f_{RDC}$  de 64 Hz presenta una reducción del consumo de energía en comparación de RPL del 22%, sin embargo conforme va disminuyendo la frecuencia  $f_{RDC}$  (8, 16 Hz) el consumo de energía de las configuraciones de la red SDWSN va aumentando, como se observa en la tabla 24 casillas amarillas. Dicha tabla presenta la diferencia de porcentaje del consumo de energía en comparación con la configuración R10B $\gamma$ .

Al aplicar frecuencias de ciclo de trabajo bajas el enrutamiento RPL consume menor energía y tiene mayor estabilidad en el tiempo de formación de la red en comparación de la formación de la red SDWSN. El impacto que presenta el ciclo de trabajo sobre el *overhead* generado para formar la red SDWSN continua siendo mayor en comparación de la formación en RPL. El impacto es más notable cuando se presentan frecuencias bajas donde puede llegar a ser hasta 13 veces mayor en comparación con la cantidad de paquetes generados para formar la red RPL, como se observa en la figura 54.

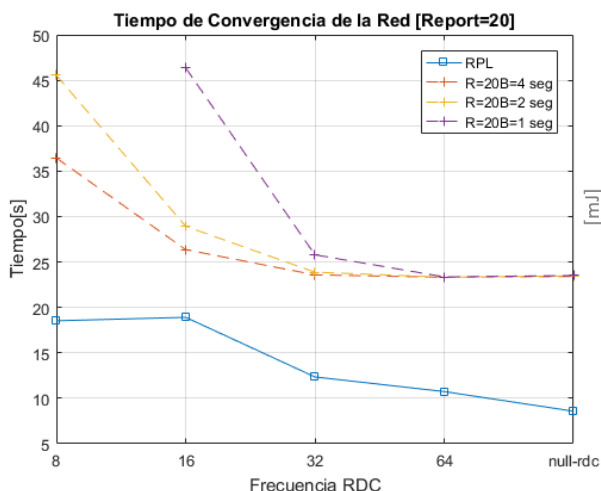
La configuración de periodicidad de paquetes R5B4 demuestra ser la combinación de configuraciones con peor desempeño para formar la red SDWSN en términos de tiempo de convergencia y energía en comparación de RPL, como se observa en las figura 52 y 53.

**Tabla 26** Diferencia de porcentaje de consumo de energía de la red (Topología #1) entre configuración R10B $\gamma$  y R5B $\gamma$ .

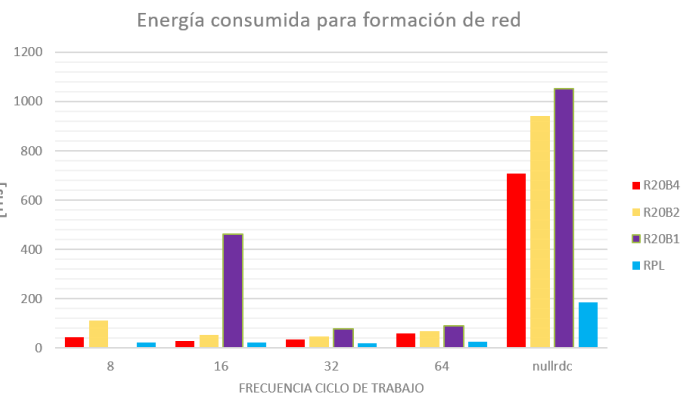
$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4-R5B4	R10B2-R5B2	R10B1-5B1
8Hz	-41.24%	-0.15%	----
16Hz	-51.20%	5.82%	-74.47%
32Hz	-40.09%	37.13%	6.93%
64Hz	-4.98%	32.42%	28.45%
nullrdc	13.52%	50.40%	42.67%

### 5.2.2 Análisis de resultados comparativos para escenario #2 Topología multi-usuario

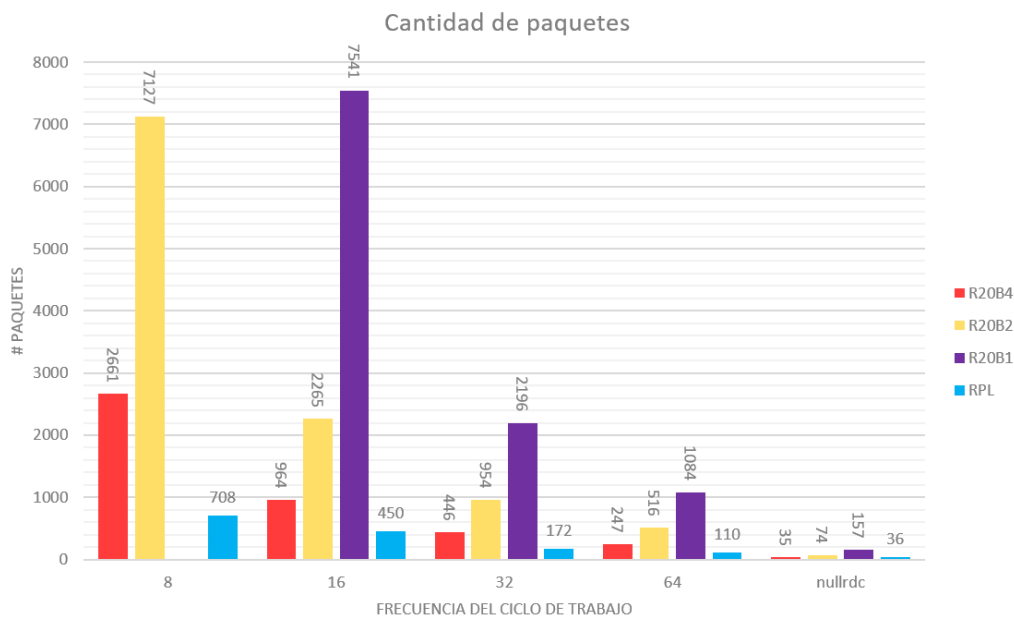
Los resultados mostrados en este apartado se obtienen utilizando la topología tipo malla a un solo salto encontrada en la sección 4.5.2 y las configuraciones de periodicidad de paquetes de control de la tabla 20. Las figuras (55 - 57) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *largo*.



**Figura 55** Comparación del tiempo de convergencia. Topología multi-usuario, configuración R20By.



**Figura 56** Comparación energía consumida. Topología multi-usuario, configuración R20By.



**Figura 57** Comparación cantidad de paquetes de control. Topología multi-usuario, configuración R20By.

La figura 55 muestra el desempeño de formación de las redes en términos de tiempo de convergencia. Se puede observar que el desempeño de ambas redes es diferente, RPL presenta un mejor desempeño al converger la red alrededor de la mitad del tiempo de lo que ocupa SDWSN

El tiempo de formación de la red SDWSN al utilizar frecuencias  $f_{RDC}$  altas (32, 64 Hz) muestra una uniformidad a lo largo de las configuraciones de periodicidad de los paquetes Beacon, esto se debe por la distribución de los nodos en la red, al encontrarse a un salto del resumidero, basta con que un paquete Report sea entregado al resumidero para formar una visualización de la red, así la distribución del paquete Openpath se facilita para llegar a su destino, además la contención generada a esas frecuencias es mínima tal que no afecta al desempeño de formación de la red. Sin embargo cuando se aplican frecuencias  $f_{RDC}$  bajas (8, 16 Hz) el tiempo de convergencia aumenta perdiendo la uniformidad presentada. Esto sucede debido a la contención existente para estas frecuencias, haciendo que la red se vuelva intermitente y aumente el uso del mecanismo de ACK para redistribuir la ruta, resultando en un aumento del tiempo de formación de la red. El impacto de utilizar una frecuencia  $f_{RDC}$  de 8 Hz a la configuración R20B1 hace que la red no converja incluso considerando el mecanismo de reconocimiento ACK para el paquete Openpath. La configuración de periodicidad de paquetes R20B $\gamma$  es poco competitiva con RPL (como se ha observado en la sección anterior) en términos de tiempo de convergencia de la red, ya que la diferencia de tiempos empleados en formar la red es alta, como se observa en la tabla 26

**Tabla 28** Diferencia de tiempo de convergencia de la red (Topología #2) entre configuración R20B $\gamma$  y configuración de periodicidad en RPL.

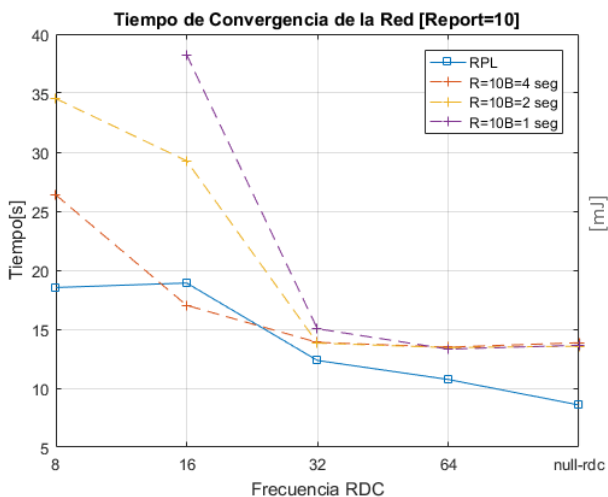
$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4	R20B2	R20B1
8	17.912	27.01	----
16	7.45	10.02	27.459
32	11.23	11.53	13.438
64	12.58	12.58	12.62
nullrdc	14.828	14.89	14.93

Con respecto al consumo de energía, observado en la figura 56, la red SDWSN al no aplicar ciclo de trabajo al radio consume hasta 5.6 veces más energía en formar la red que RPL. Sin embargo, una característica a resaltar para la red SDWSN es la periodicidad de paquetes R20B4, la cual presenta menor consumo de energía con un tiempo de convergencia igual al de las demás configuraciones de R20B $\gamma$  para esta topología. La cantidad de energía reducida al aplicar técnicas de ciclo de trabajo del radio, es notoria, con una reducción del 88 % para ambas redes.

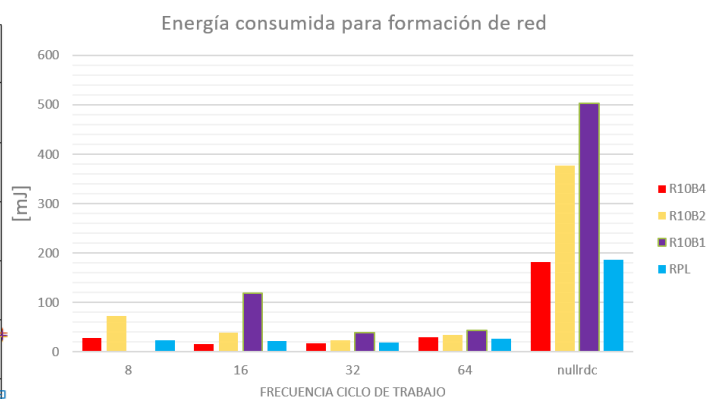
La figura 57 nos muestra una perspectiva de la cantidad de paquetes de control generados para formar la red, el impacto de overhead generado cuando se agrega técnicas de ciclo de trabajo puede ser de hasta 18.61 veces mayor en comparación de formar la red con el enrutamiento RPL.

Las figuras (58 - 60) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *medio*. Al reducir la periodicidad de los paquetes de control a una configuración *media* se aprecia reducción en el tiempo de formación de la red SDNWSN como se observa en la figura 58

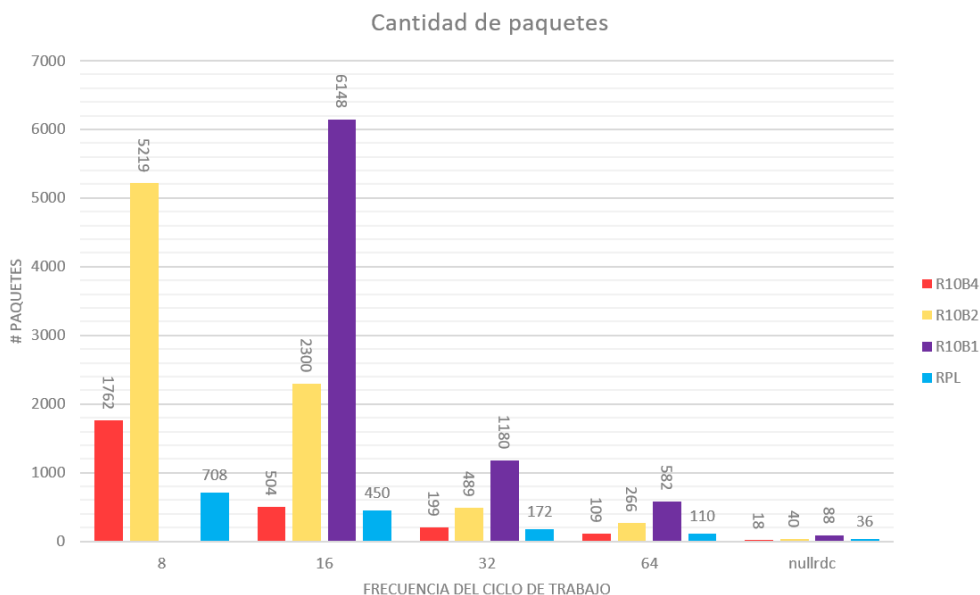
Se presenta la misma tendencia observada en la configuración de periodicidad de paquetes anterior ( $R20B\gamma$ ), donde a frecuencias altas de  $f_{RDC}$  y  $nullrdc$  se observa la uniformidad en el tiempo de convergencia, y el impacto de aplicar frecuencias bajas de  $f_{RDC}$  hace que los tiempos de convergencia de la red aumenten conforme la frecuencia de envío de paquetes Beacon aumente.



**Figura 58** Comparación del tiempo de convergencia. Topología multi-usuario, configuración R10B $\gamma$ .



**Figura 59** Comparación energía consumida. Topología multi-usuario, configuración R10B $\gamma$ .



**Figura 60** Comparación cantidad de paquetes de control. Topología multi-usuario, configuración R10By.

La configuración de periodicidad de paquetes de control *medio* en la red SDWSN hace que la red converja en un tiempo cercano al tiempo de convergencia de la red RPL cuando se utilizan frecuencias  $f_{RDC}$  de 32 y 64 Hz como se observa en la tabla 27, donde la diferencia de convergencia en ambas redes llega a ser de alrededor de un par de segundos, incluso llega en un punto a converger en un menor tiempo (configuración de periodicidad R10B4 a una  $f_{RDC}$  de 16 Hz). Sin embargo el impacto que trae implementar frecuencias de ciclo de trabajo bajas (8,16 Hz) a la red es tal que la diferencia de tiempo de formación de la red es alta en comparación con RPL, inclusive se observa el impacto a la configuración R20B1 en la cual la red no converge aun considerando el mecanismo de reconocimiento ACK para el paquete Openpath.

**Tabla 29** Diferencia de tiempo de convergencia de la red (Topología #2) entre configuración R10By y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4	R10B2	R10B1
8	7.84 seg	16 seg	---
16	-1.89 seg	10.35 seg	19.341 seg
32	1.53 seg	1.47 seg	2.683 seg
64	2.74 seg	2.7 seg	2.57 seg
nullrdc	5.27 seg	4.98 seg	5.06 seg

El consumo de energía empleado para formar esta topología se muestra en la figura 62 donde se observa una reducción del consumo de energía en comparación de la configuración R10B $\gamma$ . La diferencia del consumo de energía entre ambas configuraciones (R20B $\gamma$  – R10B $\gamma$ ) se muestra en la tabla 28, en la cual se aprecian reducciones significativas de entre el 50 y 70% al disminuir la periodicidad de los paquetes Report.

**Tabla 30** Diferencia de porcentaje de consumo de energía de la red (Topología #2) entre configuración R20B $\gamma$  y R10B $\gamma$ .

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4-R10B4	R20B2-R10B2	R20B1-R10B1
8	36.09%	35.41%	-----
16	44.19%	28.92%	74.23%
32	51.47%	51.23%	50.42%
64	50.45%	49.49%	51.69%
nullrdc	74.25%	60.08%	52.31%

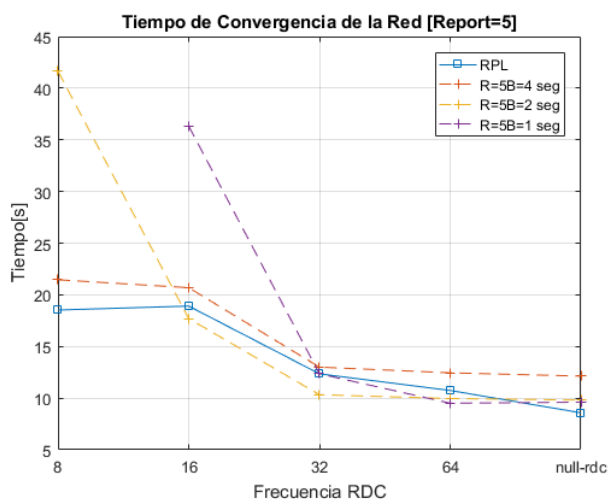
No aplicar técnicas de ciclo de trabajo en el radio en la configuración de periodicidad de paquetes R10B4, muestra tener un rendimiento competitivo a RPL teniendo ambos un consumo de energía similar. Por otro lado al aumentar la frecuencia de los paquetes Beacon (R10B2 y R10B1) presenta un consumo de energía elevado en comparación con RPL. Al reducir el tiempo de escucha en el radio mediante la técnica de ciclo de trabajo se observa una reducción del 87.69% siendo RPL la tecnología que presenta un consumo de energía menor.

Al converger la red en un menor tiempo, la cantidad de paquetes de control es reducida en comparación de la configuración R20B $\gamma$ , sin embargo, el impacto que tiene el ciclo de trabajo a la red SDWSN es significativo, en cuanto a menor frecuencia  $f_{RDC}$  y menor periodicidad de paquetes beacon, la cantidad de paquetes *overhead* utilizado es mucho mayor en comparación a RPL llegando a ser hasta 13 veces más paquetes de control para formar la red, como se observa en la figura 60.

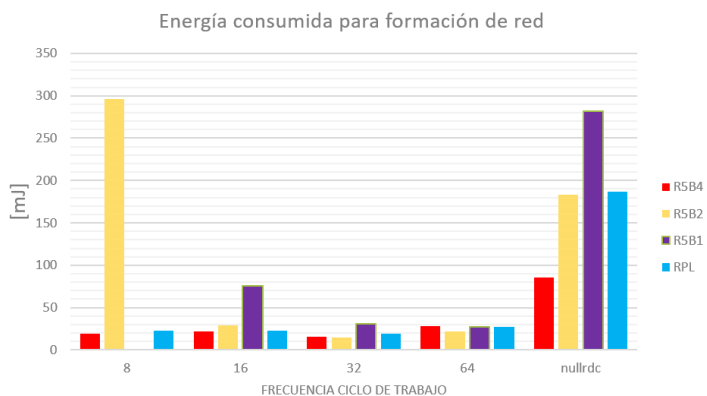
En esta topología la configuración de periodicidad de paquetes R10B4 ha demostrado ser una configuración competitiva para la red RPL que aunque no converja en un menor tiempo en comparación a RPL, presenta resultados adecuados para la comparación en términos de energía y cantidad de paquetes intercambiados, observados en las frecuencias  $f_{RDC}$  altas (32, 64 Hz) y nullrdc.

Las figuras (61 - 63) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *corto*.

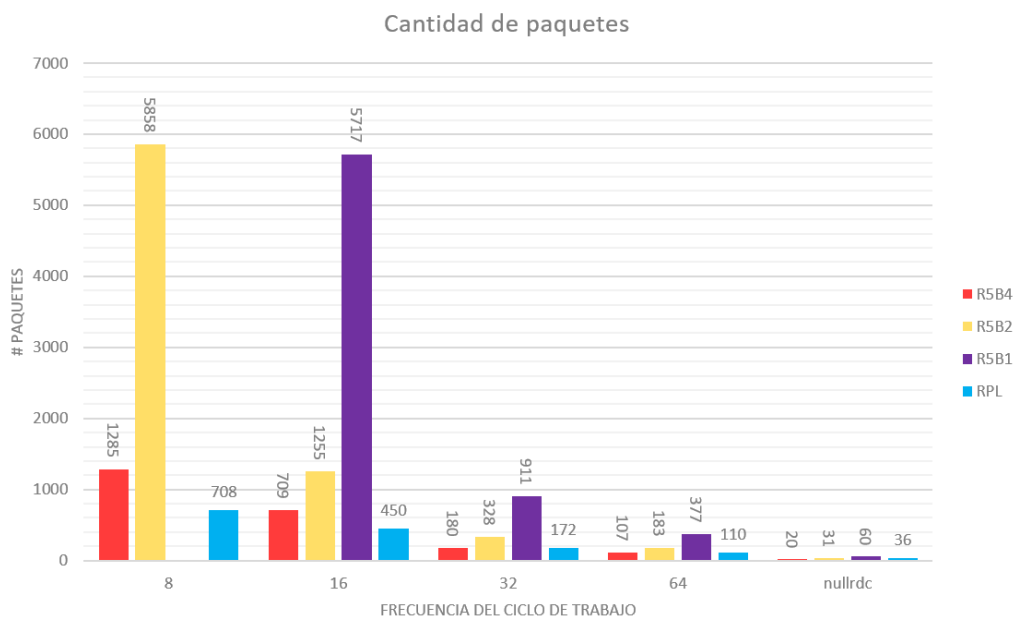
Al utilizar un esquema de paquetes de control *corto* se aprecia menor diferencia de tiempo de convergencia en comparación con RPL, teniendo en ciertos casos (R5B2 y R5B1) una convergencia de la red en menor tiempo, como se observa en la tabla 29, presentando valores de diferencia de tiempo negativos a favor a la red SDWSN.



**Figura 61** Comparación del tiempo de convergencia. Topología multi-usuario, configuración R5By.



**Figura 62** Comparación energía consumida. Topología multi-usuario, configuración R5By.



**Figura 63** Comparación cantidad de paquetes de control. Topología multi-usuario, configuración R5By.



**Tabla 31** Diferencia de tiempo de convergencia de la red (Topología #2) entre configuración R5B $\gamma$  y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R5B4	R5B2	R5B1
8	2.923 seg	23.109 seg	----
16	1.78 seg	-1.208 seg	17.42 seg
32	0.65 seg	-2.0 seg	0 seg
64	1.71 seg	-0.78 seg	-1.23 seg
nullrdc	3.55 seg	1.23 seg	1.03 seg

Las configuraciones de frecuencias bajas para configuraciones de paquetes Beacon de 2 segundos a una  $f_{RDC}$  de 8Hz y Beacon de 1 segundo a una  $f_{RDC}$  de 8 y 16 Hz respectivamente hacen que la convergencia de la red no sea tan rápida en comparación de las otras configuraciones observadas, incluso nuevamente el impacto de utilizar una frecuencia  $f_{RDC}$  de 8 Hz en conjunto de una periodicidad de paquete Beacon a 1 segundo continua haciendo que la red no converja en su totalidad incluso considerando el mecanismo de reconocimiento ACK. Esto es por el impacto que tiene aplicar esta frecuencia de ciclo de trabajo al realizar varias solicitudes del paquete Openpath.

Utilizar la configuración de periodicidad de paquetes *corto* reduce el consumo de energía para formar la red en un 50 % en comparación de la configuración R10B $\gamma$  cuando no se implementan técnicas de ciclo de trabajo. La diferencia de consumo de energía de ambas configuraciones al utilizar diferentes frecuencias de ciclo de trabajo se muestra en la tabla 30.

**Tabla 32** Diferencia de porcentaje de consumo de energía de la red (Topología #2) entre configuración R10B $\gamma$  y R5B $\gamma$ .

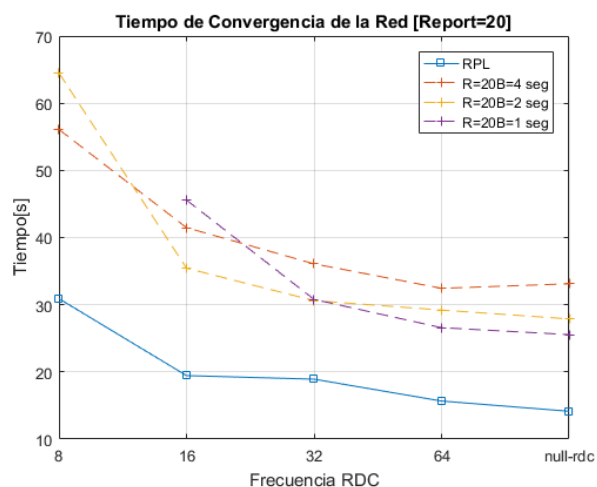
$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4-R5B4	R10B2-R5B2	R10B1-R5B1
8	31.81%	-75.36%	----
16	-26.73%	24.69%	36.74%
32	7.35%	36.89%	20.43%
64	5.38%	36.36%	37.30%
nullrdc	52.88%	51.22%	43.91%

Al implementar técnicas de ciclo de trabajo la diferencia en reducción de energía en comparación a lo observado anteriormente es menor siendo de alrededor del 30%, presentándose en algunos casos (R5B4 a 16 Hz y R5B2 a 8 Hz) un aumento del porcentaje de energía consumida para formar la red, como se observa en la figura 62, esto debido a que se presentó un incremento en el tiempo de formación de la red para estos casos con respecto a la configuración anterior (R10B $\gamma$ ).

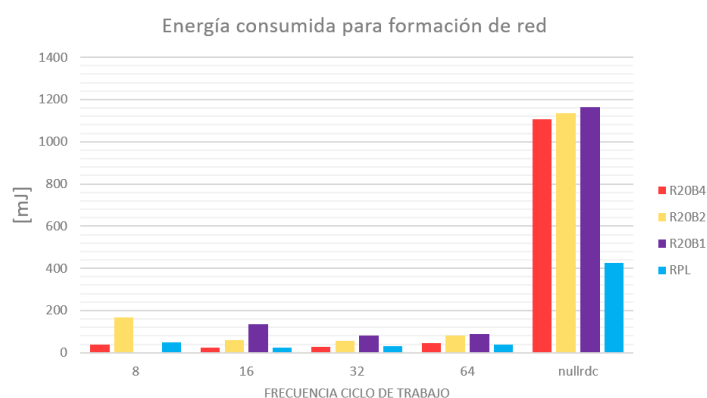
La cantidad de paquetes de control generados para formar la red SDWSN con la configuración R5B $\gamma$ , observado en la figura 63, es menor a la cantidad utilizada para formar la red con enrutamiento RPL (como se observa para la configuración R5B4 y R5B2). Como se ha observado en el análisis de la topología multisalto, al aplicar ciclo de trabajo en esta configuración hace que en la mayoría de las frecuencias  $f_{RDC}$  la cantidad de overhead sea mayor observado hasta 12.7 veces mas en comparación de lo utilizado para formar la red con RPL. Sin embargo la red SDWSN presenta buenos resultados en formar una topología donde existen múltiples usuarios compitiendo por el canal para configuraciones de periodicidad de paquetes de control *corto* (R5B2 y R5B1) y a frecuencias de ciclo de trabajo altas (64 Hz) y nullrdc, al mostrar una cantidad de paquetes de control y consumo de energía menor en un tiempo de formación de la red similar en comparación de la formación con RPL. Por otro lado, RPL muestra mayor robustez al formar la red cuando la frecuencia  $f_{RDC}$  disminuye (p.ej. 8 Hz) presentando menor cantidad de paquetes *overhead* generados, y menor cantidad de energía consumida.

### 5.2.3 Análisis de resultados comparativos para escenario #3 Topología combinada múltiples saltos- múltiples usuarios

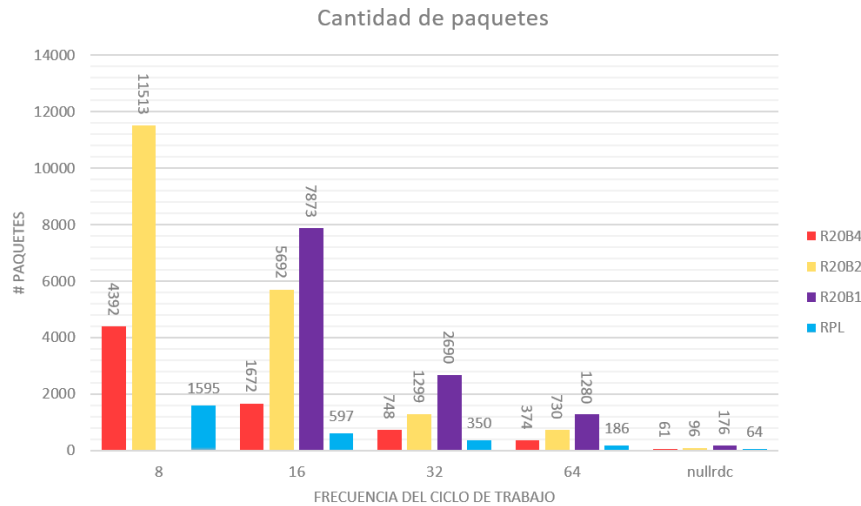
Los resultados mostrados en este apartado se obtienen utilizando la topología con múltiples usuarios distribuidos sobre múltiples saltos, encontrada en la sección 4.5.3 y las configuraciones de periodicidad de paquetes de control de la tabla 20. Las figuras (64 - 66) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *largo*.



**Figura 64** Comparación del tiempo de convergencia. Topología multi-usuario-multi-salto, configuración R20B $\gamma$ .



**Figura 65** Comparación de energía consumida. Topología multi-usuario-multi-salto, configuración R20B $\gamma$ .



**Figura 66** Comparación de cantidad de paquetes de control Topología multi-usuario-multi-salto, configuración R20By.

El tiempo de convergencia de la red al utilizar la configuración de periodicidad de paquetes *largo* se observa en la figura 64. La cantidad de tiempo para formar la red SDWSN es alta, alrededor de 1.9 veces más, en comparación con el tiempo empleado para formar la red con RPL, la diferencia de tiempo de convergencia de ambas redes para cada frecuencia  $f_{RDC}$  es mostrada en la tabla 31.

Se aprecia en la figura 64 que al disminuir la periodicidad de los paquetes Beacon al implementar frecuencias  $f_{RDC}$  bajas la pérdida de los paquetes Openpath es más notorio, esto conlleva hacer uso del mecanismo ACK implementado, generando un incremento en el tiempo de convergencia de la red. Como se ha observado en análisis de topologías anteriores utilizar una periodicidad de paquetes Beacon a 1 segundo con una frecuencia  $f_{RDC}$  de 8 Hz la red no converge en su totalidad, incluso considerando el mecanismo de reconocimiento ACK para el paquete Openpath.

**Tabla 33** Diferencia de tiempo de convergencia de la red (Topología #3) entre configuración R20By y configuración de periodicidad en RPL.

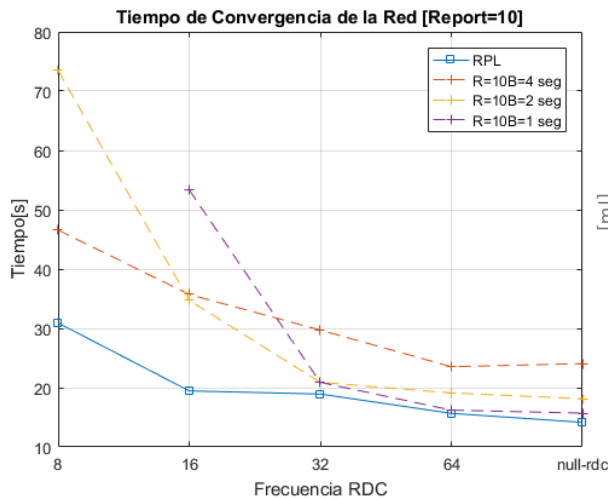
$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4	R20B2	R20B1
8	25.1863 seg	33.5962 seg	----
16	22.0248 seg	15.9817 seg	26.1291 seg
32	17.1685 seg	11.6934 seg	11.8717 seg
64	16.7806 seg	13.5402 seg	10.9159 seg
nullrdc	20.4952 seg	13.7455 seg	11.4125 seg

La cantidad de energía consumida para formar la red, observada en la figura 65, al no implementar técnicas de ciclo de trabajo es hasta 2.67 veces mayor a la energía consumida por RPL, aumentando entre cada

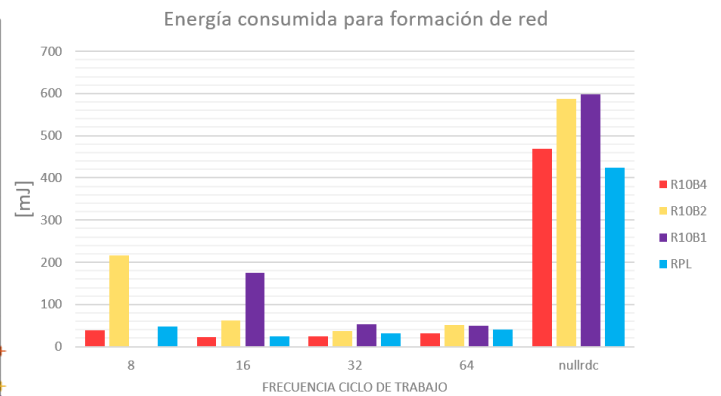
configuración de periodicidad de paquetes Beacon. Al aplicar técnicas de ciclo de trabajo al radio para formar esta topología utilizando la configuración de periodicidad de paquetes *largo* muestra una reducción del consumo de energía del 93.04 % en ambas redes. Al hacer la comparación entre procesos de formación de la red, se puede apreciar que el enrutamiento por RPL muestra menor consumo de energía para formar la red en comparación de la configuración de periodicidad de paquetes de control *largo*

En comparación con la formación de la red utilizando RPL, esta configuración de periodicidad de paquetes de control al implementar ciclo de trabajo al radio para formar la red genera una cantidad de paquetes *overhead* mucho mayor, siendo de hasta 13.2 veces como se puede observar en la figura 66.

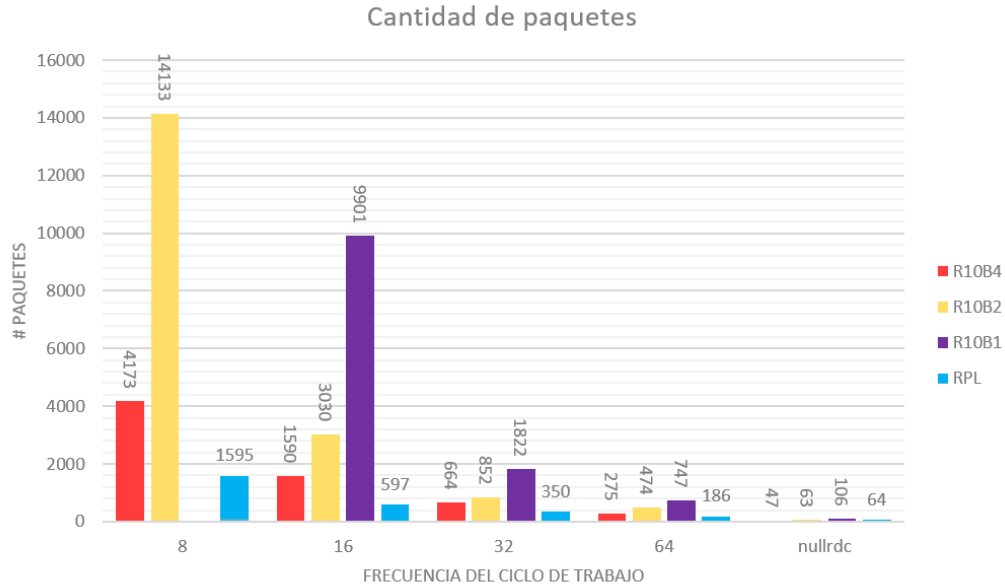
Las figuras (67 - 69) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *medio*. Al reducir la periodicidad de los paquetes de control a una configuración *media* se aprecia reducción en el tiempo de formación de la red SDNWSN como se observa en la figura 67.



**Figura 67** Comparación del tiempo de convergencia. Topología multi-usuario-multi-salto, configuración R10By.



**Figura 68** Comparación de energía consumida. Topología multi-usuario-multi-salto, configuración R10By.



**Figura 69** Comparación de cantidad de paquetes de control. Topología multi-usuario-multi-salto, configuración R10By.

El tiempo de formación empleado por la configuración R10By es muy cercano al tiempo empleado por el enrutamiento RPL para formar la red al aplicar frecuencias de ciclo de trabajo altas (32,64 Hz) y nullrdc. Sin embargo al utilizar frecuencias  $f_{RDC}$  bajas el tiempo de convergencia de la red se ve incrementado nuevamente como se observó en la configuración R20By. Al utilizar un periodo de Beacon de 1 segundo a una  $f_{RDC}$  8 Hz se presenta nuevamente que la red no converja en su totalidad, incluso considerando el mecanismo de reconocimiento ACK para el paquete Openpath, concluyendo que usar estas frecuencias para formar la red SDWSN sea poco conveniente. La diferencia de tiempos de formación entre ambas redes se muestra en la tabla 32.

**Tabla 34** Diferencia de tiempo de convergencia de la red (Topología #3) entre configuración R10By y configuración. de periodicidad en RPL

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4	R10B2	R10B1
8	15.6606 seg	42.5705 seg	-----
16	16.2623 seg	15.2726 seg	33.8944 seg
32	10.7899 seg	1.9663 seg	1.9716 seg
64	7.8607 seg	3.4399 seg	0.5454 seg
nullrdc	9.1552 seg	3.9893 seg	1.5601 seg

Disminuir la periodicidad de los paquetes de control como se ha observado en las topologías anteriores trae consigo una reducción del consumo de energía, como se observa en la figura 68. La tabla 33 muestra la cantidad de porcentaje de energía consumido en comparación de la configuración R20B $\gamma$  conforme se aplica una frecuencia  $f_{RDC}$ .

**Tabla 35** Diferencia de porcentaje de consumo de energía de la red (Topología #3) entre configuración R20B $\gamma$  y R10B $\gamma$ .

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4-R10B4	R20B2-R10B2	R20B1-R10B1
8	1.29%	-23.05%	
16	7.23%	-4.38%	-23.23%
32	17.02%	36.54%	33.29%
64	27.50%	38.17%	44.21%
nullrdc	57.62%	48.32%	48.72%

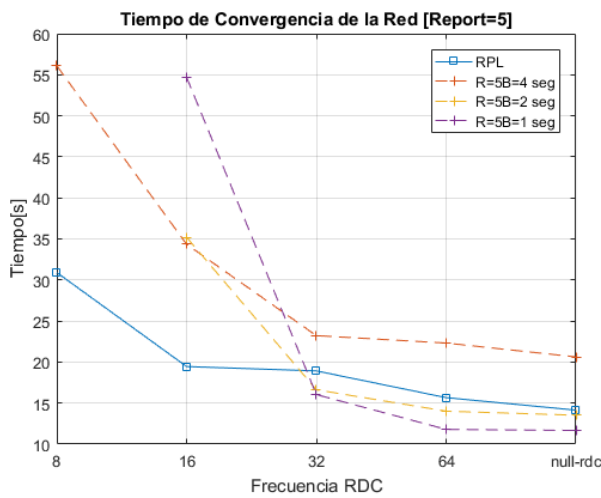
En la tabla 33 se muestra que no implementar ciclo de trabajo al radio presenta una reducción del 57.62% máxima en la configuración de periodicidad de paquetes R10B $\gamma$ . La diferencia de energía consumida cuando se implementan frecuencias  $f_{RDC}$  altas de ciclo de trabajo (32,64 Hz) trae consigo una reducción media del 32%. Por otra parte se observó el impacto que genera en el consumo de energía implementar frecuencias  $f_{RDC}$  bajas (8,16 Hz), el cual es significativo ya que la diferencia del consumo de energía entre ambos periodos de paquetes control es muy baja, siendo en algunos casos (casillas de color amarillo) un incremento en el consumo de energía para formar la red en comparación de la configuración R20B $\gamma$ .

Al aplicar ciclo de trabajo a la formación de ambos paradigmas se observa una reducción del 92 por ciento del consumo de energía como se aprecia en la figura 68. Sin embargo al aplicar frecuencias  $f_{RDC}$  bajas (8, 16 Hz) en conjunto con baja periodicidad de paquetes de control (R10B2 y R10B1) la reducción del consumo de energía se encuentra entre un 63 y 70 % respectivamente.

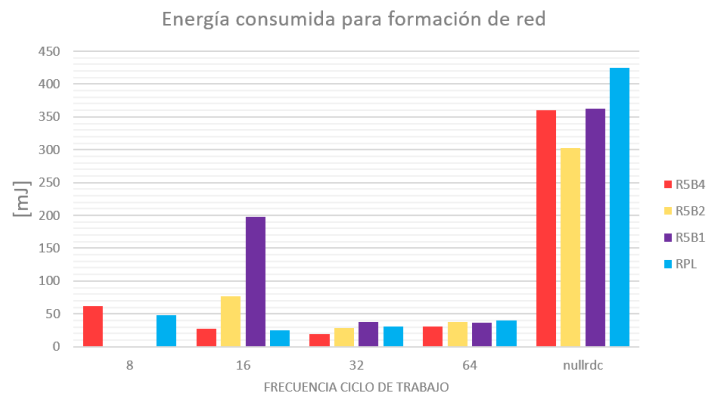
Con respecto a la cantidad de paquetes generados para dar formación a ambas redes, figura 69, se observa que la formación de la red SDWSN requiere menor cantidad de paquetes de control para las configuraciones (R10B4 y R10B2) en comparación a la formación con RPL. Sin embargo al implementar ciclo de trabajo al radio, se observa que la cantidad de overhead generado para formar la red SDWSN crece en proporción a la periodicidad de paquetes Beacon, siendo de hasta 16.5 veces mayor la cantidad de paquetes para formar la red SDWSN en comparación de la formación de la red con RPL

Las figuras (70 - 71) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *corto*

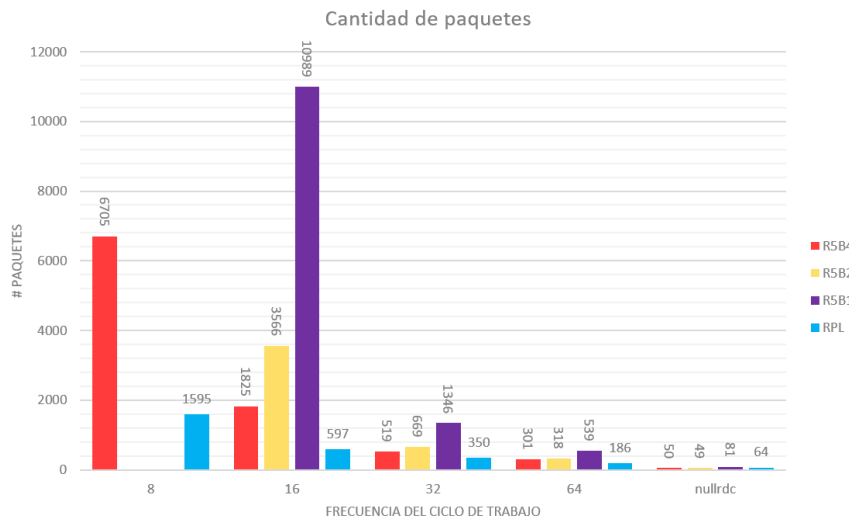
Como se ha observado al reducir la cantidad de periodo de paquete de control, la cantidad de tiempo de convergencia se reduce. Sin embargo se sigue observando la tendencia presentada en las configuraciones anteriores al utilizar frecuencias de ciclo de trabajo bajas, el impacto del ciclo de trabajo es tal que al utilizar una frecuencia  $f_{RDC}$  de 8 Hz en conjunto de una periodicidad de paquete Beacon a 1 y 2 segundos la red no converge en su totalidad incluso considerando el mecanismo de reconocimiento ACK. La diferencia de tiempo que tiene esta configuración en formar la red en comparación con la que presenta el enrutamiento RPL para formar la red se presenta en la tabla 34.



**Figura 70** Comparación del tiempo de convergencia. Topología multi-usuario-multi-salto, configuración R5By.



**Figura 71** Comparación de energía consumida. Topología multi-usuario-multi-salto, configuración R5By.



**Figura 72** Comparación de cantidad de paquetes de control. Topología multi-usuario-multi-salto, configuración R5By..

**Tabla 36** Diferencia de tiempo de convergencia de la red (Topología #3) entre configuración R5By y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R5B4	R5B2	R5B1
8	25.19 seg	-----	-----
16	14.8907 seg	15.7481111 seg	35.2466 seg
32	4.2979 seg	-2.2961111 seg	-2.8821 seg
64	6.6486 seg	-1.6493333 seg	-3.8691 seg
nullrdc	6.9052 seg	-0.6194667 seg	-2.4712 seg

Como se puede apreciar en la tabla 34, la diferencia de tiempo de formación en comparación con RPL es menor. Las configuraciones de paquetes Beacon de 2 y 1 segundos a frecuencias  $f_{RDC}$  altas (32,64 Hz) y nullrdc presentan un menor tiempo de convergencia de la red mostrando una diferencia de tiempos de formación de la red con valor negativa (indicando ventaja para la red SDWSN) respecto a la formación con RPL.

El consumo de energía empleado para formar la red se observa en la figura 71, en ella se muestra la reducción del consumo de energía al reducir el tiempo de entrega de los paquetes Report, la diferencia del consumo de energía comparada con la configuración anterior R10By se muestra en la tabla 35.

**Tabla 37** Diferencia de porcentaje de consumo de energía de la red (Topología #3) entre configuración R10By y R5By.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4-R5B4	R10B2-R5B2	R10B1-R5B1
8	-36.08%	-----	-----
16	-13.50%	-17.59%	-11.98%
32	19.99%	21.37%	29.10%
64	5.53%	26.36%	26.77%
nullrdc	23.28%	48.36%	39.36%

Al implementar frecuencias altas de  $f_{RDC}$  se observa una reducción del consumo de energía significativo, con una reducción de energía máximo del 48% para el periodo de paquetes Beacon de 2 segundos. Nuevamente como se ha observado en las topologías anteriores, aplicar frecuencias bajas de  $f_{RDC}$  presenta un incremento de energía en comparación de la configuración anterior, esto debido al tiempo de convergencia de la red empleado en estas frecuencias por la utilización del mecanismo ACK para solicitar



la ruta para converger la red. Al no implementar técnicas de ciclo de trabajo se observa que esta configuración es competitiva a RPL debido a que tiene menor consumo de energía para formar la red que RPL, teniendo una reducción máxima de 121.54 mJ encontrada en la configuración (R5B2).

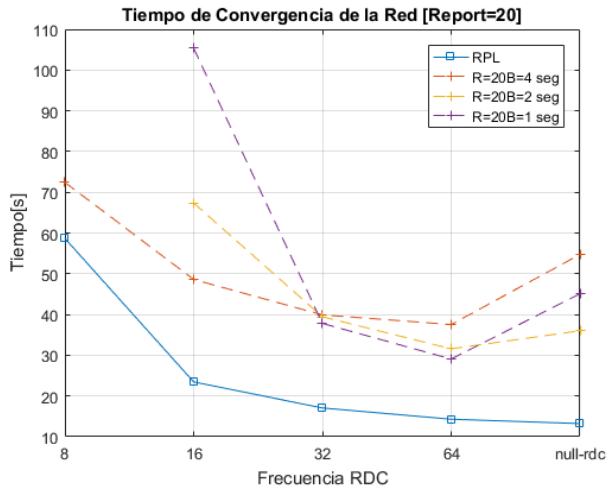
Al implementar técnicas de ciclo de trabajo la reducción del consumo de energía en ambas redes es significativa siendo del 90.26 % a lo largo de las frecuencias de ciclo de trabajo. Sin embargo como se ha observado en la figura 71 implementar bajas frecuencias de ciclo de trabajo (8, 16 Hz) con una periodicidad de paquetes Beacon de 2 y 1 segundos hace que el consumo de energía incremente, teniendo solamente una reducción del consumo de energía del 74 y 45 % comparado al consumo de energía obtenido al no tener un ciclo de trabajo aplicado.

Con respecto a la cantidad de paquetes generados para formar la red, figura 72 se observa que no tener un ciclo de trabajo hace que nuevamente la configuración presentada en SDWSN sea competitiva con RPL al presentar menor número de paquetes de control. Sin embargo, como se ha observado en topologías anteriores, el impacto que tiene agregar un ciclo de trabajo a la red SDWSN hace que se genere mayor cantidad de paquetes de control, siendo para esta configuración de hasta 18 veces más paquetes en comparación con la cantidad de paquetes generados con RPL.

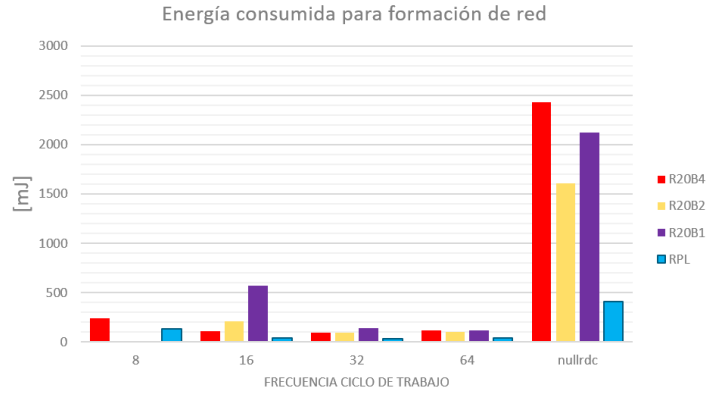
#### **5.2.4 Análisis de resultados comparativos para escenario #4 Topología con aplicación**

Los resultados mostrados en este apartado se obtienen utilizando la topología basada en una aplicación, dicha topología es encontrada en la sección 4.5.4 y las configuraciones de periodicidad de paquetes de control de la tabla 20. Las figuras (73 - 75) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *largo*.

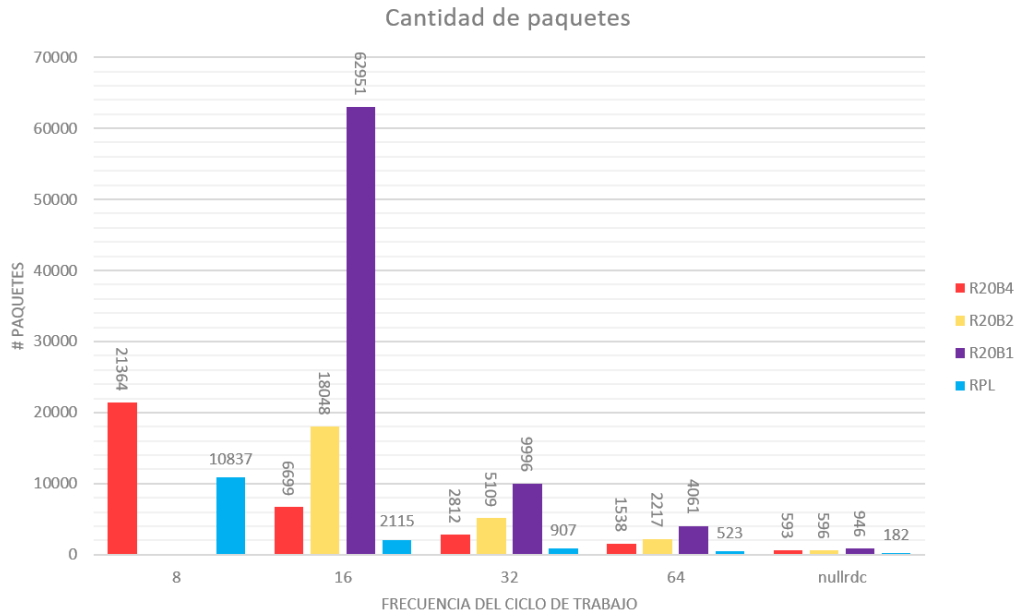
En esta topología la cantidad de nodos existentes en la red, es mayor al de las topologías previamente mostradas, por lo cual se espera un mayor tiempo de convergencia en la red. Se puede observar en las tres figuras, conforme el ciclo de trabajo aumenta, el desempeño de ambos enfoques mejora también. Sin embargo, se obtienen mejores métricas cuando se utiliza RPL para formar la red. Particularmente, el tiempo de convergencia de la red SDWSN mostrado en la figura 73 es de 2.66 veces mayor en la mayoría de las frecuencias  $f_{RDC}$  en comparación con el tiempo de formación de una WSN usando RPL. En la tabla 36 se muestra la diferencia de tiempo en segundos para formar la red en comparación con RPL.



**Figura 73** Comparación del tiempo de convergencia. Topología aplicación, configuración R20By.



**Figura 74** Comparación de energía consumida. Topología aplicación, configuración R20By.



**Figura 75** Comparación de cantidad de paquetes. Topología aplicación, configuración R20By.

**Tabla 38** Diferencia de tiempo de convergencia de la red (Topología #4) entre configuración R20By y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4	R20B2	R20B1
8	13.76 seg	---	---
16	25.14 seg	43.91 seg	81.896 seg
32	22.85 seg	22.37 seg	20.75 seg
64	23.23 seg	17.28 seg	14.77 seg
nullrdc	41.51 seg	22.772 seg	31.87 seg

Se observa que la configuración de periodicidad de paquetes *largo* presenta una diferencia de tiempo alta en la formación de la red en comparación con la red RPL. En especial se observa una cantidad de tiempo empleado para formar la red mayor cuando no se aplica ciclo de trabajo en comparación de utilizar una frecuencia  $f_{RDC}$  de 64 Hz. Al utilizar frecuencias de ciclo de trabajo bajas (8,16 Hz) en combinación de una periodicidad de paquetes de control bajo (Beacon de 1 y 2 segundos) el tiempo de convergencia de la red es incrementado, incluso se muestra que la red no llega a converger aun considerando el mecanismo de reconocimiento ACK para el paquete Openpath

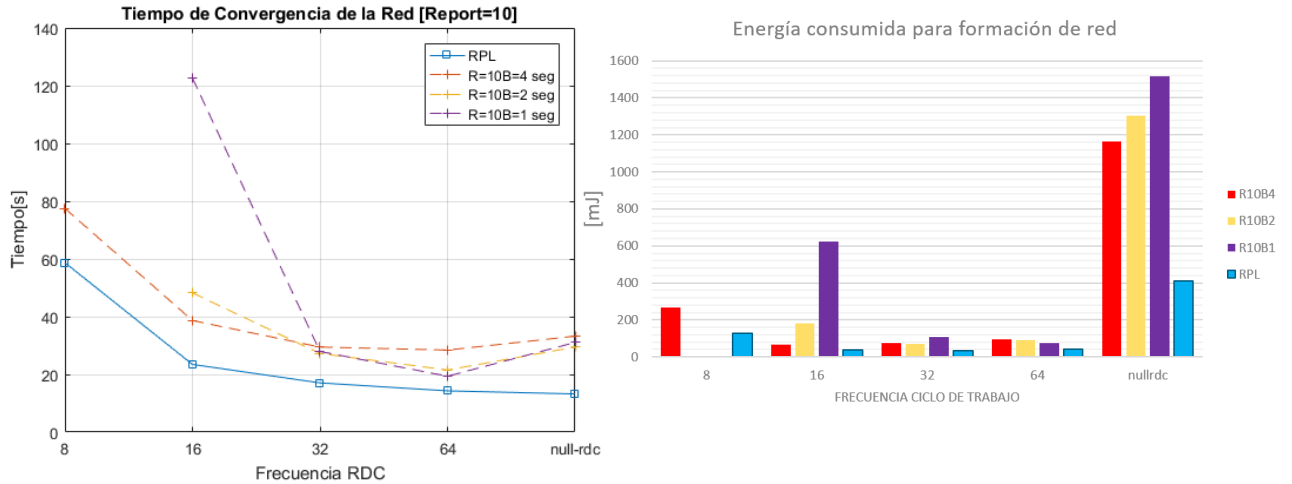
Tener un tiempo de convergencia elevado repercute en el consumo de energía empleado para formar la red, figura 74. El consumo de energía al no aplicar técnicas de ciclo de trabajo es particularmente mayor (siendo de hasta 5 veces más) para la red SDWSN que el utilizado para la red RPL.

La reducción del consumo de energía en esta topología al aplicar ciclo de trabajo es mayor, siendo para la red SDWSN una reducción del 94.09% y para RPL una reducción del 91%. Sin embargo se observa que al tener frecuencias  $f_{RDC}$  bajas (8,16 Hz) en combinación con una periodicidad baja de paquetes Beacon (1, 2 segundos) el porcentaje de reducción del consumo de energía es menor (86.97% y 72.9% respectivamente). Incluso para la red RPL a una frecuencia  $f_{RDC}$  de 8 Hz la reducción del consumo de energía también es disminuido, siendo de un 68%.

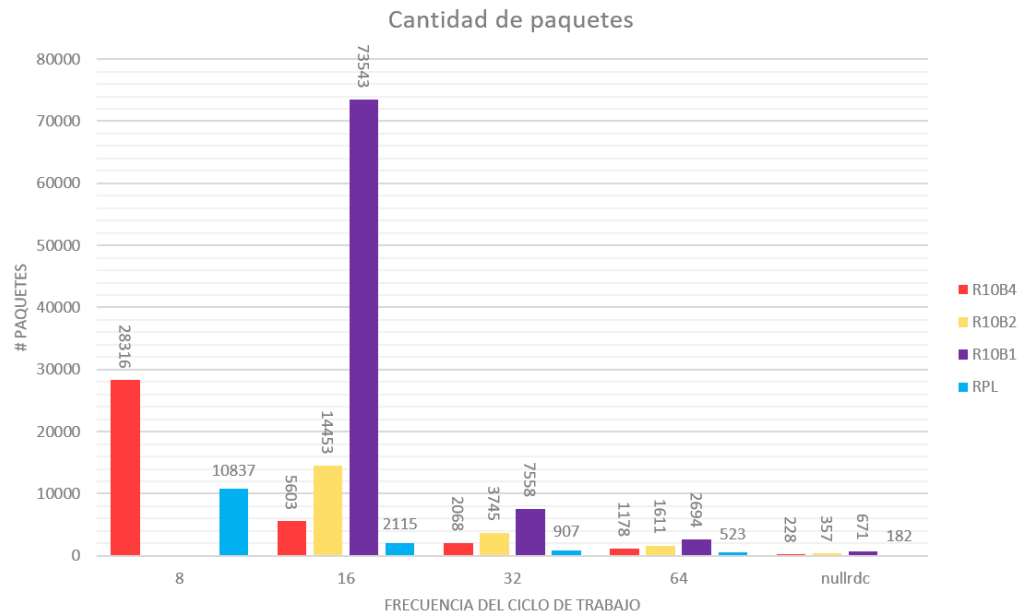
La generación de paquetes de control aumenta como se observa en la figura 75. Para formar la red SDWSN se generó mayor número paquetes de control en comparación de la red RPL, siendo de hasta 29 veces la cantidad de paquetes generados para formar la red RPL cuando se aplica ciclo de trabajo al radio.

Las figuras (76 - 78) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes *medio*.

Reducir el período de envío del paquete Report reduce el tiempo de convergencia de la red como se muestra en la figura 76. El tiempo de formación empleado por la configuración R10By es cercano al tiempo empleado por el enrutamiento RPL para formar la red al aplicar frecuencias de ciclo de trabajo altas (32,64 Hz) y nullrdc. Sin embargo al utilizar frecuencias  $f_{RDC}$  bajas el tiempo de convergencia de la red se ve incrementado nuevamente como se observó en la configuración R20By. Al utilizar un periodo de Beacon de 1 y 2 segundos a una  $f_{RDC}$  8 Hz se presenta nuevamente que la red no converja en su totalidad, incluso considerando el mecanismo de reconocimiento ACK para el paquete Openpath.



**Figura 76** Comparación del tiempo de convergencia. **Figura 77** Comparación de energía consumida. Topología aplicación, configuración R10By.



**Figura 78** Comparación de cantidad de paquetes de control. Topología aplicación, configuración R10By.

La tabla 37 muestra la diferencia del tiempo de convergencia presentado en la figura 76 entre la formación de la red SDWSN y el enrutamiento por RPL al utilizar el modo de configuración de periodicidad de paquetes *medio* para la red SDWSN. Se observa en la tabla que la diferencia de tiempo al usar esta configuración en comparación de la configuración de periodicidad *largo* es disminuido en la mayoría de las frecuencias  $f_{RDC}$ , sin embargo presenta un ligero aumento en las configuraciones de  $f_{RDC}$  altas.

**Tabla 39** Diferencia de tiempo de convergencia de la red (Topología #4) entre configuración R10B $\gamma$  y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4	R10B2	R10B1
8	18.8 seg	-----	-----
16	15.2 seg	24.77 seg	99.16 seg
32	12.41 seg	10.18 seg	10.88 seg
64	14.12 seg	7.24 seg	5.07 seg
nullrdc	20.01 seg	16.22 seg	17.83 seg

Como se ha observado en topologías anteriores, reducir la periodicidad de los paquetes Report disminuye el consumo de energía empleado para formar la red figura 77, la tabla 38 muestra la diferencia en porcentaje de la reducción de energía entre la configuración R20B $\gamma$  y la configuración R10B $\gamma$

La reducción en el consumo de energía en esta topología es variada, para el caso de no aplicar técnicas de ciclo de trabajo (nullrdc), la reducción del consumo de energía se encuentra entre un 50 y 19 %, esta variación entre configuraciones es debido al tiempo de formación de la red en ambas configuraciones. La configuración de R10B4 presenta mayor reducción en comparación del resto de configuraciones.

**Tabla 40** Diferencia de porcentaje de consumo de energía de la red (Topología #4) entre configuración R20B $\gamma$  y R10B $\gamma$ .

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R20B4-R10B4	R20B2-R10B2	R20B1-R10B1
8	-11.93%	-----	-----
16	37.65%	13.04%	-8.20%
32	21.25%	23.08%	20.99%
64	16.58%	11.83%	34.84%
nullrdc	52.11%	19.10%	28.46%

Al aplicar técnicas de ciclo de trabajo, la reducción de energía consumida es baja en comparación al análisis observado en otras topologías para esta configuración, siendo alrededor de alrededor de 22%, sin embargo en esta ocasión al combinar frecuencias bajas de  $f_{RDC}$  se presenta un incremento en comparación de la configuración anterior observado en las casillas amarillas de la tabla 38.

A pesar de que en la configuración de nullrdc se observa una reducción en el consumo de energía al formar la red SDWSN con la periodicidad de paquetes a R10Bγ, la cantidad de energía consumida sigue siendo alta, de alrededor de un 68.75 % en comparación con RPL. Aplicar ciclo de trabajo al radio al utilizar la configuración R10Bγ se observó una reducción del 93% de la energía. Sin embargo para frecuencias bajas de  $f_{RDC}$  la reducción del consumo de energía es mucho menor, siendo de hasta un 58% (para la configuración R10B1 a 16 Hz). Esta observación demuestra que aplicar frecuencias bajas de ciclo de trabajo no siempre trae un beneficio a la formación de la red en términos de consumo de energía.

La figura 78 muestra la cantidad de paquetes de control utilizados para formar esta topología. Al utilizar una configuración de periodicidad R10Bγ la cantidad de paquetes generados disminuye en comparación de la configuración R20Bγ al aplicar frecuencias  $f_{RDC}$  altas, en cambio a frecuencias  $f_{RDC}$  bajas se llega a notar un incremento en la cantidad de paquetes generados de hasta 34 veces más comparándolo con RPL. A pesar de la reducción de paquetes observada a frecuencias  $f_{RDC}$  altas en comparación con la red RPL la cantidad de paquetes para formar la red sigue siendo mayor

Las figuras (79 - 81) muestran el desempeño obtenido para formar la SDWSN al utilizar una configuración de envío de paquetes corto

Al aplicar un modo de periodicidad de paquetes corto a la formación de la red SDWSN se observa una ligera disminución en el tiempo de convergencia, como se observa en la figura 81. La tabla 39 muestra la diferencia de esta disminución en comparación de la formación de la red con enrutamiento RPL.

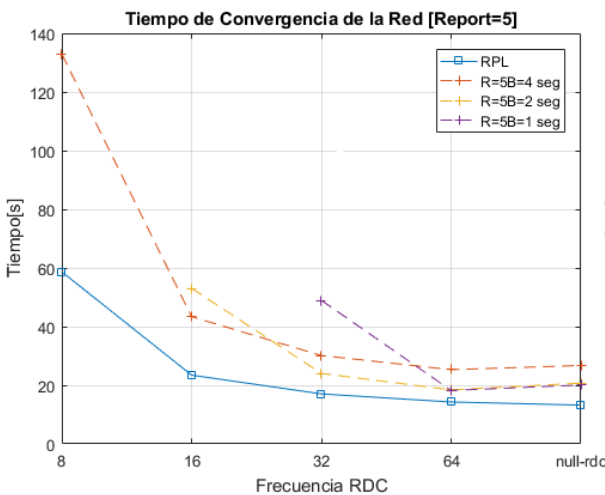


Figura 79 Comparación del tiempo de convergencia. Topología aplicación, configuración R5Bγ.

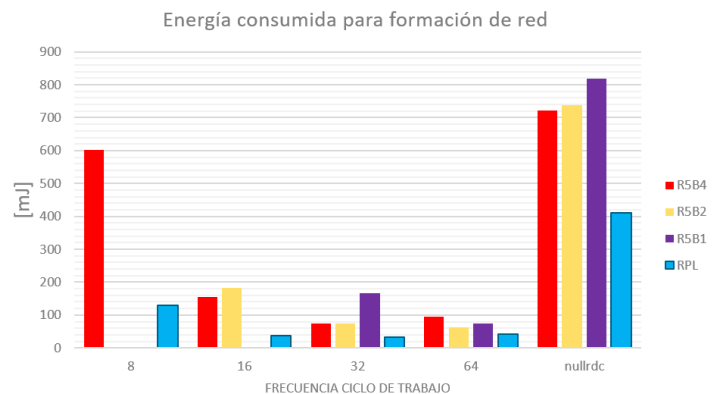
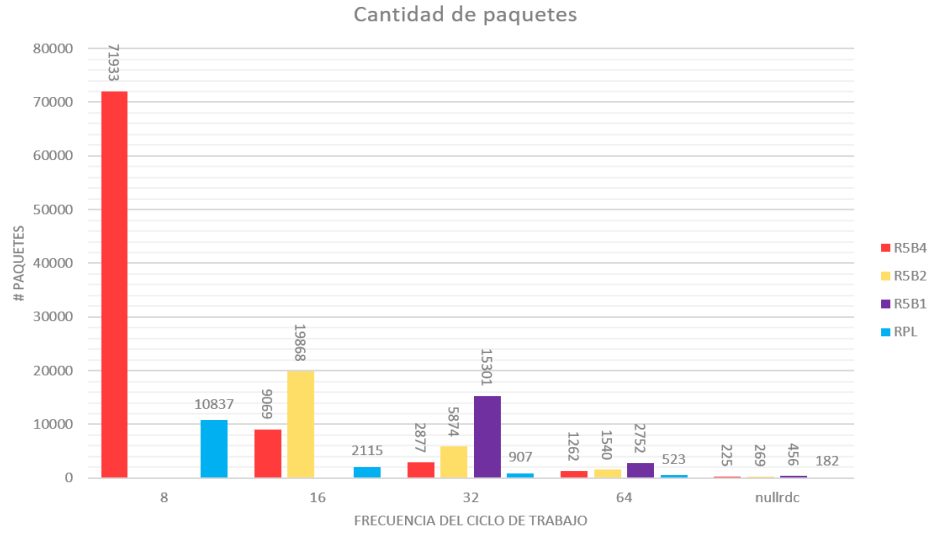


Figura 80 Comparación de energía consumida. Topología aplicación, configuración R5Bγ.



**Figura 81** Comparación de cantidad de paquetes de control. Topología aplicación, configuración R5Bγ.

**Tabla 41** Diferencia de tiempo de convergencia de la red (Topología #4) entre configuración R5Bγ y configuración de periodicidad en RPL.

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R5B4	R5B2	R5B1
8	74.271 seg	-----	-----
16	19.92 seg	29.6 seg	-----
32	13.0822 seg	6.978 seg	31.88 seg
64	11.02 seg	4.13 seg	3.94 seg
nullrdc	13.52 seg	7.51 seg	6.852 seg

A diferencia de las topologías anteriores que al reducir la periodicidad de paquetes Report a 5 segundos se presenta un decremento en el tiempo de formación de la red significativo proveyendo una convergencia de la red en menor tiempo a la formación con RPL. Para una topología con mayor cantidad de nodo, distribuidos a lo largo de una mayor cantidad de saltos, el decremento del tiempo de convergencia de la red es mínimo, haciendo que la red RPL siga convergiendo en menor tiempo que la red SDWSN.

Nuevamente se observa que el impacto del ciclo de trabajo es tal que al utilizar una frecuencia  $f_{RDC}$  de 8 Hz en conjunto de una periodicidad de paquete Beacon a 1 y 2 segundos la red no converja en su totalidad incluso considerando el mecanismo de reconocimiento ACK, dicha tendencia se ha presentado en los análisis de las configuraciones R20Bγ y R10Bγ

La cantidad de energía consumida para esta configuración (R5B $\gamma$ ) es mostrada en la figura 80 y la tabla 40 muestra la diferencia en porcentaje de la energía requerida para esta configuración en comparación de la energía requerida para formar la red con la configuración de periodicidad R10B $\gamma$ .

**Tabla 42** Diferencia de porcentaje de consumo de energía de la red (Topología #4) entre configuración R10B $\gamma$  y R5B $\gamma$ .

$f_{RDC}$	Configuración de periodicidad de paquete de control		
	R10B4-R5B4	R10B2-R5B2	R10B1-R5B1
8 Hz	-55.35%	----	----
16 Hz	-55.64%	0.28%	----
32 Hz	-1.764%	-4.04%	-35.12%
64 Hz	1.08%	30.48%	1.63%
nullrdc	37.92%	43.45%	45.97%

La energía consumida para formar la red SDWSN al no aplicar técnicas de ciclo de trabajo presenta una disminución del 42% en comparación de la configuración anterior (R10B $\gamma$ ). A pesar de esta reducción de energía en la formación de la red, la energía consumida sigue siendo mayor a la energía consumida para formar la red RPL con una diferencia de 349 mJ. Al aplicar ciclo de trabajo al radio, se aprecia que la reducción del consumo de energía es mínima en la configuración de periodicidad *corto*, donde incluso en la mayoría de las frecuencias de  $f_{RDC}$  presenta un incremento en el consumo de energía (casillas amarillas de la tabla 39), siendo más recurrente en frecuencias  $f_{RDC}$  bajas. En la figura 80 se puede observar el impacto de implementar ciclo de trabajo sobre el consumo de energía en una topología donde la red es considerada grande, conforme la  $f_{RDC}$  disminuye el consumo de energía en la red SDWSN aumenta.

La cantidad de paquetes de control generados en esta configuración (figura 81) continua siendo mayor aun cuando la periodicidad de generación de estos paquetes fue reducida, al aplicar ciclo de trabajo al radio la cantidad de overhead generado puede llegar a ser de hasta 16.86 veces más que la cantidad de overhead utilizado para formar la red con RPL. Esto demuestra que la red SDWSN necesita generar una cantidad mayor de paquetes para formar la red cuando se aplican técnicas de ciclo de trabajo al radio en comparación con la formación de la red utilizando enrutamiento RPL.



### 5.2.5 Conclusiones de análisis de resultados

Las conclusiones más relevantes de la discusión sobre las características observadas en el análisis del impacto del ciclo de trabajo en el radio sobre las diferentes topologías se enlistan a continuación:

- Aplicar ciclo de trabajo al radio de comunicación implica que cada paquete a enviar genera un overhead de paquetes duplicados para aumentar la probabilidad de una transmisión exitosa (ver sección 3.3.3), esto puede traer consigo problemas de contención en el canal.
- A pesar del overhead generado, aplicar ciclo de trabajo al radio de los nodos sensores reduce del consumo de energía entre el 87% (figura 59) y el 94% (figura 74) en ambos paradigmas de redes.
- Se demostró que el ciclo de trabajo disminuye el consumo de energía. Sin embargo, la elección de la  $f_{RDC}$  no es trivial, debido a que afecta la cantidad de overhead generado y el tiempo empleado para converger la red
- El overhead generado por las técnicas de transmisión cuando se utiliza ciclo de trabajo aumenta conforme la  $f_{RDC}$  disminuye. Esto genera mayor contención y en conjunto con periodos de envío de paquetes Beacon cortos (p.ej. a 1 segundo) existe mayor pérdida de paquetes Openpath.
- Utilizar frecuencias de ciclo de trabajo de 32 o 64 Hz presenta un tiempo de convergencia similar a no aplicar la técnica de ciclo de trabajo, presentando menor consumo de energía para formar ambas redes SDWSN y RPL.
- Para topologías multisalto el periodo de paquete Report define el tiempo de formación de la red SDWSN. Una frecuencia baja de envío de paquetes Beacon hace que el tiempo de convergencia sea cercano al periodo del paquete Report. Para topologías de tipo malla a un salto solo los paquetes Report definen el tiempo de convergencia de la red.
- En una topología multisalto al utilizar ambos paradigmas el tiempo de convergencia disminuye conforme la frecuencia  $f_{RDC}$  aumenta.
- En la topología malla a un salto de la red SDWSN, se observa una uniformidad en el periodo de convergencia al agregar frecuencias altas de ciclo de trabajo, sin embargo para frecuencias bajas (8,16 Hz) el tiempo de convergencia de la red aumenta en comparación a lo presentado a frecuencias altas. En comparación al utilizar RPL sobre esta topología, el tiempo de convergencia disminuye conforme la frecuencia  $f_{RDC}$  aumenta.
- Al utilizar técnicas de ciclo de trabajo, la cantidad de paquetes de control generados para formar una red de tipo malla SDWSN a un salto es mayor en comparación de la cantidad utilizada para

formar la red SDWSN multisalto, debido a que existe mayor probabilidad de descartar paquetes al encontrar el medio ocupado.

- La red SDWSN no converge a frecuencias  $f_{RDC}$  bajas con configuración de periodicidad de paquetes de control *cortos* (ver tabla 20 sección 5.2), en una configuración de paquetes de control *medio* presenta un consumo de energía y tiempo de convergencia de la red mayor en comparación a RPL.
- La configuración de paquetes de control *largo* para la red SDWSN no es recomendable a ser utilizada para la formación de la red. Debido a que presenta una mayor cantidad de tiempo en formar la red y mayor consumo de energía en comparación de las configuraciones *medio* y *corto*
- Al aplicar bajas frecuencias de ciclo de trabajo a la red SDWSN hace que sea necesario un mecanismo para la redistribución del paquete Openpath como el propuesto en la sección 5.3. Sin embargo el uso de este mecanismo hace que el tiempo de convergencia de la red aumente en dichas frecuencias.
- El mecanismo propuesto en la sección 5.3 dió soporte a converger la red SDWSN a frecuencias  $f_{RDC}$  de 16 Hz, sin embargo la convergencia de la red sigue presentando limitaciones al utilizar una  $f_{RDC}$  de 8 Hz, abriendo la oportunidad a diseñar mecanismos que tomen en cuenta estas limitaciones y sean eficientes en tiempo de convergencia y overhead.
- Aplicar frecuencias bajas de ciclo de trabajo sobre una topología con mayor cantidad de nodos como la presentada en la sección 4.5.4 muestra un bajo desempeño en comparación con RPL, ya que en la mayoría de los casos se presenta que la red no converja o necesite de un mayor tiempo para hacerlo.
- La comparación del rendimiento de las configuraciones de periodicidad de los paquetes en SDWSN con las configuraciones de RPL sobre las diferentes topologías analizadas se muestra en la tabla 41.

**Tabla 43** Análisis de resultados de desempeño en la formación de la red.

<b>Topología</b>	<b>Resultado de análisis</b>
<i>Red compuesta por 5 nodos distribuidos a lo largo de varios saltos para llegar al resumidero (Topología #1)</i>	Configuraciones de periodicidad R5B2 y R5B1 con frecuencia $f_{RDC}$ de 32 y 64 Hz presenta mejor desempeño en términos de energía y tiempo de convergencia en comparación de la formación con RPL
<i>Red compuesta por 7 nodos distribuidos a 1 salto del resumidero, todos dentro del rango de cobertura (Topología #2)</i>	Configuraciones de periodicidad R5B2 a una frecuencia de 64 Hz presenta un tiempo de convergencia similar a la formación con RPL pero con menor consumo de energía
<i>Red compuesta por la combinación de tener 7 nodos distribuidos a diferentes saltos del resumidero, teniendo en cada salto dos o más nodos dentro del rango de cobertura (Topología #3)</i>	La configuración de periodicidad de paquetes R5B2 a una $f_{RDC}$ de 64 Hz para la red SDWSN presenta mejores resultados de tiempo de convergencia de la red y consumo de energía en comparación con RPL
<i>Red compuesta por una gran cantidad de usuarios (19 nodos) en la red distribuidos a lo largo de múltiples saltos para llegar al resumidero (Topología #4)</i>	La configuración de periodicidad de los paquetes de control de la red SDWSN en conjunto con las frecuencias de ciclo de trabajo presenta menor desempeño para formar esta topología en comparación con las configuraciones de formación de RPL.  RPL presentó mayor robustez en términos de consumo de energía, tiempo de convergencia de la red y cantidad de overhead generado.

- Se eligió a la configuración de paquetes periódicos Report a 5 segundos con Beacon 2 segundos a una frecuencia  $f_{RDC}$  de 64 Hz como configuración adecuada para formar la red SDWSN, debido a que presenta un tiempo de convergencia y consumo de energía menor al utilizado en RPL

La tabla 42 muestra los escenarios y las características en los que cada solución presentó mejores resultados, denotado de color verde.

<i>Escenario</i>	<b>RPL</b>		<b>SDWSN</b>	
	Consumo de Energía	Tiempo de convergencia de la red	Consumo de Energía	Tiempo de convergencia de la red
<i>Característica para formar la red</i>				
<i>Topología #1</i>				
<i>Topología #2</i>				
<i>Topología #3</i>				
<i>Topología #4</i>				

## Capítulo 6. Conclusiones y trabajo a futuro

---

### 6.1 Introducción

La principal aportación del presente trabajo de tesis es la evaluación del impacto que genera aplicar técnicas de ciclo de trabajo en el radio transceptor a la formación de una red WSN utilizando como modelo el paradigma de SDN sobre las WSN (SDWSN). Como referencia se comparó el proceso de formación de la SDWSN con el desempeño observado en la formación de una WSN utilizando su enrutamiento de facto, i.e., RPL.

Otra aportación realizada de este trabajo es el diseño de un mecanismo de solicitud de ruta en caso de que el paquete Openpath no llegue a su destino. Este mecanismo de reconocimiento ACK permite realizar una comparación a la formación de la red con RPL. En este capítulo se presentan las conclusiones más importantes generadas de la evaluación de implementar técnicas de ciclo de trabajo al proceso de formación de una WSN con el paradigma de SDN y RPL, así como los métodos utilizados para la obtención del mismo.

### 6.2 Conclusiones sobre validación del simulador

Cooja es una herramienta para analizar el desempeño de las WSN, es utilizado en la mayoría de los trabajos relacionados con aplicaciones IoT. Sin embargo, es necesario validar las configuraciones que se ingresarán, para verificar que operen de acuerdo con lo esperado con los protocolos y estándares de los sistemas de comunicaciones a utilizar, en este caso el estándar de IEEE 802.15.4.

Particularmente, mediante la validación de la capa MAC de IEEE 802.15.4 en el simulador Cooja resaltan las siguientes conclusiones:

- Contiki OS. no provee un mecanismo de acceso al medio puro de IEEE 802.15.4 como el presentado por el simulador validado de Omnet++. Contiki OS adapta el mecanismo CSMA/CA para tener una capa de ciclo de trabajo, agregando mecanismos extras como una cola para la administración de los paquetes salientes. A pesar de estas adaptaciones que ContikiOs presenta, se logró obtener resultados similares con los correspondientes al simulador Omnet++.

- Los resultados de los métodos de validación propuestos en esta tesis mostrados en las secciones (4.6.1 y 4.6.2), demuestran que el modelo de simulación de la capa MAC de Cooja se encuentra validado y operable como herramienta fundamental para alcanzar los objetivos de esta tesis y realizar proyectos futuros que utilicen el estándar IEEE 802.15.4.

### 6.3 Conclusiones sobre formación de la red SDWSN en comparación con RPL e impacto de ciclo de trabajo.

Un resumen de las conclusiones más significativas de la evaluación de la formación de las SDWSN y el impacto de agregar ciclo de trabajo al radio se enlista a continuación:

- Utilizar ciclo de trabajo en el radio sobre las configuraciones observadas reduce el consumo de energía al formar la red con un máximo del 94% tanto en SDWSN como en RPL. Sin embargo el costo de esta reducción de energía se ve reflejado en incrementar el tiempo de convergencia y la cantidad de overhead, de tal forma que la cantidad de paquetes se incrementa conforme la frecuencia  $f_{RDC}$  disminuye.
- La correcta elección de la configuración de periodicidad de paquetes de control hará que la red SDWSN converja en un menor tiempo y con un menor consumo de energía en comparación con RPL. Sin embargo, la red SDWSN requiere mayor cantidad de paquetes de control para la formación de la red en contraste con RPL.
- El paradigma de SDWSN demuestra mejor desempeño en términos de consumo de energía y tiempo de convergencia en comparación a RPL; para ser utilizado en la formación de topologías multisalto compuestas por un número de nodos reducido (ver sección 4.5.3) y con configuraciones de  $f_{RDC}$  a 32, 64 Hz y nullrdc. Para la formación de topologías tipo malla a un salto (ver sección 4.5.2) con las configuraciones  $f_{RDC}$  anteriores, el paradigma SDWSN presenta mejor desempeño en términos de consumo de energía en comparación con RPL, pero con un tiempo de convergencia similar. Sin embargo, para la formación de una topología multisalto con mayor cantidad de nodos (ver sección 4.5.4), la formación de la red con RPL muestra mejor desempeño en términos de consumo de energía, tiempo de convergencia y overhead generado, en comparación de la formación de la red con SDWSN.

- A pesar de que el enfoque de SDWSN indica dar soporte a tener una capa de RDC en su pila de protocolos. En el capítulo 4.7.1 se demostró que es necesario un mecanismo de ACK para los paquetes OpenPath, esto para incrementar la tasa de éxito de formación de la red con SDWSN. En este sentido, para la frecuencia de 8Hz, incluso con el mecanismo de ACK no fue posible lograr la convergencia de la red SDWSN.
- Al reducir la frecuencia del ciclo de trabajo a 8 y 16 Hz, se espera una reducción del consumo de energía con respecto a frecuencias más altas. Sin embargo, para redes SDWSN se observó que no sucede así. Esto se puede explicar considerando que la pérdida de paquetes Openpath involucra mayor overhead, y por lo tanto mayor consumo de energía. Esto demuestra que la red SDWSN no presenta soporte completo para utilizar frecuencias bajas de  $f_{RDC}$ . En cambio la red RPL muestra un mejor desempeño ante este tipo de frecuencias, donde si bien el tiempo de convergencia de la red aumenta, el impacto no es mayor como en la red SDWSN, presentando así un menor consumo de energía y menor cantidad de paquetes de control.

## 6.4 Trabajo a futuro

Este trabajo permite conocer las configuraciones adecuadas que proporcionen un buen desempeño en consumo de energía y tiempo de convergencia al formar una red SDWSN cuando se agrega ciclo trabajo al radio en comparación con RPL. Este trabajo puede ser considerado para mejorar los mecanismos existentes del paradigma de SDN en la WSN y contribuir a la formación de un estándar como el presentado en RPL. El paradigma de SDWSN tiene unos cuantos años en desarrollo, durante los cuales se han hecho esfuerzos para generar una solución holística que permita brindar solvencia a los retos de las WSN.

Aunque los objetivos de este trabajo de investigación fueron alcanzados, la evaluación de abordar los mecanismos de ciclo de trabajo en el proceso de formación de una red SDWSN puede explorar ciertas áreas en específico que pueden complementar el trabajo presentado. En esta sección se detallan las recomendaciones para el trabajo futuro que puede extender el trabajo realizado:

- La formación de la red para SDWSN tiene una generación de paquetes de control alta en comparación con el enrutamiento de RPL, por lo que se considera que existe un área de oportunidad para investigar sobre el uso de algoritmos y técnicas de optimización para reducir la cantidad de overhead generado mediante una implementación adaptativa que modifique los periodos de envío de paquetes de control dependiendo del estado de la red.
- La movilidad es inherente a las WSN para el Internet de las Cosas, por ello se recomienda considerar una evaluación de este aspecto en la formación de la red.
- Gracias a que el paradigma de SDN-WISE es un enfoque basado en el estado de la red, se abre la oportunidad a utilizar técnicas de aprendizaje profundo reforzado para determinar la existencia de contención en la red y disminuir la periodicidad de los paquetes de control cuando se utilizan frecuencias  $f_{RDC}$  bajas en los nodos que han aprendido su ruta.



## Literatura citada

---

- Ahmed, O. 2017. Evaluation assessment of energy-efficient duty cycling MAC protocols in ContikiOS. Tesis de Maestría en Ciencias. Abo Akemic University. Finlandia. 71 pp.
- Akbari, S. 2014. Energy harvesting for wireless sensor networks review. 2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014, 2, 987–992. doi:10.15439/2014F85
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E. 2002. Wireless sensor networks: a survey. *Computer Networks*, 38, 393–422.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M. 2015. Internet of things: a survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys and Tutorials*, 17(4), 2347–2376. doi:10.1109/COMST.2015.2444095
- Ali, H. 2012. A Performance evaluation of RPL in Contiki. Tesis de Maestria en Ciencias. Swedish Instiute of Computer Science. 91 pp.
- Alves, R. C. A., Oliveira, D. A. G., Pereira, G. C. C. F., Albertini, B. C., Margi, C. B. 2018. WS3N: wireless secure SDN-based communication for sensor networks. *Security and Communication Networks*, 2018. doi:10.1155/2018/8734389
- Anadiotis, A. C. G., Galluccio, L., Milardo, S., Morabito, G., Palazzo, S. 2015. Towards a software-defined Network Operating System for the IoT. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, 579–584. doi:10.1109/WF-IoT.2015.7389118
- Aregawi-Beyene, M. 2017. Evaluation of SDN in small wireless- capable and resource-constrained devices. Tesis de Maestría en Ciencias. Norwegian University of Science and Technology. 137 pp.
- Bajaj, R. K., Rao, M., Agrawal, H. 2018. Internet of Things ( IoT ) in the smart automotive sector : a review. *IOSR Journal of Computer Engineering*, 36–44. doi:2278-8727
- Banikazemi, M., Olshefski, D., Shaikh, A., Tracey, J., Wang, G. 2013. Meridian: an SDN platform for cloud network services. *IEEE Communications Magazine*, 51(2), 120–127. doi:10.1109/MCOM.2013.6461196
- Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006, October). X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems* (pp. 307-320). ACM. Buratti, C., Stajkic, A., Gardasevic, G., Milardo, S., Abrignani, M. D., Mijovic, S., Member, G. S., Morabito, G., Verdone, R. 2016. Testing Protocols for the Internet of Things on the EuWIn Platform. *IEEE Internet of Things Journal*, 3(1), 124–133. doi:10.1109/JIOT.2015.2462030
- Capella, J. V., Campelo, J. C., Bonastre, A., Ors, R. 2016. A reference model for monitoring iot WSN-based applications. *Sensors (Switzerland)*, 16(11). doi:10.3390/s16111816
- Carrano, R. C., Passos, D., Magalhaes, L. C. S., Albuquerque, C. V. N. 2014. Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 16(1), 181–194. doi:10.1109/SURV.2013.052213.00116
- Casillas Gaxiola, M. O. 2012. Diseño de una red híbrida de dos saltos para aplicaciones de redes de sensores

en telemonitoreo y telemedicina. Tesis de Maestría en Ciencias. Centro de Investigación Científica y de Educación Superior de Ensenada. 139 pp.

- Charfi, F., Bouyahi, M. 2012. Performance evaluation of beacon-enabled IEEE 802.15.4 Under NS2. *International Journal of Distributed and Parallel systems*, 3(2), 67–79. doi:10.5121/ijdps.2012.3206
- Costanzo, S., Galluccio, L., Morabito, G., Palazzo, S. 2012. Software defined wireless networks: Unbridling SDNs. *Proceedings - European Workshop on Software Defined Networks, EWSDN 2012*, 1–6. doi:10.1109/EWSDN.2012.12
- Crossbow. s/f. MICAz Wireless Measurement System. Recuperado de: [http://www.openautomation.net/uploadsproductos/micaz\\_datasheet.pdf](http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf)
- De Gante, A., Aslan, M., Matrawy, A. 2014. Smart wireless sensor network management based on software-defined networking. *2014 27th Biennial Symposium on Communications, QBSC 2014*, 71–75. doi:10.1109/QBSC.2014.6841187
- Dely, P., Kassler, A., Bayer, N. 2011. OpenFlow for wireless mesh networks. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, 1–6. doi:10.1109/ICCCN.2011.6006100
- Dludla, A. G., Abu-Mahfouz, A. M., Kruger, C. P., Isaac, J. S. 2013. Wireless Sensor Networks Testbed: ASNTbed. *Proceedings of the IST-Africa 2013 Conference*, 1–10.
- Dunkels, A., Gronvall, B., & Voigt, T. (2004, November). Contiki-a lightweight and flexible operating system for tiny networked sensors. In *29th annual IEEE international conference on local computer networks* (pp. 455-462). doi:10.1109/LCN.2004.38
- Dunkels, Adam. 2011. The contikimac radio duty cycling protocol. SICS Technical Report T2011:13.
- Dunkels, Adam, Eriksson, J., Finne, N., Tsiftes, N. 2011. Powertrace : Network-level Power Profiling for Low-power Wireless Networks Low-power Wireless. SICS Technical Report T2011:05, 14.
- Dunkels, Adam, Schmidt, O., Voigt, T., Ali, M. 2006. Protothreads: Simplifying event-driven programming of memory-constrained embedded systems. *SenSys'06: Proceedings of the Fourth International Conference on Embedded Networked Sensor Systems*, 29–42. doi:10.1145/1182807.1182811
- Farej, Z., Abdul-hameed, A. M. 2015. Performance Comparison among ( Star , Tree and Mesh ) Topologies for Large Scale WSN based IEEE 802.15.4 Standard. *124(6)*, 41–44.
- Fernando, F., Lasso, J., Clarke, K., Nirmalathas, A. 2018. A software-defined networking framework for IoT based on 6LoWPAN. 1–7. doi:10.1109/WTS.2018.8363938
- Flauzac, O., Gonzalez, C., Nolot, F. 2015. SDN Based architecture for clustered WSN. *Proceedings - 2015 9th International Conference on innovative mobile and internet services in ubiquitous computing, IMIS 2015*, 342–347. doi:10.1109/IMIS.2015.52
- Fotouhi, H., Vahabi, M., Ray, A., Björkman, M. 2016. SDN-TAP: An SDN-based traffic aware protocol for wireless sensor networks. *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services, Healthcom 2016*. doi:10.1109/HealthCom.2016.7749527

- Galluccio, L., Milardo, S., Morabito, G., Palazzo, S. 2015. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIRELESS SENSOR networks. Proceedings - IEEE INFOCOM, 26, 513–521. doi:10.1109/INFOCOM.2015.7218418
- Ganawali, O., Levis, P. 2012. The minimum rank with hysteresis objective function. Recuperado de: <http://www.hjp.at/doc/rfc/rfc6719.html>
- Garcia-Loya, E. 2018. Medición del impacto de la capacidad en el enrutamiento sobre el estándar 6LOWPAN. Tesis de licenciatura. Universidad Autónoma de Baja California. 59 pp.
- Giulio, M. 2016. Enabling software defined networking in 6LoWPAN network for the Internet of Things : design , prototyping and experimentation. Tesis de Maestría en Ciencias. Università di Pisa. 120 pp.
- Giusto, D., Iera, A., Morabito, G., Atzori, L. 2010. The Internet of Things 20th Tyrrhenian Workshop on Digital Communications. En The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications. doi:10.1007/978-1-4419-1674-7
- Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. doi:10.1016/j.future.2013.01.010
- Hasenfratz, D., Saukh, O., Walser, C., Hueglin, C., Fierz, M., Arn, T., Beutel, J., Thiele, L. 2014. Deriving high-resolution urban air pollution maps using mobile sensor nodes. Pervasive and Mobile Computing, 16(PB), 268–285. doi:10.1016/j.pmcj.2014.11.008
- Hieu, N. Q., Huu Thanh, N., Huong, T. T., Quynh Thu, N., Quang, H. Van. 2018. Integrating trickle timing in software defined WSNs for energy efficiency. 2018 IEEE 7th International Conference on Communications and Electronics, ICCE 2018, 75–80. doi:10.1109/CCE.2018.8465747
- Huang, R., Chu, X., Zhang, J., Hu, Y. H. 2015. Energy-Efficient Monitoring in Software Defined Wireless Sensor Networks Using Reinforcement Learning: A Prototype. International Journal of Distributed Sensor Networks, 2015(October). doi:10.1155/2015/360428
- Hussain, A., Wenbi, R., Da Silva, A. L., Nadher, M., Mudhish, M. 2015. Health and emergency-care platform for the elderly and disabled people in the Smart City. Journal of Systems and Software, 110, 253–263. doi:10.1016/j.jss.2015.08.041
- I. A. N. A. 2011. Internet Control Message Protocol version 6 (ICMPv6) Parameters. Recuperado de <https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-parameters-2>
- IEEE Computer Society. 2003. 802.15.4 Standard. Recuperado de: <https://www.iith.ac.in/~tbr/teaching/docs/802.15.4-2003.pdf>
- IETF. 2012. Loop free DODAG local repair. de draft-guo-roll-loop-free-dodag-repair-00 Recuperado de: <https://tools.ietf.org/html/draft-guo-roll-loop-free-dodag-repair-00>
- IETF. 2012. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. de Request For Comments #6550 Recuperado de: <https://tools.ietf.org/html/rfc6550>

- Intanagonwiwat, C., Estrin, D., Govindan, R., Heidemann, J. 2002. Impact of network density on data aggregation in wireless sensor networks. *Proceedings 22nd International Conference on Distributed Computing Systems*, 457–458. doi:10.1109/icdcs.2002.1022289
- Internet Engineering Task Force (IETF). 2011. The trickle algorithm. de RFC6206: The trickle algorithm  
Recuperado de: <https://tools.ietf.org/html/rfc6206>
- Jacobsson, M., Orfanidis, C. 2015. Using Software-defined Networking Principles for Wireless Sensor Networks. In: *Proc. 11th Swedish National Computer Networking Workshop (SNCNW 2015) Karlstad, May 28-29, 2015, (Sncnw)*, 1–5.
- Jacquet, P., Uhlethaler, P. M., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L. 2001. Optimized link state routing protocol. *IEEE Inmic'01, IEEE*, 62–68.
- Jammal, M., Singh, T., Shami, A., Asal, R., Li, Y. 2014. Software defined networking: State of the art and research challenges. *Computer Networks*, 72, 74–98. doi:10.1016/j.comnet.2014.07.004
- Jurdak, R., Baldi, P., Lopes, C. V. 2007. Adaptive low power listening for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 6(8), 988–1004. doi:10.1109/TMC.2007.1037
- Karl, H., Willig, A. 2006. Protocols and architectures for wireless sensor networks. En *Protocols and Architectures for Wireless Sensor Networks*. doi:10.1002/0470095121
- Kechiche, I., Bousnina, I., Samet, A. 2017. A comparative study of RPL objective functions. 2017 6th International Conference on Communications and Networking, ComNet 2017 - Proceedings, 2018-Janua, 1–6. doi:10.1109/COMNET.2017.8285595
- Kevin Fall. 2011. The ns Manual-The VINT Project.
- Kim, H. W., Cho, H. S. 2017. Sounet: Self-organized underwater wireless sensor network. *Sensors (Switzerland)*, 17(2). doi:10.3390/s17020283
- Ko, J., Eriksson, J., Tsiftes, N., Dawson-Haggerty, S., Terzis, A., Dunkels, A., & Culler, D. (2011, April). Contikirpl and tinyrpl: Happy together. In *Workshop on Extending the Internet to Low Power and Lossy Networks (IP+ SN)* (Vol. 570).
- Ko, J. G., Cho, Y. H., Kim, H. 2006. Performance evaluation of IEEE 802.15.4 MAC with different backoff ranges in wireless sensor networks. 2006 IEEE Singapore International Conference on Communication Systems, ICCS 2006, (November). doi:10.1109/ICCS.2006.301525
- Ko, J., Lu, C., Srivastava, M. B., Stankovic, J. A., Terzis, A., Welsh, M. 2010. Wireless sensor networks for healthcare. *Proceedings of the IEEE*, 98(11), 1947–1960. doi:10.1109/JPROC.2010.2065210
- Kobo, H. I., Abu-Mahfouz, A. M., Hancke, G. P. 2017. A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements. *IEEE Access*, 5, 1872–1899. doi:10.1109/ACCESS.2017.2666200
- Kodali, R. K., Malothu, V. K. 2016. MIXIM framework simulation of WSN with QoS. *Proceedings of 2016 International Conference on Advanced Communication Control and Computing Technologies, ICACCCT 2016, (May 2016)*, 128–131. doi:10.1109/ICACCCT.2016.7831614

- Kreutz, D., Ramos, F., Esteves, P., Esteve, C., Azodolmolky, S., Uhlig, S. 2015. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1), 14–76. doi:10.1109/JPROC.2014.2371999
- Li, L. E., Mao, Z. M., Rexford, J. 2012. Toward software-defined cellular networks. *Proceedings - European Workshop on Software Defined Networks, EWSDN 2012*, 7–12. doi:10.1109/EWSDN.2012.28
- Lin, E. A., Rabaey, J. M., Wolisz, A. 2004. Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks. (June).
- Lohan, V., Singh, R. P. 2019. Home Automation Using Internet of Things. *Lecture Notes in Networks and Systems*, 39, 293–301. doi:10.1007/978-981-13-0277-0\_24
- Luo, T., Tan, H. P., Quek, T. Q. S. 2012a. Sensor openflow: Enabling software-defined wireless sensor networks. *IEEE Communications Letters*, 16(11), 1896–1899. doi:10.1109/LCOMM.2012.092812.121712
- Luo, T., Tan, H. P., Quek, T. Q. S. 2012b. Sensor openflow: enabling software-defined wireless sensor networks. *IEEE Communications Letters*, 16(11), 1896–1899. doi:10.1109/LCOMM.2012.092812.121712
- Mahmud, A., Rahmani, R. 2011. Exploitation of OpenFlow in wireless sensor networks. *Proceedings of 2011 International Conference on Computer Science and Network Technology, ICCSNT 2011*, 1, 594–600. doi:10.1109/ICCSNT.2011.6182029
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J. 2008. OpenFlow. *ACM SIGCOMM Computer Communication Review*, 38(2), 69. doi:10.1145/1355734.1355746
- McKeown, N., Anderson, T., Peterson, L., Rexford, J., Shenker, S., Louis, S. 2008. OpenFlow: enabling innovation in campus networks. 38(2), 69–74. doi:10.1145/1355734.1355746
- Mendivil Gastélum, L. J. 2017. Comparación de soluciones basadas en LPWAN e IEEE 802.15.4 para aplicaciones de salud móvil (“m - Health”). Tesis de Maestría. Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California, CICESE. 95 pp.
- Michel, M., Quoitin, B. 2014. Technical Report : ContikiMAC vs X-MAC performance analysis.
- Moss, D., Levis, P. 2008. BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking. *Design*, 1–12.
- Ndiaye, M., Hancke, G. P., Abu-Mahfouz, A. M. 2017. Software defined networking for improved wireless sensor network management: A survey. *Sensors (Switzerland)*, 17(5), 1–32. doi:10.3390/s17051031
- Österlind, F. 2006. A sensor network simulator for the Contiki OS. SICS Research Report.
- Österlind, F., Dunkels, A., Eriksson, J., Finne, N., Voigt, T. 2006. Cross-level sensor network simulation with COOJA. *Proceedings - Conference on Local Computer Networks, LCN*, 641–648. doi:10.1109/LCN.2006.322172

- Polastre, J., Hill, J., Culler, D. 2004. Versatile Low Power Media Access for Wireless Sensor Networks Categories and Subject Descriptors. Proceedings of the 2nd international conference on Embedded networked sensor systems SenSys 04, 3(4), 95. doi:10.1145/1031495.1031508
- Redondi, A., Chirico, M., Borsani, L., Cesana, M., Tagliasacchi, M. 2013. An integrated system based on wireless sensor networks for patient monitoring, localization and tracking. *Ad Hoc Networks*, 11(1), 39–53. doi:10.1016/j.adhoc.2012.04.006
- Reyes-Cruz, L. A. 2017. Análisis del impacto del número de usuarios y tasa de datos ofrecida en el traspaso entre resumideros de una WBAN/WPAN enfocada a aplicaciones de sistemas del cuidado de la salud. Tesis de Maestría en Ciencias. Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California. 110 pp.
- Shabbir, N., Rizwan, S. 2017. Routing protocols for wireless sensor networks (WSNs). En *Wireless Sensor Networks - Insights and Innovations*.
- Shahid, N., Aneja, S. 2012. Internet of Things: Vision, application areas and research challenges. Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017, 10(7), 583–587. doi:10.1109/I-SMAC.2017.8058246
- Sharma, S., Kumar, D., & Kishore, K. 2013. Wireless Sensor Networks-A review on topologies and node Architecture. *International Journal of Computer Sciences and Engineering*, 1(2), 19-25.
- Shnayder, V., Hempstead, M., Chen, B. R., Allen, G. W., Welsh, M. 2004. Simulating the power consumption of large-scale sensor network applications. *SenSys'04 - Proceedings of the Second International Conference on Embedded Networked Sensor Systems*, 188–200.
- Silva, H., Pereira, A., Solano, Y., T. de Oliveira, B., B. Margi, C. 2016. WARM: WSN Application development and resource management. XXXIV Simposio Brasileiro De Telecomunicações, 708–712.
- Son, B., Her, Y., Kim, J. 2006. A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains. *International Journal of Computer Science and Network Security (IJCSNS)*, 6(9), 124–130.
- Stehlik, M. (2011). *Comparison of simulators for wireless sensor networks* (Doctoral dissertation, Masarykova univerzita, Fakulta informatiky).
- Tanenbaum, A., Wetherall. 2011. *Computer Networks* (5ta ed.) Pearson.
- Texas Instruments. 2015. CC2650 - SimpleLink™ Multistandard Wireless MCU. pp. 1–59.
- The Open Networking Foundation. 2017. Software-defined networking (SDN). Recuperado de: <https://www.opennetworking.org/sdn-definition/>
- Thubert, P. 2012. Objective function zero for the routing protocol for low-power and lossy networks. *RFC 6552 & (Cisco Systems)*.
- Todorovic, Ivana. 2017. Description of distance vector routing protocols using mathematical means. Recuperado de: [https://www.researchgate.net/profile/Ivana\\_Vukotic/publication/320011962\\_Description\\_of\\_distance\\_vector\\_routing\\_protocols\\_using\\_mathematical\\_means/links/59c82169aca272c71bc7f0af/Description-of-distance-vector-routing-protocols-using-mathematical-means.pdf](https://www.researchgate.net/profile/Ivana_Vukotic/publication/320011962_Description_of_distance_vector_routing_protocols_using_mathematical_means/links/59c82169aca272c71bc7f0af/Description-of-distance-vector-routing-protocols-using-mathematical-means.pdf)

- Trevizan De Oliveira, B., Batista Gabriel, L., Borges Margi, C. 2015. TinySDN: enabling multiple controllers for software-defined wireless sensor networks. *IEEE Latin America Transactions*, 13(11), 3690–3696. doi:10.1109/TLA.2015.7387950
- Tsapardakis, E., Ojo, M., Chatzimisios, P., Giordano, S. 2018. Performance evaluation of SDN and RPL in Wireless Sensor Networks. 2018 Global Information Infrastructure and Networking Symposium, GIIS 2018, (February 2019), 1–5. doi:10.1109/GIIS.2018.8635599
- Tsiftes, N., Eriksson, J., Dunkels, A. 2010. Low-power wireless IPv6 routing with ContikiRPL. 4–5.
- Tsvetkov, T. 2011. RPL: IPv6 Routing Protocol for Low Power and Lossy Networks. Seminar Sensorknoten: Betrieb, Netze und Anwendungen SS2011 Lehrstuhl Netzarchitekturen und Netzdienste, Lehrstuhl Betriebssysteme und Systemarchitektur, (July), 67–73. doi:10.2313/NET-2011-07-1
- Varga, A., & Hornig, R. (2008, March). An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops* (p. 60). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- Wang, Y., Chen, H., Wu, X., Shu, L. 2016. An energy-efficient SDN based sleep scheduling algorithm for WSNs. *Journal of Network and Computer Applications*, 59, 39–45. doi:10.1016/j.jnca.2015.05.002
- Wei, Y., Heidemann, J., Estrin, D. 2002. An Energy efficient MAC protocol for wireless sensor networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10602 LNCS(c), 422–429. doi:10.1007/978-3-319-68505-2\_36
- Winter, T., Thubert, P., Kelsey, R., Brandt, A., Clausen, T., Hui, J., Levis, P., Pister, K., Struik, R., Vasseur, J. 2011. RPL: IPv6 routing protocol for low power and lossy networks draft-ietf-roll-rpl-18.
- Xu, L. Da, He, W., Li, S. 2014. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233–2243. doi:10.1109/TII.2014.2300753
- Xue, X., Gonzalez-Argueta, L., Sundararajan, V. 2009. Energy scavenging for Wireless Sensor Networks. (September), 361–366. doi:10.1115/detc2007-35829
- Yacchirema, D. C., Esteve, M. 2017. Enable IoT interoperability in ambient assisted living : active and healthy aging scenarios. 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), 53–58. doi:10.1109/CCNC.2017.7983081
- Ye, W., Heidemann, J., Estrin, D. 2004. Medium access control with coordinated adaptive sleeping for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 12(3), 493–506.
- Zeng, D., Li, P., Guo, S., Miyazaki, T., Hu, J., Xiang, Y. 2015. Energy minimization in multi-task software-defined sensor networks. *IEEE Transactions on Computers*, 64(11), 3128–3139. doi:10.1109/TC.2015.2389802
- Zhou, H., Guan, X. 2009. Idle-listening reduction for data aggregation in distributed sensor networks. *IEEE Transaction on Parallel and Distributed Systems*. 1–13.
- Zolertia. 2010. Z1 Datasheet. pp. 1–20. Recuperado de: [http://zolertia.sourceforge.net/wiki/images/e/e8/Z1\\_RevC\\_Datasheet.pdf](http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf)