



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES

TEMA:

**Evaluación de una red definida por software para redes de sensores  
inalámbricos utilizando el protocolo de descubrimiento de topología y  
manejo de paquetes**

AUTOR:

Ing. García Vásquez, Gabriel Santiago

Trabajo de Titulación previo a la obtención del Grado Académico de  
**MAGÍSTER EN TELECOMUNICACIONES**

TUTOR:

M. Sc. Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

21 de febrero del 2020



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Magíster **García Vásquez, Gabriel Santiago** como requerimiento parcial para la obtención del Grado Académico de **MAGÍSTER EN TELECOMUNICACIONES**.

TUTOR

---

M. Sc. Palacios Meléndez, Edwin Fernando

DIRECTOR DEL PROGRAMA

---

M. Sc. Romero Paz, Manuel de Jesús

Guayaquil, 21 de febrero del 2020



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **García Vásquez, Gabriel Santiago**

**DECLARÓ QUE:**

El trabajo de titulación: **“Evaluación de una red definida por software para redes de sensores inalámbricos utilizando los protocolos de descubrimiento de topología y manejo de paquetes”**, previa a la obtención del grado Académico de **Magíster en Telecomunicaciones**, ha sido desarrollado, respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de la tesis del Grado Académico en mención.

Guayaquil, 21 de febrero del 2020

EL AUTOR

---

García Vásquez, Gabriel Santiago



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES

**AUTORIZACIÓN**

Yo, **García Vásquez, Gabriel Santiago**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución de la Tesis de Maestría titulada: “**Evaluación de una red definida por software para redes de sensores inalámbricos utilizando los protocolos de descubrimiento de topología y manejo de paquetes**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 21 de febrero del 2020

EL AUTOR

---

García Vásquez, Gabriel Santiago

# REPORTE DE URKUND

The screenshot shows the URKUND interface. On the left, a document summary is displayed:

- Documento:** [Gabriel\\_Garcia\\_MET2019.docx](#) (D59707659)
- Presentado:** 2019-11-26 18:32 (-05:00)
- Presentado por:** fernandopm23@hotmail.com
- Recibido:** edwin.palacios.ucsg@analysis.orkund.com
- Mensaje:** Revisión TT MET Gabriel Garcia [Mostrar el mensaje completo](#)  
1% de estas 26 páginas, se componen de texto presente en 2 fuentes.

On the right, a 'Lista de fuentes' (List of sources) panel is visible, showing a table of sources:

Categoría	Enlace/nombre de archivo	
	<a href="#">Roberto_Dender_final.docx</a>	<input checked="" type="checkbox"/>
	<a href="#">TESIS ARMANDO ESPARZA V4.docx</a>	<input type="checkbox"/>
	<a href="#">TESIS NATALIA ROMERO Final.docx</a>	<input checked="" type="checkbox"/>
<b>Fuentes alternativas</b>		
	<a href="#">TESIS NATALIA ROMERO Final.docx</a>	<input type="checkbox"/>
	<a href="#">FUENTES_PABLO_FINAL.docx</a>	<input type="checkbox"/>

At the bottom of the interface, there is a toolbar with icons for search, navigation, and actions like '0 Advertencias', 'Reiniciar', 'Exportar', and 'Compartir'.

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

TEMA: Evaluación de una red definida por software para redes de sensores inalámbricos utilizando el protocolo de descubrimiento de topología y manejo de paquetes

AUTOR: Ing. García Vásquez, Gabriel Santiago

Trabajo de Titulación previo a la obtención del Grado Académico de Magister en Telecomunicaciones

TUTOR: M. Sc. Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

19 de noviembre del 2019

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO MAESTRÍA EN TELECOMUNICACIONES

## **Dedicatoria**

Se lo dedico al forjador de mi camino, mi padre celestial, el que me acompaña y siempre me levanta de mi continuo tropiezo al creador de mis padres y de las personas que más amo, con mi más sincero amor.

García Vásquez, Gabriel Santiago

## **Agradecimientos**

Al finalizar este trabajo quiero utilizar este espacio para agradecer a Dios por todas sus bendiciones, a mis Padres, hijos y todas las personas especiales en mi vida.

También quiero agradecer a la Universidad Católica Santiago de Guayaquil, directivos y profesores por la organización del programa de Maestría en Telecomunicaciones.

García Vásquez, Gabriel Santiago



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

SISTEMA DE POSGRADO  
MAESTRÍA EN TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_

**M. SC. PALACIOS MELÉNDEZ, EDWIN FERNANDO**  
TUTOR

f. \_\_\_\_\_

**M. SC. CÓRDOVA RIVADENEIRA, LUIS SILVIO**  
REVISOR

f. \_\_\_\_\_

**M. SC. ZAMORA CEDEÑO, NESTOR ARMANDO**  
REVISOR

f. \_\_\_\_\_

**M. SC. ROMERO PAZ MANUEL DE JESÚS**  
DIRECTOR DEL PROGRAMA



## ÍNDICE GENERAL

Capítulo 1: Generalidades del proyecto de grado.....	2
1.1. Introducción.....	2
1.2. Antecedentes.....	3
1.3. Definición del problema .....	3
1.4. Justificación del problema a investigar. ....	3
1.5. Objetivos.....	3
1.6. Hipótesis.....	4
1.7. Metodología de investigación. ....	4
Capítulo 2: Fundamentación Teórica. ....	6
2.1. Antecedentes técnicos.....	6
2.2. Redes definidas por software. ....	7
2.3. Redes de sensores inalámbricos definidas por software.....	9
2.3.1. Redes inalámbricas definidas por software. ....	10
2.4. Comparación de protocolos del Internet de las Cosas (IoT).....	14
2.5. Trabajos relacionados a SDWN. ....	15
2.6. Protocolo ZigBee. ....	16
2.7. Protocolo 6LoWPAN.....	19
2.7.1. Pila de protocolos.....	19
2.7.2. Protocolo de enrutamiento. ....	20
Capítulo 3: Desarrollo de la propuesta.....	23
3.1. Descripción general de SD-WSN. ....	23
3.2. Descripción general de SD-WSN. ....	23
3.2.1. Requisitos para SD-WSN.....	24
3.2.2. Enfoque de SD-WSN.....	25
3.2.3. Arquitectura de protocolo SD-WSN.....	27
3.3. Detalles del protocolo SD-WSN.....	30
3.3.1. Descubrimiento de topología (TD).....	30
3.3.2. Manejo de paquetes. ....	32

3.4. Prototipo y banco de pruebas.....	37
3.5. Evaluación del desempeño de la SD-WSN. ....	39
Conclusiones .....	47
Recomendaciones .....	48
Bibliografía .....	49

## ÍNDICE DE FIGURAS

### Capítulo 2:

Figura 2. 1: Pila de protocolos SDWN. ....	11
Figura 2. 2: Arquitecturas de protocolo (a) Izquierda: SDWN, (b) Centro: 6LoWPAN, y (c) Derecha: ZigBee. ....	17

### Capítulo 3:

Figura 3. 1: Pila de protocolos SD-WSN. ....	29
Figura 3. 2: Encabezado de paquete WSN. ....	33
Figura 3. 3: Tabla de flujo de WSN. ....	33
Figura 3. 4: Topología ejemplar. ....	35
Figura 3. 5: Máquina de estado finito que implementa una política tal que los paquetes generados por A se descartan si los últimos datos medidos por B son inferiores (o iguales) a $xThr$ . ....	36
Figura 3. 6: Opciones de implementación de SD-WSN. ....	38
Figura 3. 7: Integración con el simulador OMNeT ++. ....	39
Figura 3. 8: Despliegue de nodos. ....	40
Figura 3. 9: CDF del RTT para diferentes tamaños de carga útil y diferentes distancias entre el nodo de origen y el de destino. ....	41
Figura 3. 10: RTT promedio frente al tamaño de la carga útil, para diferentes valores del número de saltos. ....	42
Figura 3. 11: Desviación estándar de los valores de RTT frente al tamaño de la carga útil, para diferentes valores del número de saltos. ....	42
Figura 3. 12: CDF del RTT en el caso de multidifusión para diferentes tamaños de carga útil. ....	43
Figura 3. 13: Eficiencia para diferentes valores de entrada de TTL en tabla de flujo WSN. ....	44
Figura 3. 14: Eficiencia para diferentes valores del período de envío de balizas. ....	44
Figura 3. 15: Tiempos de respuesta del controlador para diferentes topologías utilizando 50 nodos. ....	45

Figura 3. 16: Tiempos de respuesta del controlador para diferentes topologías utilizando 60 nodos. ....	45
Figura 3. 17: Tiempos de respuesta del controlador para diferentes topologías utilizando 70 nodos. ....	46

## Resumen

El presente trabajo de examen complejo consiste en evaluar una red definida por software (SDN) para redes de sensores inalámbricos (WSN) utilizando los protocolos de descubrimiento de topología y manejo de paquetes. Actualmente, muchos profesionales realizan investigaciones orientadas a redes definidas por software, aunque existe otro grupo de investigaciones donde el enfoque son las redes inalámbricas definidas por software. En el capítulo 1, se presenta la definición y justificación del problema a investigar, así como, el objetivo general, objetivos específicos, la hipótesis y tipo de investigación utilizado en el presente trabajo investigativo. En el capítulo 2, se presenta los fundamentos teóricos de las redes definidas por software y de redes inalámbricas definidas por software. En el capítulo 3, se explica la red definida por software para redes de sensores (SD-WSN) y se presenta los resultados obtenidos de los escenarios realizados en OMNeT ++. Finalmente, se presentan las conclusiones y recomendaciones del presente trabajo.

**Palabras claves:** REDES, SOFTWARE, PROTOCOLOS, TOPOLOGÍAS, SENSORES, INALÁMBRICA

## **Abstract**

The present complex examination work consists in evaluating a software-defined network (SDN) for wireless sensor networks (WSN) using topology discovery protocols and packet handling. Currently, many professionals conduct research oriented to software-defined networks, although there is another group of research where the focus is software-defined wireless networks. In chapter 1, the definition and justification of the problem to be investigated is presented, as well as the general objective, specific objectives, the hypothesis and type of research used in this research work. In Chapter 2, the theoretical foundations of software-defined networks and software-defined wireless networks are presented. In chapter 3, the software-defined network for sensor networks (SD-WSN) is explained and the results obtained from the scenarios in OMNeT ++ are presented. Finally, the conclusions and recommendations of this work are presented.

**Keywords:** NETWORKS, SOFTWARE, PROTOCOLS, TOPOLOGIES, SENSORS, WIRELESS

## Capítulo 1: Generalidades del proyecto de grado.

### 1.1. Introducción.

La literatura reconoce que las redes definidas por software (*Software Defined Network, SDN*) enfrentan un gran desafío con respecto a los conocimientos actuales que existen en los centros de datos. Según Valdivieso C., Barona L., & Garcia V., (2013), la integración de SDN en una empresa necesita una estimación de la cantidad necesaria de controladores mediante la determinación de la topología y la localización de estos controladores. Además, la falta de conocimientos puede generar un alto riesgo de seguridad, ya que el controlador centralizado de una red definida por software puede ser muy vulnerable en comparación con las redes tradicionales.

SDN es una idea que promete simplificar y unificar la forma en que se diseñan y administran las redes. SDN emplea una abstracción que separa el control del plano de datos e introduce la noción de un controlador lógicamente centralizado. Estos conceptos brindan beneficios como una mayor flexibilidad en la administración de la red, unificación de tecnologías heterogéneas y menores gastos operativos, que abren muchas oportunidades para trabajar en nuevas soluciones de red innovadoras. (Foukas et al., 2015)

SDN es la tecnología de administración de red centralizada que podría reducir la sobrecarga de la administración de la red y la aplicación de políticas en la red IP tradicional. El controlador SDN es el sistema operativo de la red responsable de todas las operaciones de la red. (Priya & Radhika, 2019)

El paradigma de SDN es un cambio de paradigma que se hizo frente a estos desafíos, ya que brinda un control centralizado para definir políticas para la gestión y el enrutamiento del tráfico y la aplicación distribuida de estas políticas, tales como control de flujo, lógica de reenvío, QoS de la red. En la infraestructura SDN, la capa de control está centralizada, que es la raíz de la confianza y cierto nivel de confianza de identifica se distribuye a través de múltiples componentes SDN. (Najeem & Krishnan, 2019)

## **1.2. Antecedentes.**

Durante la búsqueda de información en la base de datos de los trabajos de investigación realizados por los maestrantes de la Maestría en Telecomunicaciones se encontraron trabajos relacionados a SDN y al empleo de herramientas de simulación utilizando plataformas, tales como, Opnet Modeler, NS-2, NS-3, Matlab, entre otras. También, fue necesaria la búsqueda de información en revistas científicas donde investigadores se enfocan en mostrar sus resultados de redes definidas por software.

## **1.3. Definición del problema**

Necesidad de evaluar una red definida por software para redes de sensores inalámbricos utilizando los protocolos de descubrimiento de topología y manejo de paquetes.

## **1.4. Justificación del problema a investigar.**

En la actualidad es de suma importancia la integración de SDN en la arquitectura de redes convencionales en especial en redes de sensores inalámbricos. Es decir, que controlador lógicamente centralizado permite mejorar el proceso de decisión de políticas, distribuyendo el proceso de aplicación de políticas a través de los conmutadores. Por ejemplo, en las redes heredadas, las funciones de red completas se implementan en forma de dispositivos o cajas intermedias independientes y, por lo tanto, se implementan de forma independiente. La red definida por software es una técnica en ascenso para tratar el control y la administración de la red de patrimonio de reemplazo (hardware acoplado y programa de software) separando el plano de control (programa de software) del plano de información (hardware). Da adaptabilidad a los ingenieros al influir en el control focal al plano directamente programable.

## **1.5. Objetivos**

### **1.5.1. Objetivo General:**

Evaluar una red definida por software para redes de sensores inalámbricos utilizando los protocolos de descubrimiento de topología y manejo de paquetes mediante la plataforma de simulación OMNeT ++.



### **1.5.2. Objetivos específicos:**

- ✓ Describir los fundamentos teóricos de las redes definidas por software, sus protocolos y arquitectura.
- ✓ Diseñar la red inalámbrica definida por software real y de los escenarios de simulación.
- ✓ Analizar los resultados obtenidos de la red definida por software para sensores inalámbricos (SD-WSN).

### **1.6. Hipótesis**

El paradigma de redes definidas por software está cambiando la forma en que las redes se conciben con implicaciones disruptivas en el diseño, implementación, operación y mantenimiento de la red. En consecuencia, el presente trabajo demostrará la importancia de utilizar redes de sensores inalámbricos definidas por software, que ha sido concebido como una forma de reducir la complejidad y aumentar la flexibilidad.

### **1.7. Metodología de investigación.**

El presente trabajo emplea la investigación descriptiva. El objetivo principal de la investigación descriptiva es la descripción del estado de cosas. La característica principal de este método es describir los fundamentos teóricos.

Existen dos enfoques básicos para la investigación, a saber, el enfoque cuantitativo y el enfoque cualitativo (no aplica para carreras de ciencias puras sino de ciencias sociales). El primero implica la generación de datos en forma cuantitativa que puede ser sometida a un análisis riguroso de manera formal y rigurosa. Este enfoque puede subclasificarse en investigación inferencial, experimental y de simulación.

El propósito del enfoque inferencial para la investigación es formar una base de datos a partir de la cual se infieren características o relaciones de la población. Esto generalmente significa encuestas donde la población es estudiada (cuestionada u observada). El propósito del enfoque experimental se caracteriza por tanto mayor control sobre el entorno de investigación y en

este caso algunas variables son manipuladas para observar su efecto sobre otras variables. Mientras, que el enfoque de simulación implica un entorno artificial. Esto permite una observación del comportamiento dinámico de un sistema (o su subsistema) bajo condiciones controladas. Dados los valores de las condiciones iniciales, los parámetros y las variables exógenas, un proceso de tiempo representa la simulación. El enfoque de simulación también puede ser útil en la construcción de modelos para comprender las condiciones futuras.

En consecuencia, el trabajo de posgrado emplea el enfoque cuantitativo, ya que se basan en la recolección de datos obtenidos durante la ejecución de las simulaciones. Y este enfoque utiliza el método experimental y de simulación. EL primero permite tener mayor control de las variables y el segundo donde se realizan los escenarios de simulación.

## Capítulo 2: Fundamentación Teórica.

### 2.1. Antecedentes técnicos.

A principios de la década del año 2000, los sistemas microelectromecánicos (Micro-Electro-Mechanical Systems, MEMS), las comunicaciones inalámbricas y la electrónica digital alcanzaron el nivel de madurez necesario para desarrollar pequeños nodos de sensores inalámbricos de bajo costo y bajo consumo capaces de comunicarse de forma inalámbrica entre sí, sin la necesidad de una infraestructura previamente implementada, es decir, para formar lo que comúnmente se conoce como redes de sensores inalámbricos (*Wireless Sensor Network, WSN*).

Todo esto fue impulsada por la promesa de que las WSNs habrían producido un impacto radical en varios escenarios de aplicación, en la última década la comunidad de investigación de redes ha dedicado un esfuerzo inmenso al estudio de las WSNs y la definición de soluciones apropiadas para ellos. Si bien dicho esfuerzo ha resultado en una comprensión profunda de la cuestión relacionada con WSN, el despliegue a gran escala esperado de WSN no ha sucedido completamente hasta hoy.

Las razones del lento despegue comercial de las WSN son múltiples. Sin embargo, en la base hay una razón técnica: las WSNs se caracterizan por requisitos profundamente diferentes según la aplicación específica y el escenario de implementación. En consecuencia, como se reconoce ampliamente, no hay algo como una solución única para WSN. En cambio, hay una gran cantidad de soluciones específicas para aplicaciones verticales que han resultado en un contexto y mercado extremadamente fragmentados.

El problema anterior se puede superar haciendo programables las WSNs y, por lo tanto, se ha dedicado un esfuerzo de investigación significativo para diseñar WSNs programables. Sin embargo, en la mayoría de las implementaciones actuales de WSNs en el mundo real, la programación suele estar estrechamente relacionada con el sistema operativo, lo que requiere que

los desarrolladores de aplicaciones se centren en detalles intensivos de bajo nivel en lugar de la lógica de la aplicación.

## **2.2. Redes definidas por software.**

El paradigma de las redes definidas por software (*Software Defined Networking, SDN*) y OpenFlow (basado en conmutadores Ethernet, con una tabla de flujo interna y una interfaz estandarizada para agregar y eliminar entradas de flujo) (McKeown et al., 2008). OpenFlow actualmente es la instancia más popular de SDN, se han propuesto recientemente para resolver problemas análogos en el dominio cableado. Gracias a las interfaces estandarizadas, que pueden funcionar en redes hechas de conmutadores heterogéneos, en OpenFlow los nodos de la red manejan los paquetes entrantes como se especifica en las denominadas tablas de flujo. Cada entrada de la tabla de flujo está relacionada con un flujo y está compuesta por tres secciones:

- (i) una regla de coincidencia que especifica los valores del campo de encabezado que deben encontrarse en los paquetes que pertenecen al flujo;
- (ii) la acción que debe ejecutarse en los paquetes del flujo (por ejemplo, soltar, reenviar a, etc.); y
- (iii) alguna información estadística sobre el flujo.

Si la tabla de flujo no contiene ninguna entrada que especifique cómo tratar un paquete determinado, el nodo envía una solicitud a una entidad de software llamada “plano de control” que tiene una abstracción de alto nivel de los elementos de la red. El controlador puede ejecutarse en un servidor remoto de forma centralizada (lógicamente). El controlador responde con la información requerida para completar una nueva entrada de la tabla de flujo para manejar el paquete.

De esta manera, OpenFlow separa claramente (incluso físicamente) el plano de datos del plano de control y entrega una red, que:

- es fácil de configurar y administrar,

- puede evolucionar porque, en principio, se pueden introducir nuevos servicios y políticas de administración en la red tan simple como instalar un nuevo software en una PC,
- un nodo de red determinado puede ser reemplazado por otro producido por cualquier proveedor, liberando al operador del bloqueo del proveedor y permitiendo el uso de hardware básico.

Lo anterior son ventajas cruciales para los operadores de red que, por lo tanto, están invirtiendo grandes esfuerzos en el dominio de las SDNs en términos de adquisición de nuevos equipos y/o desarrollo de conocimientos. Como resultado, rara vez el interés en un nuevo paradigma de redes ha aumentado a un ritmo tan rápido como sucede con SDN.

De hecho, incluso si OpenFlow estaba destinado inicialmente para las universidades y las redes de campus, los enormes beneficios introducidos en términos de simplicidad y capacidad de evolución lo hicieron adecuado para otros contextos y ahora es posible encontrar incluso dispositivos de red de nivel de operador que sean compatibles con OpenFlow.

Por lo tanto, la mayoría de los operadores de red están realizando experimentos piloto de redes OpenFlow, los fabricantes están produciendo equipos de red compatibles con OpenFlow, y la comunidad de investigación (tanto académica como industrial) está involucrada en una gran cantidad de actividades de investigación y desarrollo (I+D) relacionadas con las SDNs. De hecho, las aplicaciones de las SDNs están ahora en todas partes.

Las SDNs se han utilizado en redes de campus (backbone), redes de área amplia (*Wide Area Network, WAN*), redes de operadores, pero también en dispositivos inalámbricos y móviles. Un vistazo rápido a la lista de miembros de la Fundación de Redes Abiertas (llamada en inglés como Open Networking Foundation), cuya organización promueve el desarrollo de estándares relacionados con SDNs, siendo suficiente para comprender que esta exageración también se ha extendido al dominio inalámbrico.

### **2.3. Redes de sensores inalámbricos definidas por software**

A pesar de la gran adopción de OpenFlow y SDN en general, en el dominio cableado, no existía una solución tan ampliamente aceptada en redes inalámbricas y, especialmente, en WSN. Por ejemplo, Sensor OpenFlow fue el primer intento de implementar un protocolo SDN para WSNs. La arquitectura OpenFlow, considera que los nodos deben mantener una tabla de flujo con entradas de formato específico y predefinido. En consecuencia, Sensor OpenFlow admite el procesamiento en redes principalmente para permitir la agregación de datos, tal como se hace comúnmente en WSNs para la conservación de energía.

Hay que tener en cuenta que Sensor OpenFlow no puede admitir la amplia gama de protocolos, estándar o patentados que se han utilizado en el contexto de WSN. Además, el enfoque Sensor OpenFlow está integrado con otras técnicas de programación de WSNs. También presentaron la idea de explotar la tecnología OpenFlow para abordar la confiabilidad en WSNs. Algunas investigaciones, afirman que los sensores basados en OpenFlow son más confiables que los sensores típicos, y los resultados de simulaciones muestran que el enfoque propuesto logra un mejor rendimiento para redes grandes.

Por ejemplo, el uso de OpenFlow en una red inalámbrica enmallada permite un cambio rápido de algoritmos de reenvío y enrutamiento. Este capítulo afirma que la tecnología de redes definida por software (SDN) tendrá que enfrentar problemas relacionados con el corte, el aislamiento, el informe de estado y las transferencias, mientras que mejorará la conectividad, la calidad del servicio, la planificación, la seguridad y la localización.

En los sistemas SDNs los experimentos demuestran que se reduce el consumo de energía y proporciona un mayor nivel de flexibilidad en la gestión de la red. Por ejemplo, otra solución similar a las redes definida por software (SDN) es TinySDN, que se centra en el soporte de las operaciones de SDN en diferentes plataformas, lo que se logra construyendo sobre TinyOS. TinySDN permite la interoperabilidad de nodos habilitados para SDN con

varios controladores, y son implementados y probados con el simulador Cooja.

Sin embargo, estos enfoques se derivan directamente de OpenFlow y, por lo tanto, requieren que todos los nodos sean instruidos por el controlador para procesar los paquetes que deben manejar. Por lo tanto, implican una gran cantidad de intercambio de señalización entre los nodos y el controlador. Además, dado que cada flujo que atraviesa un nodo debe tener una entrada adecuada en una estructura de datos dada (indicada como tabla de flujo en OpenFlow) para especificar cómo distinguir los paquetes que pertenecen al flujo y cómo tratarlos, es probable que los nodos lo hagan tener que mantener grandes tablas de flujo.

### **2.3.1. Redes inalámbricas definidas por software.**

La primera implementación de una red inalámbrica definida por software (*Software Defined Wireless Networking, SDWN*) se desarrolló en octubre de 2012. La idea principal detrás del protocolo fue adaptar un enfoque centralizado, como el propuesto en las redes definidas por software convencionales a un entorno inalámbrico, dando así la oportunidad de soportar la definición flexible de reglas y cambios de topología.

La figura 2.1 muestra la pila de protocolos de una red inalámbricas definida por software (SDWN) donde las capas, tanto física (PHY) como de control de acceso al medio (*Medium Access Control, MAC*) son las correspondientes al estándar IEEE 802.15.4, mientras que las capas superiores están inspiradas en el paradigma de redes definidas por software (SDN).

Una SDWN típica está compuesta por un dispositivo controlador, un nodo receptor y varios otros nodos. El controlador recopila la información de los nodos, mantiene una representación de la red y establece rutas de enrutamiento para cada flujo de datos. El sumidero (sink) es el único nodo que está conectado directamente al controlador y actúa como puerta de enlace

para los nodos. Por lo general, el sumidero coincide con el nodo coordinador (NC) de red y su pila de protocolos es equivalente a la de un nodo genérico.

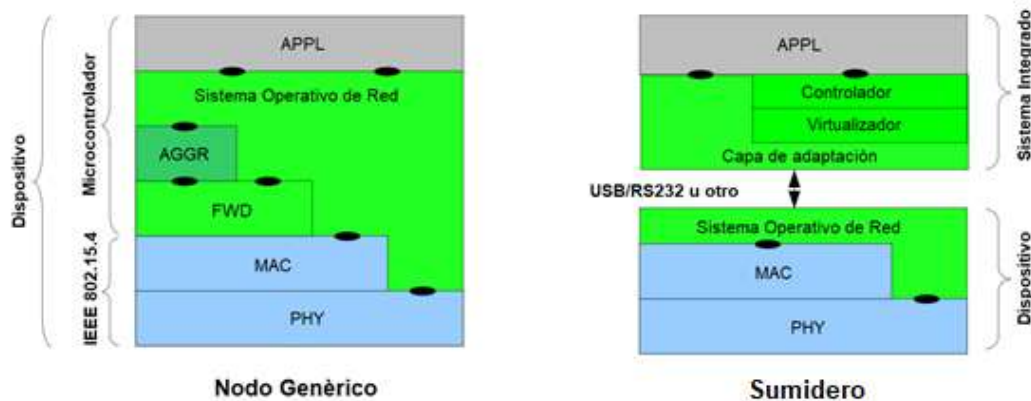


Figura 2. 1: Pila de protocolos SDWN.  
Fuente: (Filho et al., 2018)

La pila de un nodo genérico se divide en tres partes:

- Capa de reenvío (*Forwarding, FWD*);
- Capa de agregación (*Aggregation, AGGR*); y
- Sistema operativo de red (*Network Operating System, NOS*).

Mientras que la capa MAC proporciona paquetes entrantes a la capa FWD que identifica el tipo de paquete. A continuación, seis tipos diferentes de paquetes se definen de la siguiente manera:

- datos: generados (entregados) por (a) la capa de aplicación;
- beacon (baliza): enviado periódicamente en difusión por todos los nodos de la red;
- informe: contiene la lista de vecinos de un nodo;
- solicitud de regla: generada cuando recibe un paquete para el manejo del cual no tiene información (es decir, la ruta);
- respuesta de la regla: generada por el controlador como respuesta a la solicitud de la regla;
- ruta abierta: se usa para configurar una sola regla en diferentes nodos.

Cuando la capa FWD recibe un paquete sin baliza, se envía al NOS que busque la regla correspondiente en una estructura de datos apropiada llamada tabla de flujo. La tabla de flujo almacena todas las reglas que



proviene del controlador. Para cada regla, hay tres tipos de acciones que podrían ejecutarse:

- 1) Enviar a un nodo,
- 2) Modificar el paquete o descartarlo.
- 3) Si un paquete no coincide con ninguna de las reglas de la tabla, se envía una solicitud de regla al controlador.

La ruta entre el sumidero y el nodo para enviar/recibir paquetes de solicitud de reglas/respuesta debe elegirse de manera efectiva, considerando tanto la confiabilidad de la ruta como su longitud. Cada nodo almacena constantemente su distancia (en número de saltos) desde el sumidero, y el indicador de intensidad de señal recibida (*Received Signal Strength Indicator, RSSI*) que es el nivel de potencia que recibe desde el siguiente salto hacia el sumidero.

Durante la inicialización de la red, cada nodo está en un estado inactivo esperando mensajes. Cuando el sumidero se ENCIENDE, envía una baliza que contiene el número de saltos desde el sumidero (cero en este caso). Cuando un nodo A recibe la baliza, realizará las siguientes cuatro operaciones:

1. Agregar la fuente de la baliza y el RSSI recibido en la lista de nodos (tabla de vecinos) que están a un salto de distancia del nodo A.
2. Analizar la distancia contenida en la baliza y el RSSI del mensaje recibido, luego se compara estos valores con otros valores almacenados correspondientes: si el número de saltos es menor y el RSSI es mayor, la fuente de la baliza se elige como el mejor próximo salto hacia el sumidero (sink), y los valores almacenados en A se actualizan.
3. El temporizador de baliza se activa y el nodo A enviará periódicamente su propia baliza en difusión.
4. El temporizador de mensaje de informe se activa: la tabla de vecinos de A se envía periódicamente al nodo sumidero (sink) utilizando el mejor salto siguiente hacia el sumidero. Después de cada transmisión, la lista de vecinos se elimina para tener una vista actualizada de la

red. El período del informe debe ser mayor que el período utilizado para transmitir mensajes de baliza (período de baliza).

El controlador utiliza la información incluida en los mensajes de informe para crear un mapa de la red. Basado en esta representación, el controlador puede responder a las solicitudes de reglas y decidir las rutas de enrutamiento para los paquetes de datos, mientras que la solicitud de reglas seguirá la ruta descubierta previamente. La implementación real de controladores utiliza el algoritmo de enrutamiento de Dijkstra para resolver solicitudes de reglas. El peso de los bordes en la representación de topología es una función del RSSI recibido.

Un posible cambio en la red se notifica al controlador mediante mensajes de informe. Como se especificó anteriormente, el controlador obtiene periódicamente todas las listas de nodos vecinos, de acuerdo con el período del informe que está limitado por el período de baliza. Al disminuir el último período, se podría obtener una respuesta más rápida a los cambios ambientales en detrimento de tener una sobrecarga mayor.

En la implementación real de SDWN, el controlador envía una respuesta de regla solo después de recibir una solicitud de regla de un nodo y las reglas contenidas en los nodos caducan después de un período de tiempo configurable. Por lo tanto, al final de este período, el controlador recibe una nueva solicitud de regla para los paquetes inmanejables.

Como se mencionó anteriormente, se puede ejecutar más de una acción para un paquete entrante, logrando así la comunicación de multidifusión. Al realizar múltiples acciones, el controlador puede clonar un mensaje entrante en múltiples mensajes salientes. Desafortunadamente, un inconveniente de este enfoque es que la multidifusión se ejecuta localmente al transmitir una serie de mensajes de unidifusión. En otros términos, la naturaleza de transmisión de la comunicación inalámbrica no se explota. En el presente trabajo de titulación se utiliza como base el desarrollo de una versión mejorada de SDWN llamada SD-WSN que se describe en el capítulo 3.

## **2.4. Comparación de protocolos del Internet de las Cosas (IoT).**

El Internet de las cosas (*Internet of Things, IoT*) es un paradigma emergente que evoluciona en torno al concepto de cosas (objetos, automóviles, etc.), equipado con dispositivos de radio y con una dirección única. La noción de IoT ha sido reconocida por los líderes industriales y los medios de comunicación como la próxima ola de innovación, que penetra en nuestra vida diaria. Los sensores son cada vez más generalizados e intentan satisfacer las necesidades de los usuarios finales, proporcionando así la facilidad de uso en nuestras actividades cotidianas.

Los estándares comunes para aplicaciones IoT son ZigBee e IPv6 a través de redes de área personal inalámbricas de baja potencia (6LoWPAN). Ambos estándares se implementan sobre el estándar IEEE 802.15.4; sin embargo, ZigBee utiliza direcciones de control de acceso al medio (*Medium Access Control, MAC*) 802.15.4, mientras que 6LoWPAN emplea las direcciones IPv6.

Recientemente, Costanzo, Galluccio, Morabito, & Palazzo, (2012) han propuesto un tercer enfoque basado en el paradigma de red definida por software (SDN) que lo denominaron redes inalámbricas definidas por software (SDWN) y utilizaron un protocolo de enrutamiento centralizado. El coordinador/ puerta de enlace recopila información sobre el estado de la red de cosas y lleva este conocimiento a un controlador que puede decidir sobre la explotación de los recursos dentro de la red inalámbrica.

El controlador tiene una visión centralizada de la red de las cosas e incluso puede controlar cosas que se encuentran detrás de varios coordinadores/puertas de enlace. Este enfoque ofrece la ventaja potencial de una explotación óptima de los recursos, siempre que se controlen los gastos generales y el entorno no cambie con demasiada frecuencia.

El propósito de este capítulo es comparar de manera justa ZigBee con 6LoWPAN y ambos con SDWN. Pero estas tres soluciones indicadas ya fueron comparadas a través de experimentos realizados en la plataforma del

Laboratorio Europeo de Comunicaciones Inalámbricas para el Futuro de Internet (EuWIn). Es decir, que de manera particular han utilizado la instalación de banco de pruebas de topología flexible (Flextop) de EuWIn (ubicado dentro de la Universidad de Bolonia). Flextop consta de 53 nodos, equipados con interfaces IEEE 802.15.4. Flextop proporciona un entorno controlado y a priori conocido para la experimentación, lo que permite la comparación equitativa entre diferentes protocolos, a pesar de que las pruebas se realizan en instancias de tiempo diferentes.

Los resultados que se obtienen durante las pruebas experimentales son las siguientes métricas:

- (a) rendimiento,
- (b) tasa de pérdida de paquetes,
- (c) tiempo de ida y vuelta, y
- (d) los gastos generales generados en la red, considerando diferentes topologías y tamaños de red, tamaños de carga útil y condiciones ambientales, desde estáticas hasta dinámicas.

Los resultados también demostrarían que SDWN logra ser el mejor en rendimiento en términos de todas las métricas consideradas para diferentes escenarios (estáticos y cuasi-estáticos). Sin embargo, se ha observado una grave degradación del rendimiento cuando los cambios en la topología de la red son frecuentes y significativos.

## **2.5. Trabajos relacionados a SDWN.**

Muchos trabajos de investigación tratan sobre la implementación de redes ZigBee. Por ejemplo, Liang, Huang, Jiang, & Yao, (2008) se refieren a la implementación de una red ZigBee para aplicaciones de casas inteligentes, donde se mide el impacto de la interferencia de Wi-Fi en las redes ZigBee. En el trabajo realizado por Dong, Zhang, & Tong, (2014) proporcionan un análisis experimental de redes ZigBee en estrella y en árbol basadas en hardware y software disponibles comercialmente, para determinar las limitaciones de la tecnología. Finalmente, Farooq & Kunz, (2016) proporciona una comparación

entre ZigBee Pro y ZigBee IP, en términos de latencia, donde una red se compone de cinco nodos.

En referencia a 6LoWPAN, el trabajo de Kamma, Palla, Nelakuditi, & Yarrabothu, (2016) presenta una implementación sobre dispositivos MSP430 de Texas Instruments mediante una topología en estrella con un enrutador de borde y tres nodos, y se probaron las características de direccionamiento IP. Mientras que Pellerano, Falcitelli, Petracca, & Pagano, (2013) presenta una nueva arquitectura para aplicaciones de soporte en el campo de los sistemas inteligentes de transporte.

La implementación y evaluación de las diferentes políticas de administración de nodos vecinos aplicadas al protocolo de enrutamiento para redes de baja potencia y con pérdidas (*Routing Protocol for Low-power and Lossy Networks, RPL*) fue realizada por los autores Dawans, Duquennoy, & Bonaventure, (2012).

Otros trabajos proporcionan una comparación cualitativa, sin abordar ninguna evaluación cuantitativa del desempeño de los protocolos. Así mismo, otros autores presentan una evaluación comparativa del rendimiento de los protocolos ZigBee y 6LoWPAN para aplicaciones industriales. Desafortunadamente, no hay trabajos en la literatura que aborden la comparación del enfoque SDWSN y el enfoque distribuido representado por ZigBee y 6LoWPAN.

## **2.6. Protocolo ZigBee.**

En este trabajo, se considera analizar la versión ZigBee-Pro-2007 cuya pila de protocolos se muestra en la figura 2.2. Se considera el perfil de automatización del hogar y se implementa el enrutamiento de muchos a uno (*Many to One, MTO*), que se describe a continuación. El enrutamiento MTO permite establecer una topología de árbol, enraizada en el nodo coordinador. Para formar y mantener la topología de árbol, el nodo coordinador envía periódicamente un paquete de solicitud de ruta MTO (*MTO-Route Request, MTO-RR*) en difusión.

Cada nodo, que recibe un MTO-RR antes de retransmitirlo, lee el costo acumulado de la ruta (es decir, la suma de los costos de los enlaces de la ruta inversa hacia el nodo coordinador) incluido en el paquete y selecciona el siguiente salto hacia el nodo coordinador.

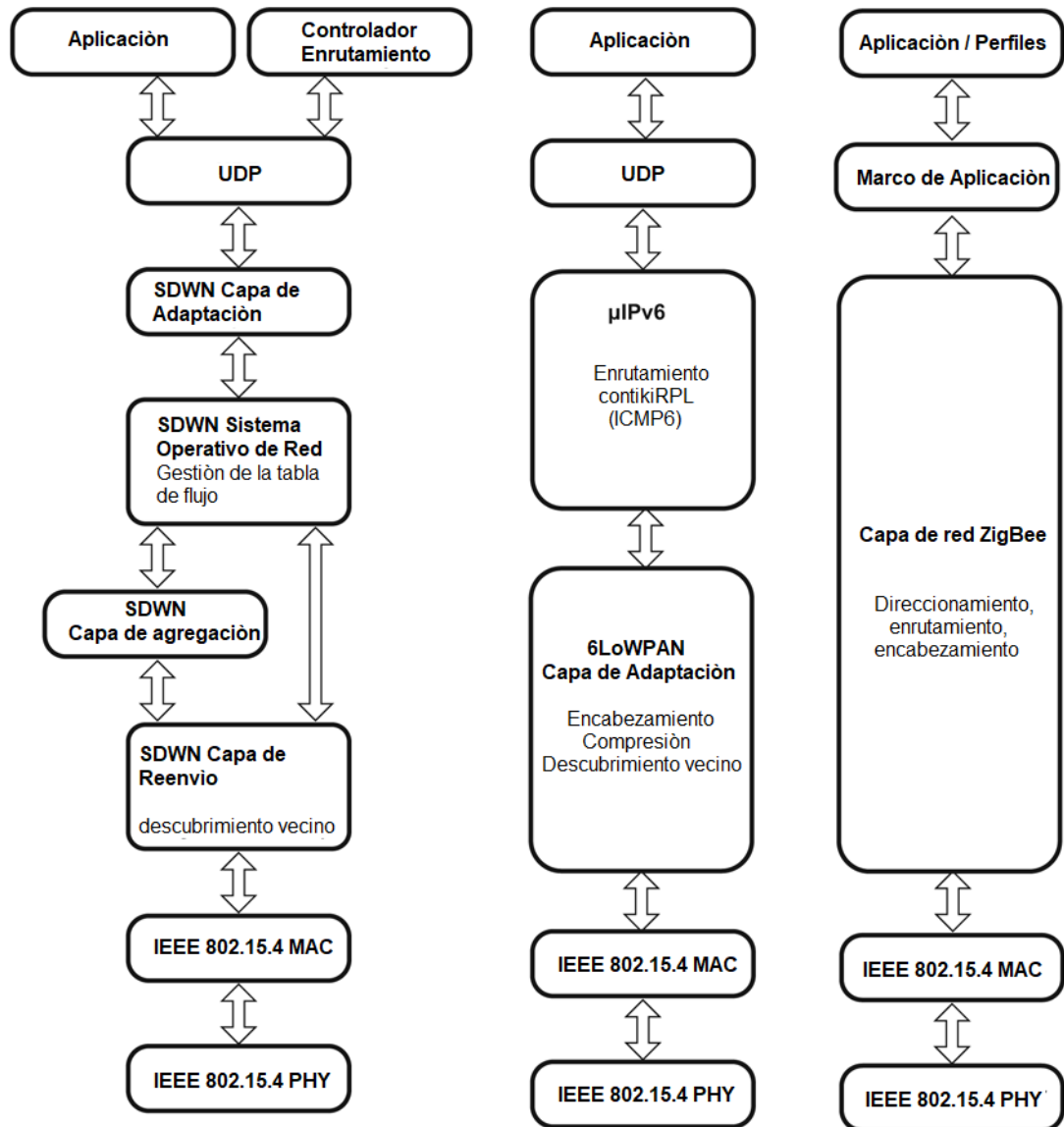


Figura 2. 2: Arquitecturas de protocolo (a) Izquierda: SDWN, (b) Centro: 6LoWPAN, y (c) Derecha: ZigBee.

Fuente: (Ma & Sun, 2013)

Elaborado por: Autor

En particular, si un nodo recibe varios MTO-RR de diferentes nodos, elige siguiente salto el nodo caracterizado por el costo total mínimo de ruta para el coordinador. Al final de esta transmisión MTO-RR, todos los nodos de la red conocen el siguiente salto que se utilizará para transmitir sus datos al

nodo coordinador, es decir, sus padres en la topología de árbol. Sin embargo, si el nodo coordinador quiere conocer la ruta para llegar a un nodo específico en la red (o un conjunto de nodos por multidifusión), el enrutamiento MTO debe combinarse con el enrutamiento de origen (*Source Routing, SR*).

Después de la transmisión MTO-RR, una vez que un nodo tiene un paquete de datos para enviar al nodo coordinador, primero envía un paquete de registro de ruta (*Route RECORD, RREC*) a través de la ruta seleccionada. Cada nodo en la ruta que recibe el paquete RREC, agrega en el campo de la lista de retransmisión su propia dirección y reenvía el nuevo paquete RREC hacia el nodo coordinador.

El nodo coordinador analiza el paquete RREC y almacena esa información en la tabla de ruta de origen. Cada vez, que el nodo coordinador tiene que enviar un paquete a un nodo, lee la lista de retransmisión de esta tabla y envía el paquete a través de la ruta seleccionada.

Para permitir que los nodos calculen los costos de enlace que se utilizarán en el enrutamiento MTO para la selección de la ruta, cada nodo en la red envía periódicamente paquetes de estado de enlace en difusión en un salto. Cada nodo que recibe el paquete de estado del enlace calcula el costo del enlace, en función del indicador de calidad del enlace del paquete recibido.

A pesar de que el nodo coordinador envía periódicamente los MTO-RR y no se generan a pedido (lo que haría que el protocolo sea proactivo), ZigBee guarda la característica reactiva mediante el uso del protocolo de vector de distancia a demanda Ad-hoc (*Ad-hoc On-demand Distance Vector, AODV*), cuando sea necesario. En particular, en caso de falla del enlace, AODV se utiliza para descubrir una nueva ruta hacia el destino.

Según AODV, un nodo que busca un nodo de destino envía un paquete de solicitud de ruta (*Route REQuest, RREQ*) en difusión, que es retransmitido por todos los nodos receptores hasta que llega al destino. Durante el proceso de retransmisión de RREQ, los nodos intermedios registran en sus tablas de

descubrimiento de rutas la dirección del remitente RREQ y el costo total correspondiente de la ruta inversa a la fuente.

La comparación entre los costos de las rutas de los paquetes relacionados con el mismo RREQ permite elegir la mejor ruta. Una vez que el destino recibe el RREQ, responde enviando una respuesta de ruta (RREP) en unidifusión de regreso a la fuente a lo largo del camino inverso.

En el caso de la transmisión de multidifusión, se debe establecer una ruta entre el coordinador y el grupo de multidifusión. En la parte experimental del capítulo 3, se utilizará el protocolo de enrutamiento AODV para establecer la ruta entre el nodo coordinador y el grupo de multidifusión; en este caso, el paquete RREQ, enviado en difusión, incluye la dirección del grupo de multidifusión a descubrir. Los nodos en la red que están vinculados al grupo de multidifusión objetivo envían un RREP al coordinador a través de la ruta seleccionada.

La última ruta se utiliza para la transmisión de paquetes de consulta. En la dirección del enlace ascendente, es decir, desde los nodos consultados hasta el coordinador, los nodos utilizan el mismo protocolo que para las transmisiones de unidifusión, por lo tanto, MTO.

## **2.7. Protocolo 6LoWPAN.**

El grupo de trabajo IETF 6LoWPAN publicó su primer documento en agosto de 2007 (Park et al., 2012). Entre las varias soluciones 6LoWPAN disponibles está la pila IPv6.

### **2.7.1. Pila de protocolos**

La pila de protocolos 6LoWPAN se muestra en la figura 2.3 donde las capas más bajas se basan en las capas física (*PHYSical, PHY*) y control de acceso al medio (*Medium Access Control, MAC*) dentro del estándar IEEE 802.15.4. Debido al hecho de que la integración directa entre capas de redes inferiores de IPv6 e IEEE 802.15.4 no es posible, el grupo de trabajo IETF 6LoWPAN ha especificado una capa de adaptación y un esquema de



compresión de encabezado para la transmisión de paquetes IPv6 a través de enlaces de radio IEEE 802.15.4.

El propósito de la capa de adaptación es proporcionar un mecanismo de fragmentación y reensamblado que permita a los paquetes IPv6 (la unidad de transmisión máxima para IPv6 sea 1280 bytes) se transmitan en tramas IEEE 802.15.4, que tienen un tamaño máximo de 127 bytes de la unidad de datos del protocolo de control de acceso al medio (*Mac Protocol Data Unit, MPDU*). En la capa de red, se utiliza el protocolo de enrutamiento IPv6 para redes de baja potencia y con pérdida (RPL).

En la capa de transporte, se aplica el protocolo de datagramas de usuario (*User Datagram Protocol, UDP*), que proporciona la mejor calidad de servicio de esfuerzo. Finalmente, en la capa de aplicación, el protocolo de aplicación restringida (CoAP) está presente.

### **2.7.2. Protocolo de enrutamiento.**

Según RPL, se construye un gráfico acíclico dirigido orientado al destino (DODAG), donde cada nodo puede tener más de un padre hacia la raíz. Uno de los nodos padres se llama padre preferido y se usa para enrutar hacia la raíz. Es decir, que el nodo coordinador actúa como la raíz. La topología se configura en función de una métrica de rango, que codifica la distancia de cada nodo con respecto a su raíz de referencia, según lo especificado por la función objetivo.

En particular, se va a utilizar la métrica de conteo de saltos como función objetivo; por lo tanto, el rango de un nodo dado representa el número de saltos que separan el nodo del coordinador. Las rutas en el DODAG se seleccionan para minimizar el rango. Los nodos RPL intercambian información de señalización para configurar y mantener el DODA. La construcción de DODAG es iniciada por la raíz que envía mensajes de objeto de información DODAG (DIO) a sus vecinos para anunciar un valor de rango mínimo. Al recibir un mensaje DIO, un nodo RPL:

1. actualizar la lista de sus nodos vecinos;
2. calcular su propio valor de rango;
3. seleccione su padre preferido usado como próximo salto para llegar a la raíz como el más fuerte (es decir, el que recibe el mayor poder); y
4. comenzar a transmitir mensajes DIO, que contienen su rango respectivo en el DODAG (una distancia a la raíz DODAG de acuerdo con el conteo de saltos).

Los nodos RPL también pueden enviar mensajes de solicitud de información (DIS) DODAG cuando se unen a la red para sondear a sus vecinos y solicitar mensajes DIO. Finalmente, los mensajes de objeto de anuncio de destino (DAO) se utilizan para propagar la información de destino hacia arriba a lo largo del DODAG. Los mensajes DAO son enviados en unidifusión por el nodo RPL al padre seleccionado para anunciar su dirección. Cuando un nodo recibe un DAO, actualiza su tabla de enrutamiento y luego la raíz DODAG utiliza esta información para construir rutas descendentes.

En nuestra implementación, cada enrutador de la ruta registra el identificador de ruta y el siguiente salto correspondiente hacia el destino. RPL utiliza un mecanismo de temporizador adaptativo, llamado temporizador de goteo, para controlar la velocidad de envío de mensajes DIO. El algoritmo de goteo implementa un modelo de verificación para verificar si los nodos RPL tienen información de enrutamiento desactualizada.

La frecuencia de los mensajes DIO depende de la estacionalidad de la red, y la frecuencia aumenta cuando se detecta la inconsistencia. Una vez que la red se estabiliza, el algoritmo de goteo reduce exponencialmente la velocidad a la que se emiten los mensajes DIO. RPL admite tráfico de unidifusión y multidifusión. En el caso de multidifusión, se va a utilizar el protocolo de reenvío RPL multidifusión sin estado (SMRF). Según este último, los nodos se unen a un grupo de multidifusión al anunciar su dirección en sus mensajes DAO salientes, que solo viajan ascendientemente en el DODAG.

Al recibir el mensaje de uno de sus nodos hijos, un enrutador ingresa una entrada en su tabla de reenvío para esta dirección de multidifusión. Esta entrada indica que un nodo en el DODAG es miembro del grupo. Este enrutador luego anunciará esta dirección en sus propios DAO y transmitirá datagramas de multidifusión destinados a esta dirección.

## Capítulo 3: Desarrollo de la propuesta.

### 3.1. Descripción general de SD-WSN.

Como ya se mencionó, recientemente han aparecido diferentes trabajos con el objetivo de extender los conceptos SDN a redes inalámbricas de sensores (WSN) y otras redes inalámbricas de área personal. Al presentar una nueva solución llamada redes definidas por software para redes inalámbricas de sensores (*Software Defined Networking for Wireless Sensor Networks, SD-WSN*), se irá más allá de los trabajos realizados únicamente con SDN de la siguiente manera. Se define una arquitectura completa que permita a los desarrolladores de software implementar controladores utilizando cualquier lenguaje de programación de su elección.

Además, SD-WSN presenta una capa de software que permite que varias redes virtuales se ejecuten en el mismo sensor inalámbrico físico o red W-PAN, de forma similar a lo que hace FlowVisor en las redes OpenFlow. SD-WSN define mecanismos simples para la definición y manipulación de la tabla de flujo (Flow Table) que hacen que SD-WSN tenga estado en comparación con OpenFlow tradicional que no tiene estado. De esta forma, los nodos WSN se pueden programar como máquinas de estados finitos, lo que puede ser útil para reducir la señalización entre nodos y controlador, lo que permitiría implementar políticas que no se pueden admitir de forma apátrida (sin estado).

El resto de este capítulo está organizado de la siguiente manera. En la sección 3.2 se ofrece una descripción general de SD-WSN. Los detalles de las principales características de la solución propuesta se explican en la sección 3.3, mientras que en la sección 3.4 se describe el prototipo SD-WSN que fue desarrollado. El rendimiento de SD-WSN se evalúa experimentalmente en la sección 3.5.

### 3.2. Descripción general de SD-WSN.

En esta sección, se realiza una descripción general de la solución SD-WSN. Más específicamente, primero se especifica brevemente los requisitos

que se deben satisfacer en el diseño de SD-WSN; luego se proporciona la descripción general del enfoque técnico de SD-WSN.

### **3.2.1. Requisitos para SD-WSN.**

Los requisitos para extender el paradigma SDN a WSN ya se han analizado en el capítulo 2. Dichos requisitos son la consecuencia obvia de las características de WSN que son significativamente diferentes de las de las redes cableadas. De hecho, las WSNs se caracterizan por bajas capacidades en términos de memoria, procesamiento y disponibilidad de energía. Además, las aplicaciones de WSN generalmente no son exigentes en términos de velocidad de datos. Por lo tanto, SD-WSN debe ser eficiente en el uso de los recursos del sensor, incluso si dicha eficiencia resultará en una tasa de datos más baja.

Para ser energéticamente eficiente, SD-WSN admite el ciclo de trabajo que es la posibilidad de apagar periódicamente la interfaz de radio de un nodo sensor y la agregación de datos. Estas características no se consideraran en los escenarios cableados del software OpenFlow.

Además, las interacciones entre los nodos sensores y los controladores deben reducirse tanto como sea posible para lograr la eficiencia del sistema. En este contexto, algún nivel de lógica de control programable en los nodos del sensor puede permitir tomar decisiones sin interactuar con el controlador cuando solo se necesita información local. Sin embargo, esto requiere la introducción de un estado, mientras que la instancia estándar de OpenFlow de SDN no tiene estado.

Dado que las WSNs están intrínsecamente centradas en los datos, se han propuesto varias soluciones que hacen que los protocolos de red conozcan el contenido del paquete. En consecuencia, los nodos SD-WSN pueden manejar paquetes basados en el contenido almacenado en su encabezado y carga útil. Además, en OpenFlow, los paquetes se clasifican en función de la igualdad entre un determinado campo en el encabezado del paquete y una cadena de bytes determinada; diferente a eso, en SD-WSN

dicha clasificación se puede hacer en función de otros operadores relacionales más complejos, por ejemplo, mayor que, menor que, diferente de, etc.

Finalmente, la naturaleza centrada en los datos de WSN implica otra diferencia significativa entre el comportamiento esperado de SD-WSN y OpenFlow. De hecho, en OpenFlow los recursos de red se dividen por FlowVisor en segmentos, cada uno asignado a un controlador, y un paquete puede pertenecer a un solo segmento.

En cambio, en las WSNs, el mismo dato puede ser de interés para varias aplicaciones que utilizan diferentes controladores. Por lo tanto, en SD-WSN, un paquete no está necesariamente vinculado con un controlador, es decir, diferentes controladores pueden especificar diferentes reglas para el mismo paquete.

### **3.2.2. Enfoque de SD-WSN.**

El comportamiento de los nodos sensores para SD-WSN está completamente codificado en tres estructuras de datos, a saber: la matriz de estados WSN, la matriz de IDs aceptados y la tabla de flujo WSN. Como en la mayoría de los enfoques de SDN, tales estructuras se llenan con la información proveniente de los controladores, que se ejecuta en los servidores apropiados. De esta manera, los controladores definen las políticas de red que implementarán los nodos sensores.

En cualquier momento, los nodos SD-WSN se caracterizan por un estado actual para cada controlador activo. Un estado es una cadena de bits  $s_{State}$ . La matriz de estados WSN es la estructura de datos que contiene los valores de los estados actuales.

Dada la naturaleza de transmisión del medio inalámbrico, los nodos sensores también recibirán paquetes que no están destinados a ellos (ni siquiera para el reenvío). La matriz de IDs aceptados permite que cada nodo sensor seleccione solo los paquetes que debe procesar más. De hecho, el

encabezado de los paquetes contiene un campo en el que se especifica una ID aceptada.

Un nodo, al recibir un paquete, controla si la ID contenida en dicho campo aparece en su matriz de ID aceptadas. Si este es el caso, el nodo procesará aún más el paquete; de lo contrario lo dejará caer. En el caso de que el paquete deba procesarse, el nodo sensor examinará las entradas de su tabla de flujo en WSN. Cada entrada de la tabla de flujo en WSN contiene una sección de reglas coincidentes que especifica las condiciones bajo las cuales se aplica la entrada. En SD-WSN, las reglas de coincidencia pueden considerar cualquier parte del paquete actual, así como cualquier parte del estado actual.

Si se cumplen las reglas de coincidencia, el nodo sensor realizará una acción especificada en la sección restante de la entrada de la tabla de flujo en WSN. Hay que tener en cuenta que dicha acción puede referirse a cómo manejar el paquete, así como a modificar el estado actual. Si no se incluye ninguna entrada en la tabla de flujo en WSN cuyas reglas coincidentes se apliquen al paquete/estado actual, se envía una solicitud a los controladores.

Para ponerse en contacto con los controladores, un nodo sensor debe tener una entrada de tabla de flujo en WSN que indique su mejor salto siguiente hacia uno de los sumideros (sink). Esta entrada es diferente de las demás porque no está configurada por un controlador, sino que es descubierta por cada nodo de forma distribuida.

Para este fin, la capa de descubrimiento de topología (*Topology Discovery, TD*) ejecuta un protocolo apropiado como se describirá más adelante, que se basa en el intercambio y procesamiento de los paquetes apropiados llamados paquetes TD. Dichos paquetes contienen información sobre el nivel de la batería y la distancia desde el sumidero (más cercano) en términos de número de saltos. Cada vez que un nodo recibe uno de esos paquetes, compara el mejor salto siguiente actual con la información que acaba de adquirir, luego elige el mejor salto siguiente dando prioridad al

número de saltos, luego el valor RSSI recibido con el mensaje y finalmente el nivel de batería residual. Esta información también se utiliza para completar una lista de nodos vecinos de WSN.

Esta lista contiene las direcciones de los nodos vecinos, sus RSSI y sus niveles de batería. Esta tabla se envía periódicamente a la capa de gestión de topología (*Topology Management, TM*), tal como se detalla a continuación para construir una representación gráfica de la red. Después de eso, la tabla se borra totalmente y se reconstruye con los paquetes TD entrantes para tener siempre una vista actualizada de la topología local.

Uno de los controladores actúa como un proxy entre la red física y los otros controladores, esto se llama WSN-Visor y es el análogo de Flow Visor en las redes tradicionales de OpenFlow. Los controladores especifican las políticas de administración de red que debe implementar las WSNs y pueden depender de la aplicación. En consecuencia, los controladores pueden interactuar con la aplicación.

Tener en cuenta que los nodos sensores tienen capacidades limitadas en términos de memoria, por lo tanto, la selección del tamaño de las diferentes estructuras de datos es muy importante. La elección óptima de dicho tamaño depende de varias características específicas de implementación establecidas por WSN -Visor durante la fase de inicialización.

### **3.2.3. Arquitectura de protocolo SD-WSN.**

En las redes SD-WSN, se pueden distinguir los nodos sensores y un (o varios) sumidero. Los sumideros son las puertas de enlace entre los nodos sensores que ejecutan el plano de datos y los elementos que implementan el plano de control. La pila de protocolos del plano de datos, ejecutada principalmente por nodos de sensores, se muestra en el lado izquierdo de la figura 3.1.

La pila de protocolos del sumidero (Sink) y los otros elementos que implementan el plano de control se describen en el lado derecho de la figura



4.1. Los nodos de sensor incluyen un transceptor IEEE 802.15.4 y una unidad de microcontrol (MCU) comúnmente llamado microcontrolador. Por encima de la pila de protocolos IEEE 802.15.4, la capa de reenvío se ejecuta en el MCU que maneja los paquetes entrantes como se especifica en una tabla de flujo de WSN. Esta tabla se actualiza continuamente por la capa de reenvío de acuerdo con los comandos de configuración enviados por los controladores.

La capa de procesamiento de paquetes dentro de la red (*In-Network Packet Processing, INPP*) se ejecuta sobre la capa de reenvío. Este es responsable de operaciones como la agregación de datos u otro procesamiento dentro de la red. En la implementación actual de SD-WSN, la capa INPP concatena pequeños paquetes que deben enviarse por rutas similares. Esto reduciría la sobrecarga de la red. Además, ya existen soluciones que permiten al INPP realizar una codificación de red que es muy eficiente en varios escenarios de WSN.

La capa de descubrimiento de topología (TD), en cambio, puede acceder a todas las capas de la pila de protocolos mediante las API apropiadas. Por lo tanto, puede recopilar información local de los nodos y controlar su comportamiento en todas las capas, de acuerdo con las indicaciones proporcionadas por los controladores. La capa TD también proporciona una API a la capa de aplicación, que extiende las API IEEE 802. Esto garantiza el legado con el plano de control, las lógicas de administración de red están dictadas por uno o varios controladores, uno de los cuales es el WSN -Visor.

WSN -Visor incluye una capa de gestión de topología (TM) que abstrae los recursos de la red para que diferentes redes lógicas, con diferentes políticas de gestión establecidas por diferentes controladores, puedan ejecutarse en el mismo conjunto de dispositivos físicos. La capa de TM tiene acceso a las API que ofrecen todas las capas de protocolo. Dichas API permiten controlar el comportamiento de todas las capas de protocolo y, por lo tanto, implementar operaciones entre capas.

El uso de la capa TM está impulsado por dos requisitos: (a) recolectar información local de los nodos y enviarla a los controladores en forma de un gráfico de la red (información de informe relacionada con la topología, nivel de energía residual, SNR en los enlaces, etc.), y (b) controlar todas las capas de la pila de protocolos según lo especificado por los controladores.

Para este propósito, entre el dispositivo de sumidero (caracterizado por la misma pila de protocolos de nodos sensores) y el WSN -Visor existe la capa de adaptación que es responsable de formatear los mensajes recibidos del sumidero de tal manera que puedan ser manejados por WSN -Visor y viceversa.

Los controladores pueden ejecutarse en el mismo nodo que aloja la capa TM o en servidores remotos. Como consecuencia, las interacciones entre los Controladores y la capa TM pueden ocurrir de varias maneras, tal como se muestra en la parte central de la figura 3.1. De hecho, en el caso de que los controladores se ejecuten en el mismo nodo sensor que aloja la capa TM, las interacciones ocurrirán a través de los métodos Java ofrecidos por la capa de TM.

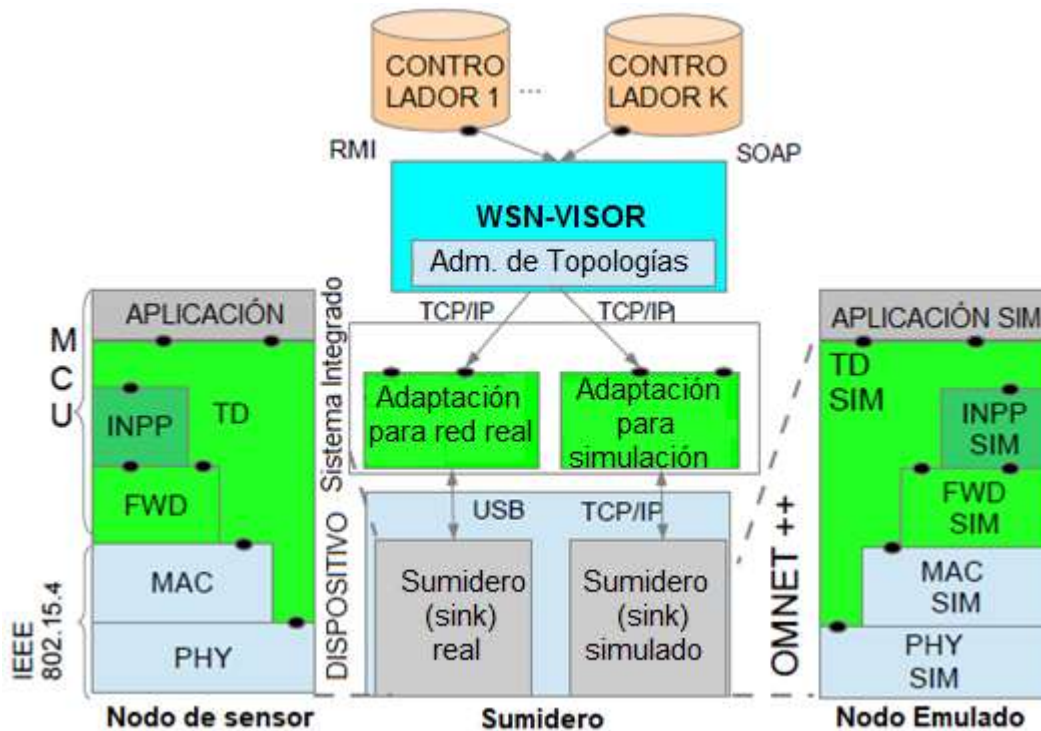


Figura 3. 1: Pila de protocolos SD-WSN.

Elaborado por: Autor

Alternativamente, las interacciones pueden ocurrir a través de la invocación de método remoto (*Remote Method Invocation, RMI*) de Java o el protocolo simple de acceso a objetos (*Simple Object Access Protocol, SOAP*). De esta manera, los programadores pueden implementar controladores en Java o en algunos lenguajes de programación web.

Finalmente, tenga en cuenta que la pila de protocolos SD-WSN también incluye una capa de adaptación específica que puede interactuar con un sumidero simulado (no un sumidero real). De esta manera, el plano de control puede establecer las políticas de red de una red simulada. En otras palabras, SD-WSN ofrece una herramienta muy similar al programa de simulación Mininet.

### **3.3. Detalles del protocolo SD-WSN.**

En esta sección se describe en detalle las principales características de los protocolos SD-WSN. Más específicamente, en la sección 3.3.1 se explica el protocolo de descubrimiento de topología (TD). Luego, en la sección 3.3.2 se describe en detalle cómo se comportan los nodos sensores cuando reciben un nuevo paquete.

#### **3.3.1. Descubrimiento de topología (TD)**

En la sección 3.2.3 se ha explicado que el módulo de gestión de topología en WSN-Visor crea una vista coherente del estado actual de la red. Para este propósito, fue necesario recopilar información de topología local generada por los nodos sensores. El protocolo de descubrimiento de topología (TD) ejecutado por todos los nodos sensores es responsable de generar dicha información y entregarla al WSN-Visor.

El protocolo TD mantiene actualizada la información sobre el próximo salto hacia los controladores y sus nodos vecinos actuales. Con este fin, todos los sumideros (ya se indicó que puede existir varios sumideros en la misma red SD-WSN.) en la red SD-WSN transmiten periódicamente y (casi) simultáneamente un paquete de descubrimiento de topología (TD packet) a través del canal inalámbrico de difusión. Dicho paquete contiene la identidad

del sumidero que lo ha generado, un nivel de batería y la distancia actual del sumidero que inicialmente se establece en 0.

Un nodo sensor A que recibe un paquete TD del nodo sensor B (tenga en cuenta que B puede ser un sumidero o Sink) realiza las siguientes operaciones:

1. Inserta el nodo B en la lista de sus nodos vecinos actuales junto con el RSSI actual y el nivel de batería. Obviamente, si el nodo B ya está presente en la lista de vecinos actuales, solo se actualizan los valores de nivel de batería y RSSI;
2. Controla si se ha recibido recientemente un paquete TD con un valor más bajo de la distancia actual desde el sumidero (Sink). Si este no es el caso, el nodo A actualiza el valor informado en el paquete TD al valor actual más uno y establece su próximo salto hacia los controladores igual al nodo B;
3. Establece su nivel de batería en el campo correspondiente del paquete TD;
4. Transmite el paquete TD actualizado a través del canal inalámbrico de transmisión.

Periódicamente, cada nodo de sensor genera un paquete que contiene su lista actual de nodos vecinos y lo envía al WSN-Visor. Hay que tener en cuenta que la lista de nodos vecinos se borra periódicamente. Los nodos que reciben paquetes dirigidos hacia el WSN-Visor o los controladores los transmiten al conjunto de nodos como su próximo salto hacia los controladores.

La tasa de generación de paquetes de TD, así como de los paquetes que contienen información de topología local afecta el rendimiento de SD-WSN. De hecho, cuanto mayor es la frecuencia, mayor es la sobrecarga generada por el protocolo. Sin embargo, tales frecuencias no pueden ser demasiado bajas en escenarios dinámicos (con cambios rápidos de topología); en consecuencia, su configuración es específica de la aplicación y puede ser controlada por WSN-Visor.

### 3.3.2. Manejo de paquetes.

En esta sección, se describe cómo funciona el protocolo de reenvío descrito en la sección 3.2.3 al recibir un paquete. Para este propósito, primero se proporciona una descripción del formato de paquete WSN; luego, una descripción de la estructura de la tabla de flujo WSN y, finalmente, se explica cómo se utiliza la tabla de flujo WSN al recibir un paquete.

Como se muestra en la figura 3.2, los paquetes SD-WSN tienen un encabezado fijo que consta de 10 bytes divididos en los siguientes campos:

- El campo longitud del paquete (packet length) proporciona la longitud del paquete, incluida la carga útil (si la hay), en bytes.
- El alcance (scope) identifica un grupo de controladores que han expresado interés en el contenido del paquete. El valor de alcance se establece inicialmente en 0 (por defecto) pero se puede modificar a través de las entradas apropiadas en la tabla de flujo WSN del nodo sensor que genera el paquete. En esta implementación, los valores de alcance tienen validez global, ya que WSN-Visor garantiza la coherencia de toda la red.
- Las direcciones de origen (source address) y destino (destination address) obviamente especifican las direcciones (se utiliza direcciones de dos bytes en la implementación) del nodo que ha generado el paquete y el destino previsto.
- La bandera (flag) U se usa para marcar paquetes que deben entregarse al sumidero más cercano.
- El campo tipo de paquete se usa para distinguir entre diferentes tipos de mensajes, de hecho, además de paquetes de datos, paquetes de TD y los paquetes que contienen información de topología local, que ya se ha discutido, SD-WSN utiliza otros tipos de paquetes para solicitar una nueva entrada a los controladores, para la introducción de una nueva entrada en la tabla de flujo WSN de un nodo sensor determinado, para abrir una ruta en una secuencia de nodos sensores y para desactivar la interfaz inalámbrica de un nodo sensor durante

un cierto intervalo de tiempo. El tipo de paquete determinará la interpretación de la carga útil del paquete.

- El TTL es el tiempo de vida y se reduce en uno en cada salto.
- Finalmente, la ID del siguiente salto es el campo que debe estar presente en la matriz de ID aceptadas para que el paquete sea procesado por el nodo del sensor (como se explicó en la sección 3.2.2).

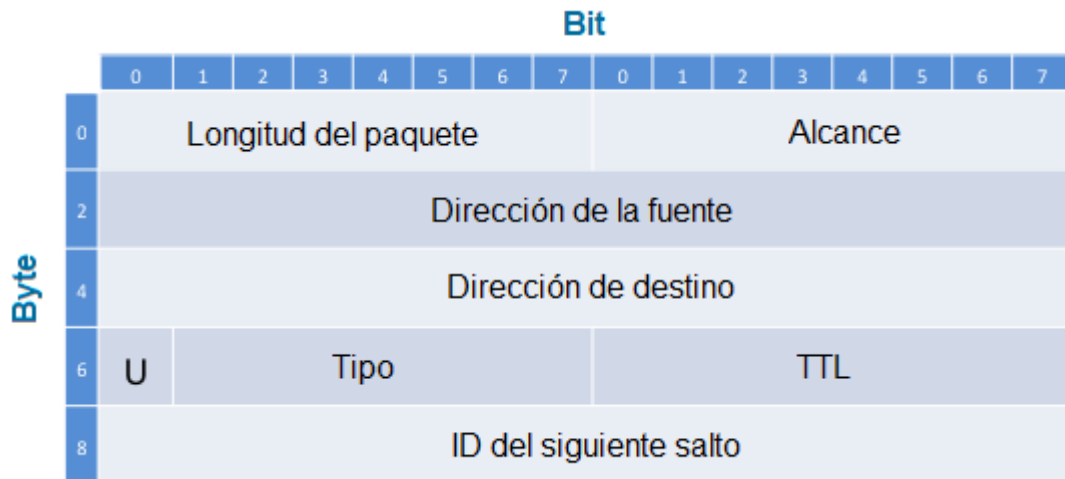


Figura 3. 2: Encabezado de paquete WSN.  
Elaborado por: Autor

En la figura 3.3 se muestra la estructura de la tabla de flujo WSN. Como en el caso de OpenFlow, se pueden distinguir tres secciones: (i) reglas de coincidencia (Matching Rule), (ii) acciones (Action) y (iii) estadísticas (Statistics). Las reglas de coincidencia especifican hasta tres condiciones. Si se cumplen dichas condiciones, se ejecuta la acción correspondiente y se actualiza la información presentada en la sección estadística.

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Op.	Size	S	Offset	Value	Op.	Size	S	Offset	Value	Op.	Size	S	Offset	Value	Type	M	S	Offset	Value	TTL	Counter
=	2	0	2	8	>	2	0	10	$x_{th}$	=	1	1	0	0	Modify	1	1	0	1	122	23
*	2	0	2	8	≤	2	0	10	$x_{th}$	=	1	1	0	1	Modify	1	1	0	0	122	120
=	2	0	2	8	-	0	-	-	-	-	0	-	-	-	Forward	0	0	0	0	122	143
*	2	0	2	A	=	1	1	0	0	-	0	-	-	-	Drop	0	0	-	-	100	42
*	2	0	2	A	=	1	1	0	1	-	0	-	-	-	Forward	0	0	0	0	100	32

Figura 3. 3: Tabla de flujo de WSN.  
Elaborado por: Autor

Cada regla de coincidencia consta de un campo (S) que especifica si la condición se refiere al paquete actual ( $S = 0$ ) o al estado ( $S = 1$ ); los campos Offset y Size especifican el primer byte y el tamaño, respectivamente, de la cadena de bytes en el paquete o el estado que debe considerarse, el campo operador le da al operador relacional que se verificará con el valor dado en la regla.

Por ejemplo, la segunda regla de coincidencia de la primera entrada en la tabla de flujo WSN dada en la figura 3.3 se cumple si se asumen los primeros 2 bytes ( $Size = 2$ ) después del byte 10 ( $Offset = 10$ ) del paquete actual ( $S = 0$ ) un valor que es más alto ( $Op = ">"$ ) que  $x_{Thr}$  ( $Valor = x_{Thr}$ ).

Si se cumplen todas las condiciones especificadas en la sección reglas de coincidencia (si  $Size=0$ , entonces no se considera la regla de coincidencia), se ejecuta la acción correspondiente. Una acción se especifica mediante cinco campos. El tipo (Type) especifica el tipo de acción. Los valores posibles del campo tipo (Type) pueden ser "Reenviar a" (Forward to), "Descartar" (Drop), "Modificar" (Modify), "Enviar a INPP" (Send to INPP), "Apagar la radio" (Turn off radio).

La bandera M especifica si la entrada es exclusiva ( $M = 0$ ) o no ( $M = 1$ ). En el primer caso ( $M = 0$ ), si se satisfacen las condiciones, el nodo sensor ejecuta la acción y luego deja de examinar la tabla de flujo WSN. En el segundo caso ( $M = 1$ ), en cambio, después de ejecutar la acción, el nodo sensor continúa para navegar a la tabla de flujo WSN y ejecutar otras acciones si se cumplen las condiciones correspondientes especificadas en la sección de reglas de coincidencias.

El significado de los otros dos campos (es decir, Offset y Value) depende del tipo de acción. Por ejemplo, si la acción es "Forward to", deben especificar cuál es la ID del próximo salto (que se escribirá en el paquete), si es "Drop", proporcionan la probabilidad de caída, así como la ID del próximo salto en caso de que el paquete no se descarta, si es "Modify", especifican el desplazamiento y el nuevo Value a escribir, si es "Send to INPP", especifican

el tipo de procesamiento que debe ejecutarse, si es "Turn off radio" especifican después de cuánto tiempo se debe volver a encender la radio.

En caso de que la acción sea "Modify", el indicador S especifica si la acción debe ejecutarse en el paquete o en el estado. Las estadísticas son utilizadas como en OpenFlow estándar y, por lo tanto, no son discutidas más en este documento.

A continuación, se muestra cómo los nodos sensores utilizan sus estructuras de datos en un escenario ejemplar destacando las características específicas de SD-WSN. Considere la topología de red que se muestra en la figura 3.4 y suponga que los datos medidos por el sensor A son significativos solo si los datos medidos por el sensor B son más altos que un umbral dado  $x_{Thr}$ .

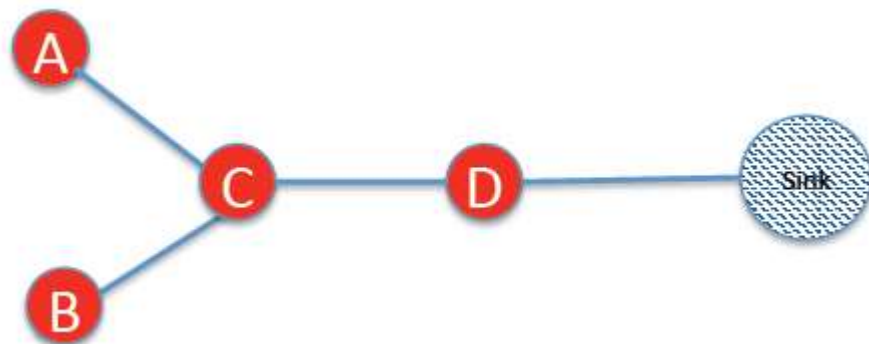


Figura 3. 4: Topología ejemplar.  
Elaborado por: Autor

Por lo tanto, si se busca la eficiencia energética, se debe implementar una política de red que obligue al nodo C a descartar paquetes si el paquete recibido por B contiene datos medidos inferiores a  $x_{Thr}$ . Al usar soluciones tradicionales similares a OpenFlow, es imposible aplicar el comportamiento anterior por las siguientes razones:

- La coincidencia se ejecuta solo para verificar la equivalencia entre un campo en el encabezado del paquete y un valor específico, es decir, no es posible observar la carga útil y no se admiten relaciones de tipo "superior a";



- En soluciones sin estado es imposible hacer que el manejo del paquete dependa del contenido de otro paquete.

En cambio, en SD-WSN, la política anterior se puede realizar fácilmente a través de la máquina de estados finitos que se presenta en la figura 3.5 que se puede implementar a través de las cinco entradas de la tabla de flujo WSN que se mostró en la figura 3.3. De hecho, las dos primeras líneas especifican las transiciones entre los estados 0 y 1 y viceversa, dependiendo del valor contenido en el décimo byte de los paquetes generados por el nodo B.

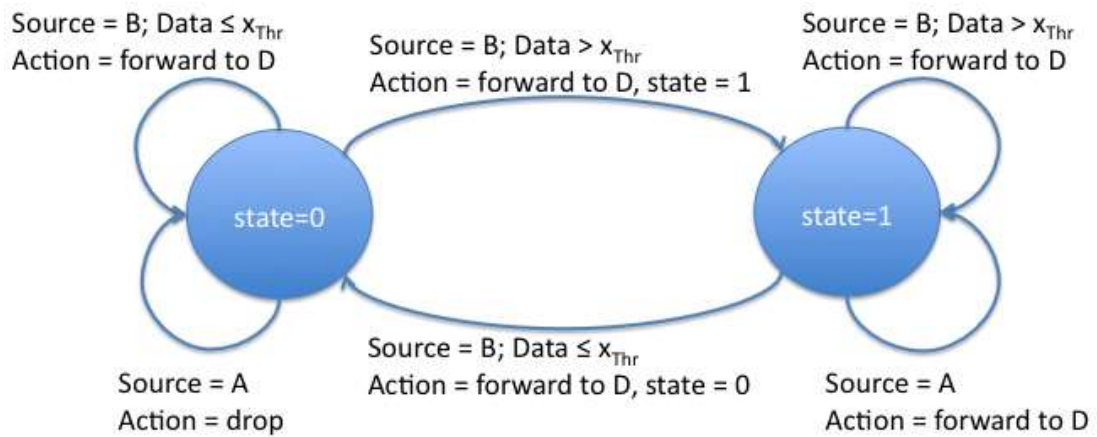


Figura 3. 5: Máquina de estado finito que implementa una política tal que los paquetes generados por A se descartan si los últimos datos medidos por B son inferiores (o iguales) a  $x_{Thr}$ .

Elaborado por: Autor

Más específicamente, tener en cuenta que, en la primera entrada, la primera regla de coincidencia selecciona los paquetes que provienen de B, la segunda regla de coincidencia selecciona los que tienen en el décimo y undécimo bytes un valor superior a  $x_{Thr}$ , finalmente la tercera regla de coincidencia selecciona los casos en los que el estado actual del nodo es 0. Si se cumplen todas las reglas anteriores, el estado se establece en 1 como se muestra en la sección acción. Análogamente, la segunda entrada selecciona los casos en que B ha generado el paquete entrante contienen datos medidos inferiores o iguales a  $x_{Thr}$ , y el estado actual es uno; y en tales casos establece el estado en 0. La tercera entrada en la tabla se ejecuta cada vez que se recibe un paquete generado por B y especifica que el paquete debe reenviarse a D, en cualquier caso. Finalmente, la cuarta y quinta entrada

especifican que los paquetes que provienen de A deben descartarse si el estado actual es 0 (ver la cuarta entrada) o reenviarse a D si el estado actual es 1 (ver la quinta entrada).

### **3.4. Prototipo y banco de pruebas**

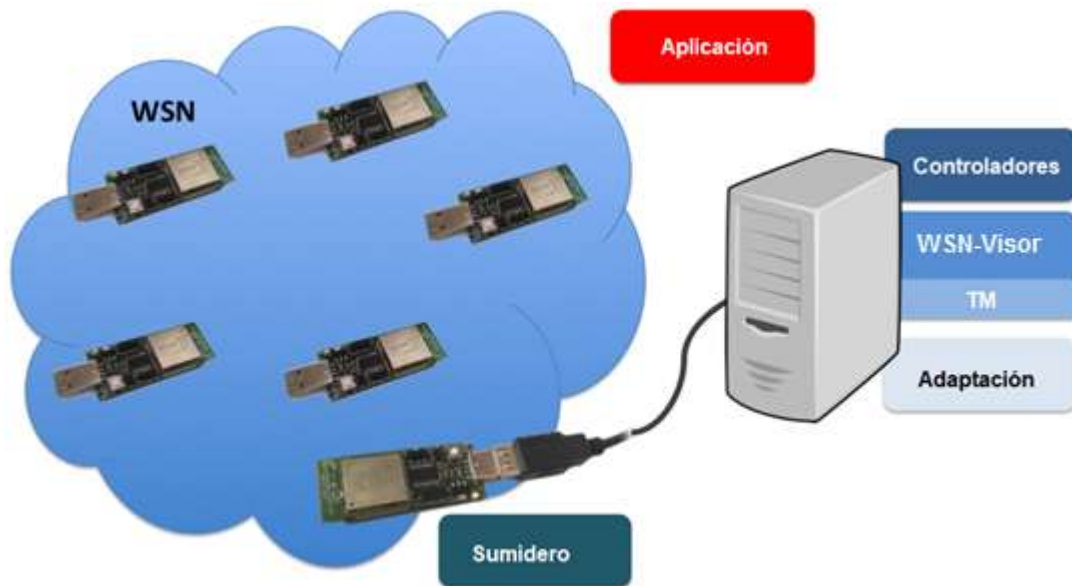
Considerando el despliegue de SD-WSN, es importante notar que cada nodo está equipado con 8 kB de RAM y 256 kB de memoria Flash, de los cuales 40 kB se utilizan para la capa MAC (TIMAC para CC2530 v1.4.0) y 10 kB se utilizan para el protocolo SD-WSN. En lo que concierne al plano de control, el prototipo diseñado admite diferentes opciones de implementación. El más simple es el que se muestra en la figura 3.6a, en el que el nodo que aloja el sumidero (Sink) se conecta a la computadora de escritorio o laptop mediante USB 2.0.

El banco de pruebas realizado, WSN-Visor y los controladores están alojados en la computadora de escritorio o laptop que está equipada con Intel (R) Core (TM) 5 CPU 2.40 GHz y 8 GB de RAM con Windows 10, de 64 bits. Los controladores se han implementado utilizando Java 7. La información de topología se almacena en un objeto gráfico de JGraphT.

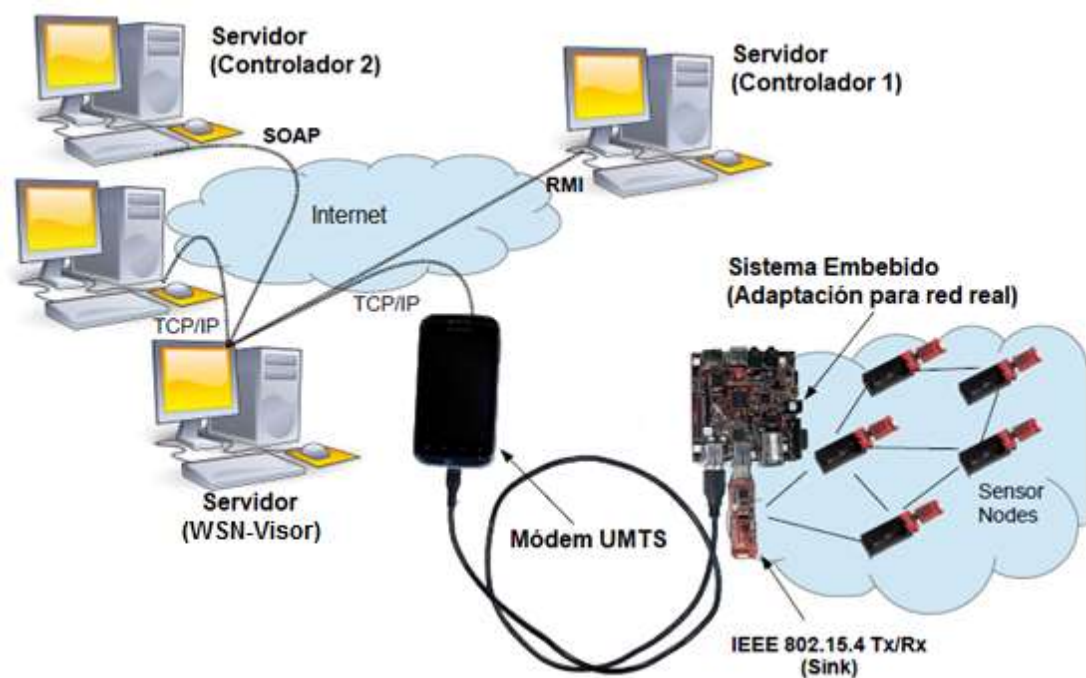
La opción de implementación anterior requiere la presencia de un nodo (la PC) con recursos computacionales significativos en el área donde se implementan los nodos sensores. Sin embargo, en varios escenarios, no es posible implementar nodos tan potentes en el área de la red. En estos casos, el sumidero (Sink) generalmente está conectado a un sistema integrado que accede a Internet a través de alguna interfaz de comunicación. Por ejemplo, en el banco de pruebas experimental (véase la figura 3.6b), el sumidero (Sink) es un dispositivo TI CC2500 conectado por USB a un Beagleboard que ejecuta un sistema operativo Linux (Ubuntu 12.46).

La capa de adaptación se implementa en el Beagleboard que envía paquetes de control al WSN-Visor en un servidor remoto. En el banco de pruebas, el Beagleboard está equipado con una interfaz UMTS (el teléfono

inteligente en la figura 3.6b) y la comunicación entre Adaptación y WSN-Visor se realiza a través de conexiones TCP / IP.



(a) La opción de implementación más simple.



(b) Opción de implementación distribuida.

Figura 3. 6: Opciones de implementación de SD-WSN.

Elaborado por: Autor

Los controladores pueden ser alojados por otras PC (o máquinas virtuales) e interactuar con la capa WSN-Visor usando SOAP, REST, RMI o UDP. Finalmente, las simulaciones que modelan el comportamiento de los nodos sensores y los sumideros se pueden ejecutar en otra PC.

En la figura 3.7 se muestra una captura de pantalla de la topología de la red de sensores simulada. El nodo 0 es el sumidero e interactúa a través del módulo de adaptación con una instancia real de WSN-Visor. En consecuencia, los controladores 1 y 2 pueden ser controladores reales que determinan las políticas que aplican los nodos sensores simulados. Además, el Sink (emulado) se puede usar para crear una extensión de red virtual para que los nodos simulados y reales estén completamente integrados y puedan interactuar con los demás. Esto puede ser útil para probar un escenario de red real en el que no hay suficientes dispositivos reales. En este caso, solo se utiliza un controlador para ambos nodos (real y simulado) y los trata a todos sin hacer ninguna distinción.

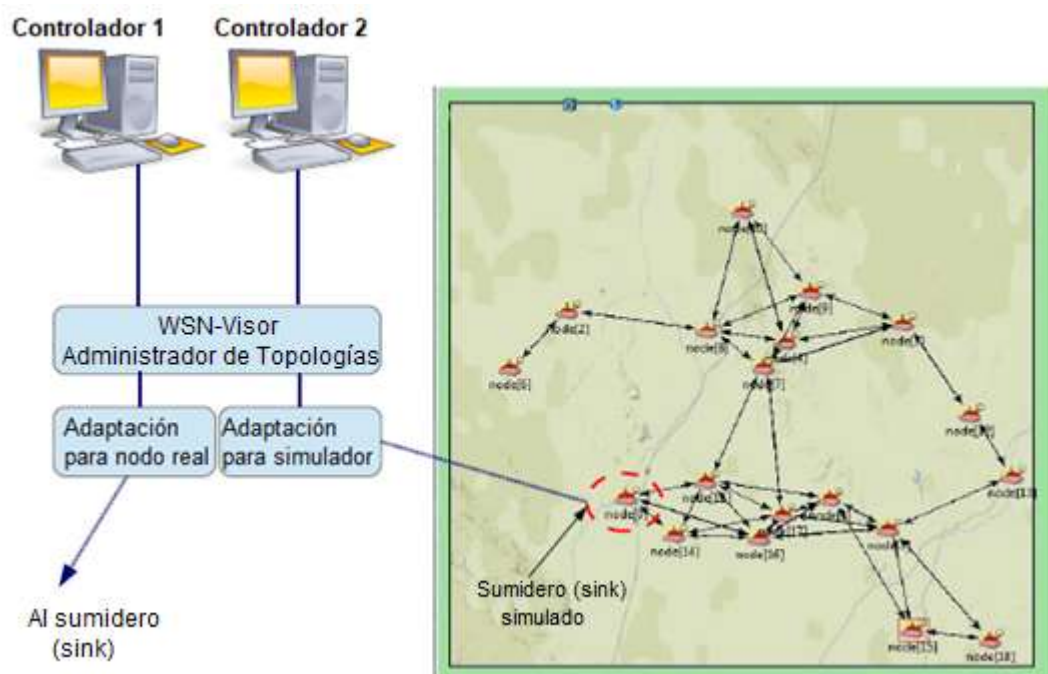


Figura 3. 7: Integración con el simulador OMNeT ++.  
Elaborado por: Autor

### 3.5. Evaluación del desempeño de la SD-WSN.

En esta sección se ilustran los resultados obtenidos por la plataforma SD-WSN en un banco de pruebas físico. Más específicamente, se han implementado 6 nodos (5 nodos sensores y un sumidero) tal como se muestra en la figura 3.8. En las pruebas experimentales, el sumidero se conectó a través de USB a una PC que ejecuta la capa de adaptación y toda la funcionalidad del plano de control (véase la figura 3.6a). Finalmente, el controlador fue implementado en Java y ejecuta el algoritmo Dijkstra.

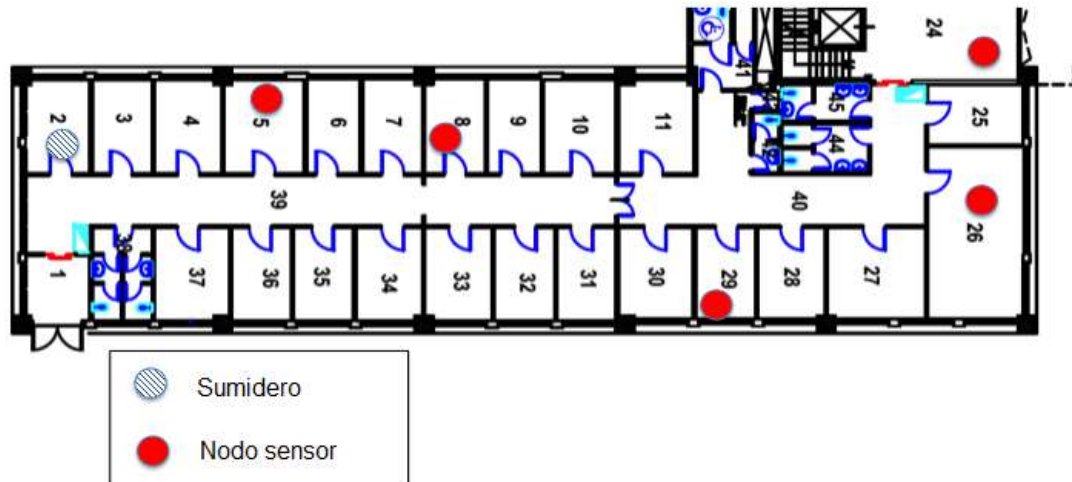


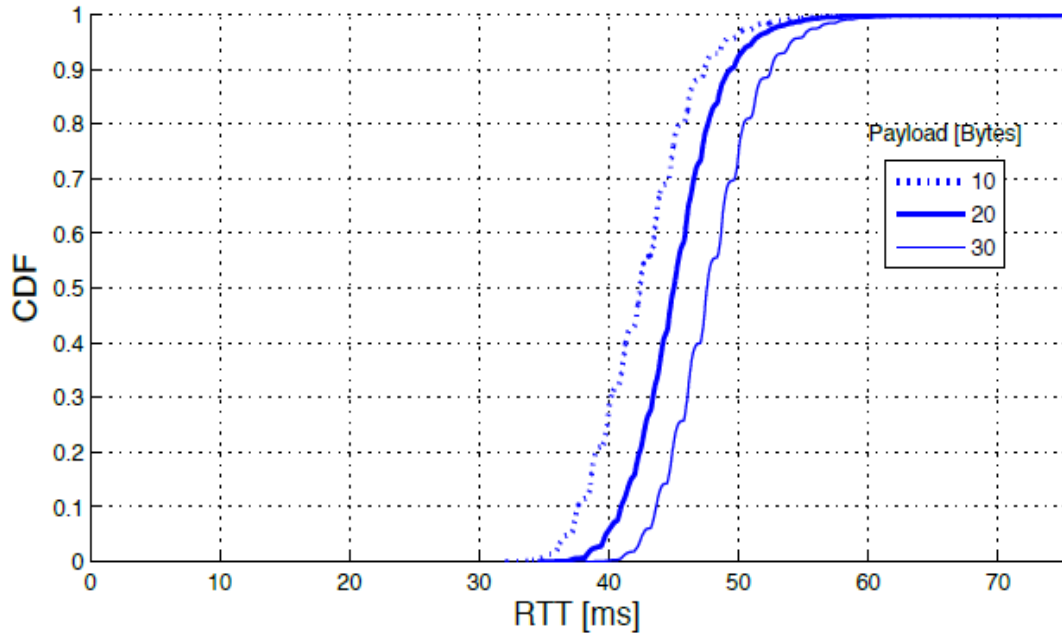
Figura 3. 8: Despliegue de nodos.  
Elaborado por: Autor

En cada operación de medición, se han enviado 5000 paquetes de datos, cada uno cada 15 segundos. Se han considerado diferentes tamaños de carga útil para dichos paquetes (10, 20 y 30 bytes). Además, se ha cambiado el intervalo de tiempo,  $T$ , entre dos generaciones consecutivas de paquetes de TD. En cada operación se ha establecido el intervalo de tiempo entre las transmisiones de información de topología local al doble del valor de  $T$ . A continuación, se describe el rendimiento alcanzado por SD-WSN en términos de:

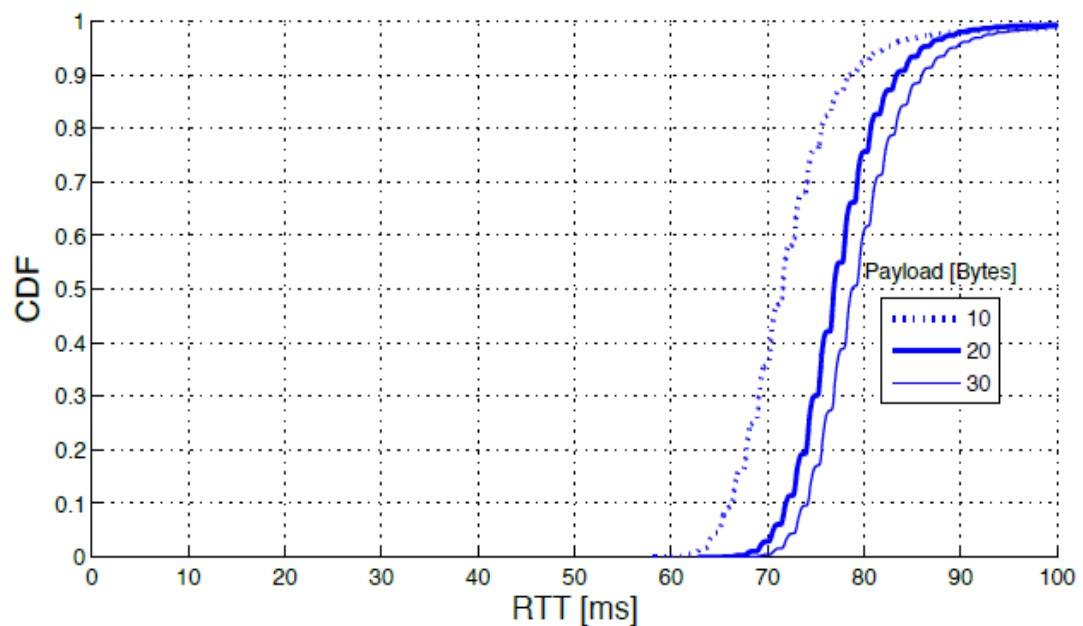
- Tiempo de ida y vuelta (*Round Trip Time, RTT*), es decir, el intervalo de tiempo entre la generación de paquetes de datos y la recepción del acuse de recibo correspondiente;
- Eficiencia, medida como la relación entre el número de bytes de carga útil recibidos por los destinos previstos y el número total de bytes que circulan en la red;
- Tiempo de respuesta del controlador, medido como la duración del intervalo de tiempo cuando el controlador recibe una solicitud de una nueva entrada y el instante de tiempo cuando el controlador envía la entrada correspondiente.

En las figuras 3.9a y 3.9b se representan las funciones de distribución acumulativa (*Cumulative Distribution Functions, CDF*) del RTT cuando la distancia entre el origen del paquete y el destino del paquete es igual a 3 y 5,

respectivamente. En cada figura se presentan tres curvas obtenidas para diferentes valores del tamaño de la carga útil (10, 20 y 30 bytes). Como se esperaba, RTT se incrementa a medida que aumenta la distancia y la carga útil.



(a) Número de saltos = 3.



(b) Número de saltos = 5

Figura 3. 9: CDF del RTT para diferentes tamaños de carga útil y diferentes distancias entre el nodo de origen y el de destino.

Elaborado por: Autor

Además, se espera un comportamiento similar de la desviación estándar. De hecho, esto se refleja en las figuras 3.10 y 3.11, donde se muestra el

promedio y la desviación estándar del RTT frente al tamaño de la carga útil para diferentes valores de la distancia entre el origen y el destino.

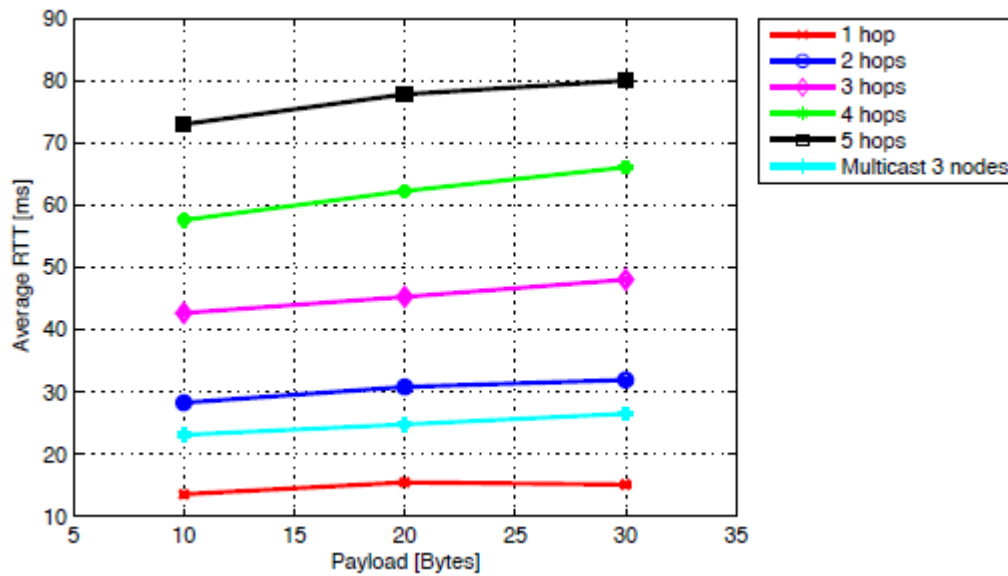


Figura 3. 10: RTT promedio frente al tamaño de la carga útil, para diferentes valores del número de saltos.  
Elaborado por: Autor

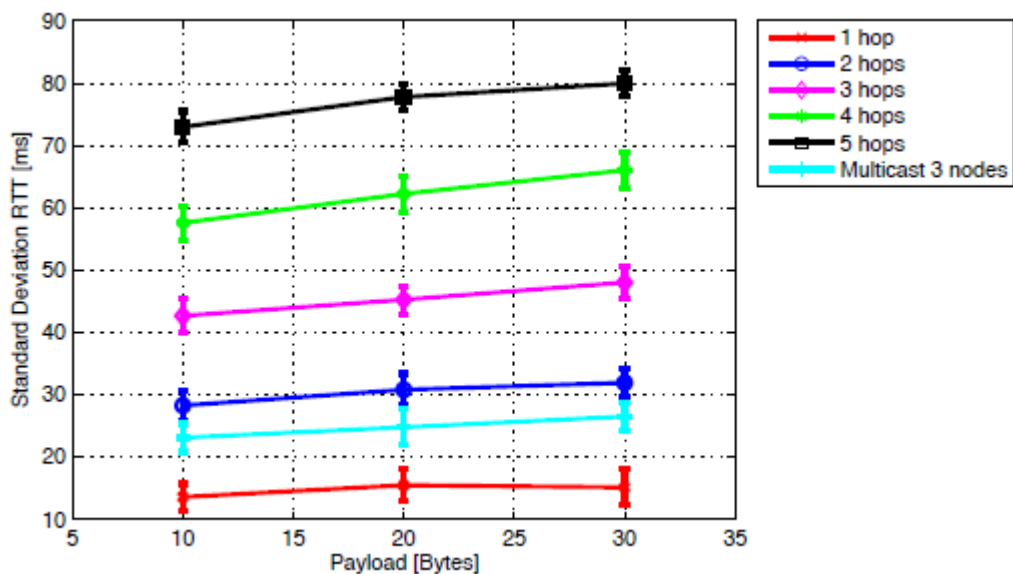


Figura 3. 11: Desviación estándar de los valores de RTT frente al tamaño de la carga útil, para diferentes valores del número de saltos.  
Elaborado por: Autor

En las figuras 3.10 y 3.11 también se bosquejan una curva para el caso de multidifusión. Esto se ha obtenido midiendo el tiempo instantáneo entre la transmisión de un paquete y la recepción del acuse de recibo desde el último destino. En este caso, solo se consideraron tres destinos y se desplegaron

dentro del alcance de radio de la fuente. Obviamente, el promedio y las desviaciones estándar de la RTT son ligeramente mayor que en el caso de unidifusión análogo (un salto). Los CDF correspondientes se representan en la figura 3.12.

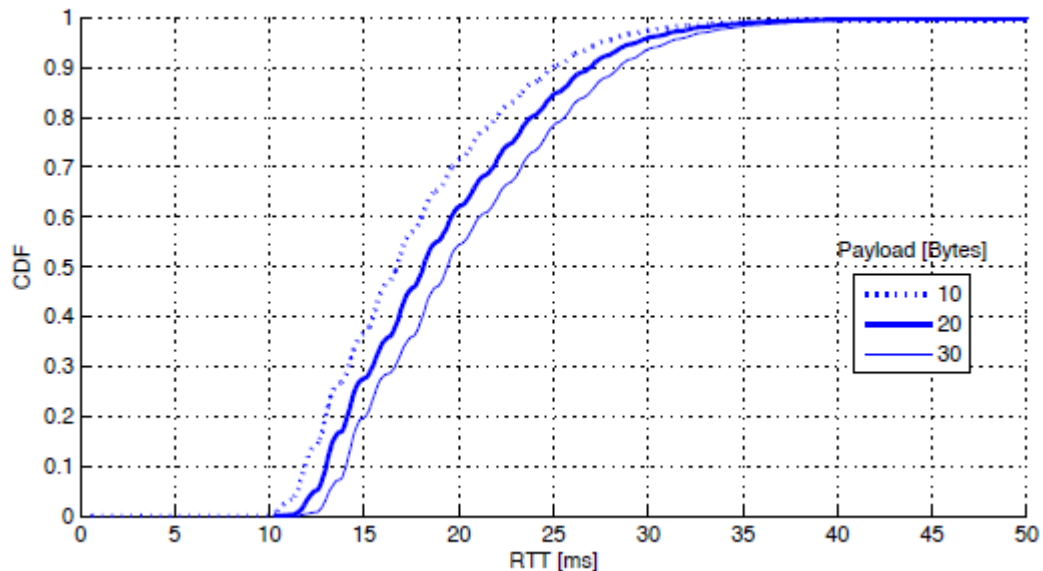


Figura 3. 12: CDF del RTT en el caso de multidifusión para diferentes tamaños de carga útil.

Elaborado por: Autor

El rendimiento en términos de eficiencia se muestra en las figuras 3.13 y 3.14. Más específicamente, en la figura 3.13 se presenta la eficiencia frente al tamaño de la carga útil para diferentes valores de la vida útil de una entrada en la tabla de flujo de WSN, que se denota aquí como TTL, en cambio, en la figura 3.14 se muestran las mismas curvas obtenidas para diferentes valores del intervalo entre transmisiones consecutivas de los paquetes TD, T.

Hay que tener en cuenta que la mayor parte de la ineficiencia se debe a la alta relación entre el tamaño del encabezado y el tamaño de la carga útil.

Finalmente, en las figuras 3.15 a 3.17 se muestran los tiempos de respuesta del controlador a las solicitudes de los nodos para nuevas entradas. Fueron simulados los procesos de generación de solicitudes por parte de los nodos que modelan una red que consta de 50, 60 y 70 nodos. Además, se asumió que inicialmente solo el 10% de los posibles enlaces están activos,



pero fueron aumentándose ese número en un 10% cada 100 solicitudes, y al final se obtiene una red totalmente mallada.

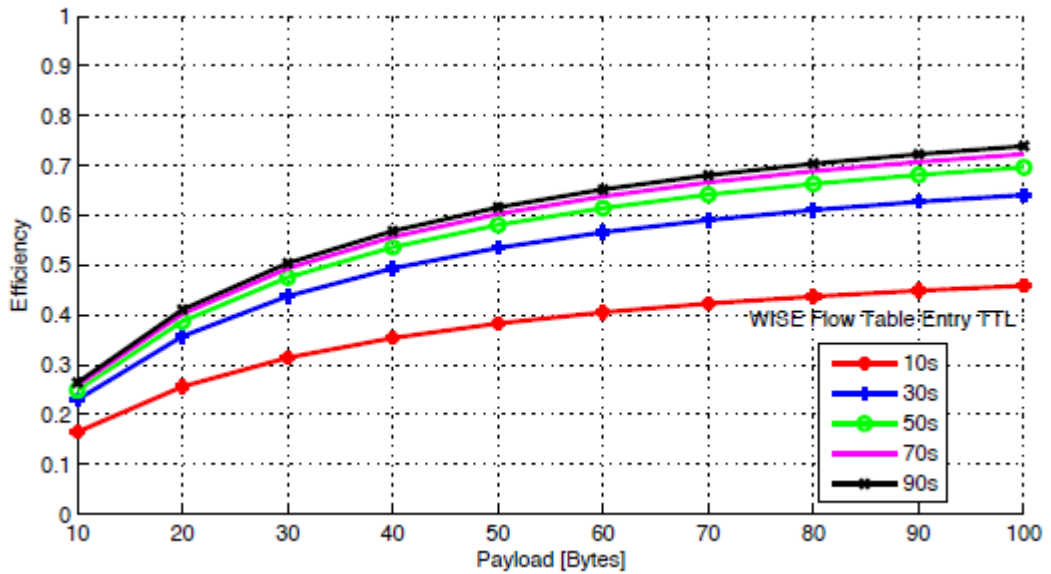


Figura 3. 13: Eficiencia para diferentes valores de entrada de TTL en tabla de flujo WSN.

Elaborado por: Autor

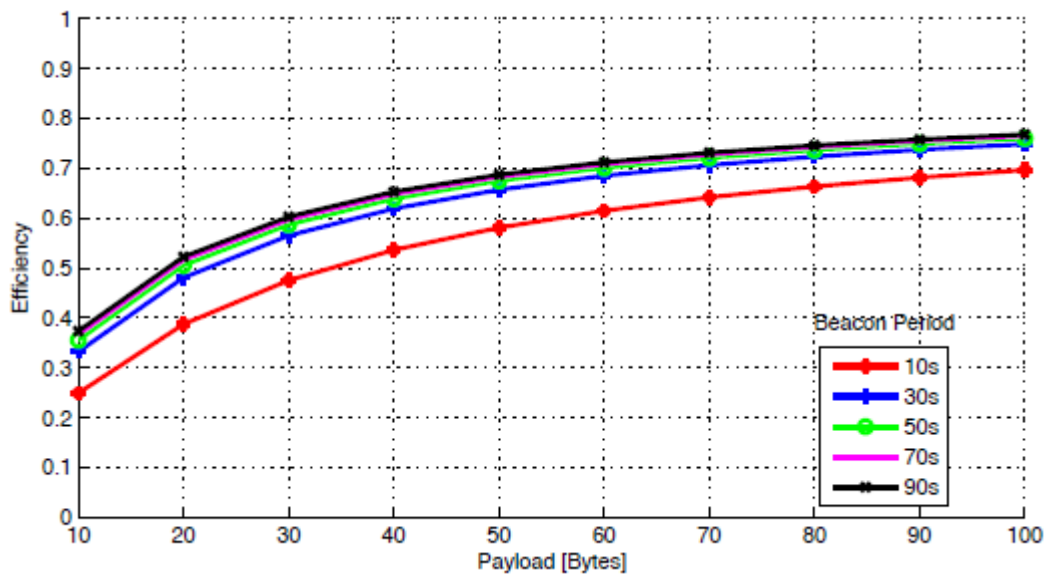


Figura 3. 14: Eficiencia para diferentes valores del período de envío de balizas.

Elaborado por: Autor

Lo que se observa es que hay exageraciones en las tramas que están en correspondencia con un aumento en el número de enlaces que requiere una nueva ejecución del algoritmo Dijkstra. En cualquier caso, el retraso de la

respuesta siempre es inferior a 100 ms. Tal valor podría reducirse aún más ejecutando el controlador en un hardware más potente.

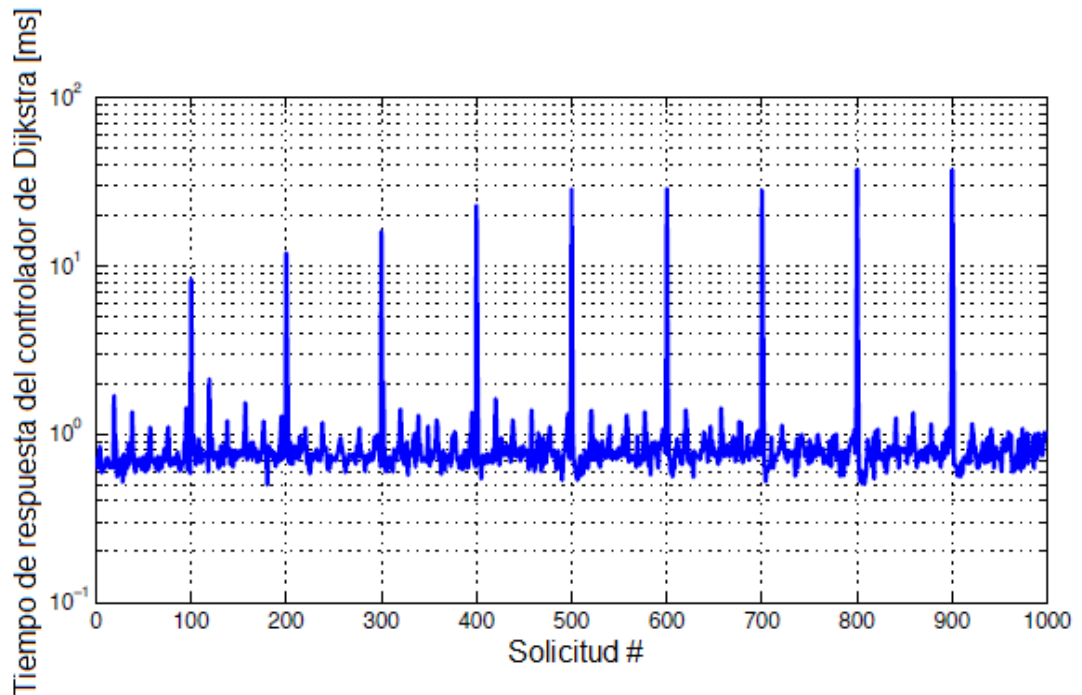


Figura 3. 15: Tiempos de respuesta del controlador para diferentes topologías utilizando 50 nodos.  
Elaborado por: Autor

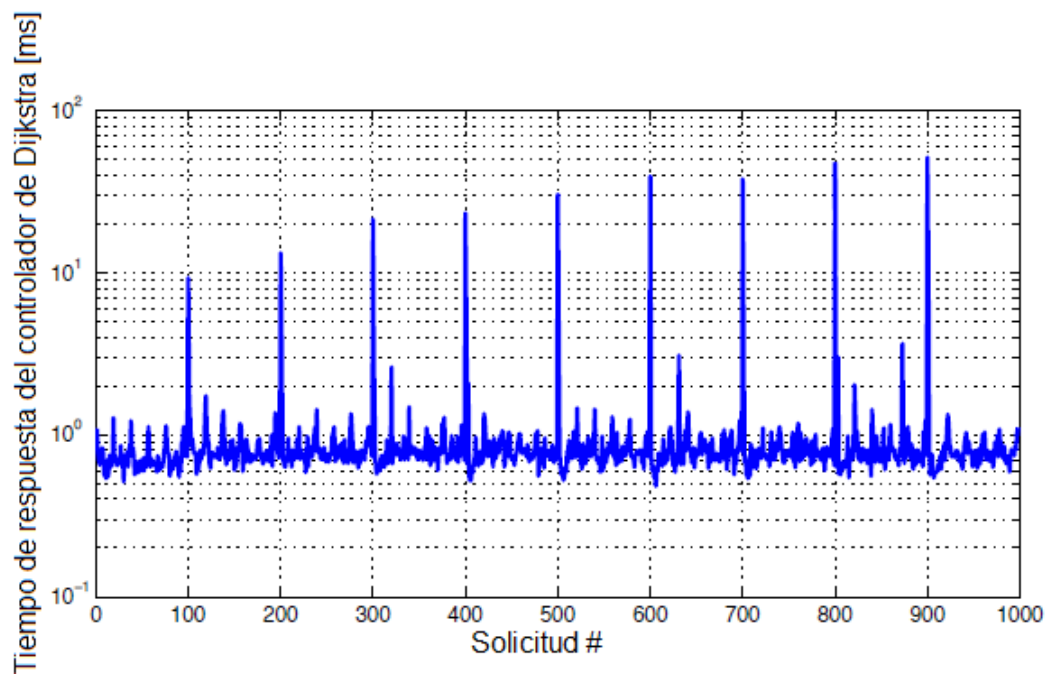


Figura 3. 16: Tiempos de respuesta del controlador para diferentes topologías utilizando 60 nodos.  
Elaborado por: Autor

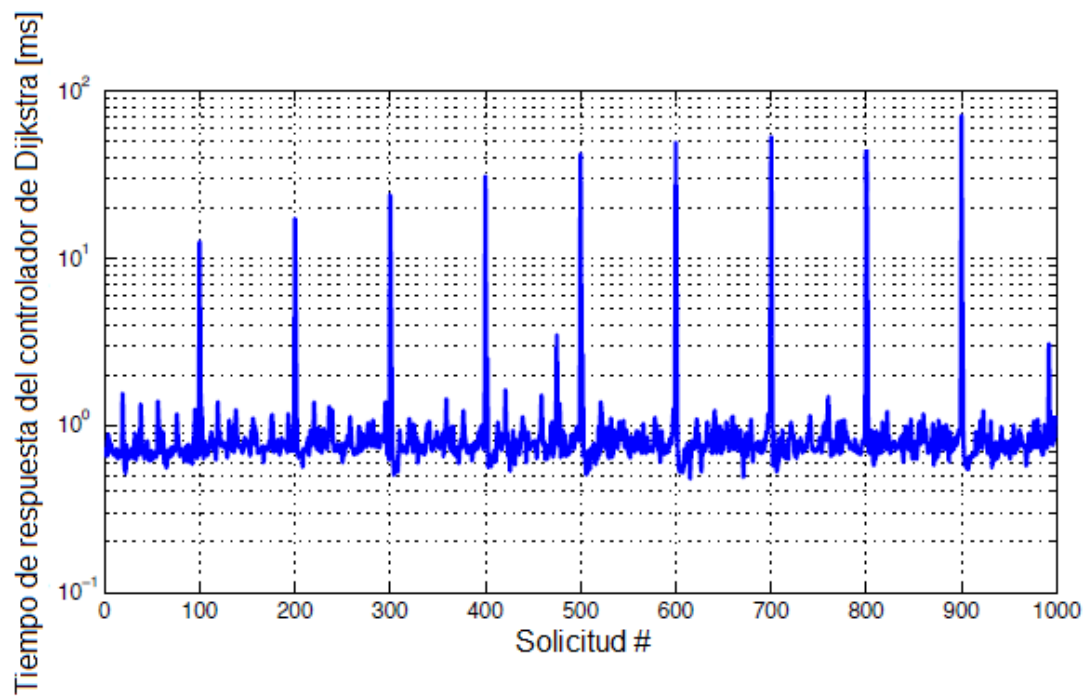


Figura 3. 17: Tiempos de respuesta del controlador para diferentes topologías utilizando 70 nodos.  
Elaborado por: Autor

## Conclusiones

1. La fundamentación teórica fue de gran importancia para el desarrollo del proyecto, en el cual se establecieron las bases teóricas de las redes de sensores inalámbricos basadas en redes definidas por software.
2. Este trabajo de intervención se basa en un enfoque novedoso basado en Redes Definidas por Software (SDN) en dispositivos IoT. Esta solución ha sido ampliamente probada en una variedad de escenarios que muestran que las redes de sensores inalámbricos definidos por software (SD-WSN) tiene muchas ventajas en comparación con las soluciones existentes, En particular, los resultados experimentales muestran que, en términos de condiciones estáticas y cuasi-estáticos, se considera una solución independiente del tamaño de la red, el tamaño de la carga útil, el tráfico generado y la métrica de rendimiento.
3. En este capítulo se presentó a SD-WSN, como una solución de red definida por software para redes de sensores inalámbricos. SD-WSN tiene como objetivo reducir la cantidad de información intercambiada entre sensores y controladores utilizando SDN. Se proporcionan detalles sobre la pila de protocolos SD-WSN, así como los resultados obtenidos de medidas exhaustivas en un banco de pruebas físico. SD-WSN es un enfoque prometedor para la realización de WSN programables.

## **Recomendaciones**

A continuación, se recomiendan dos trabajos a futuro que se pueden desarrollar a partir de la presente investigación:

1. Implementar protocolos de enrutamiento dinámicos en redes de sensores inalámbricos definidos por software (SDWSN).
2. Análisis de la conservación de energía en dispositivos de sensores inalámbricos utilizando redes definidas por software (SDN)

## Bibliografia

- Costanzo, S., Galluccio, L., Morabito, G., & Palazzo, S. (2012). Software Defined Wireless Networks: Unbridling SDNs. *2012 European Workshop on Software Defined Networking*, 1–6. <https://doi.org/10.1109/EWSDN.2012.12>
- Dawans, S., Duquennoy, S., & Bonaventure, O. (2012). On link estimation in dense RPL deployments. *37th Annual IEEE Conference on Local Computer Networks -- Workshops*, 952–955. <https://doi.org/10.1109/LCNW.2012.6424087>
- Dong, P., Zhang, Z., & Tong, F. (2014). Experiment based analysis of ZigBee transmissions under severe Wi-Fi interference. *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, 317–322. <https://doi.org/10.1109/CYBER.2014.6917482>
- Farooq, M. O., & Kunz, T. (2016). Benefits of the IEEE 802.15.4's MAC layer acknowledgements in Ad-Hoc networks: An experimental analysis. *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, 1–7. <https://doi.org/10.1109/MoWNet.2016.7496631>
- Filho, J. Q., Cunha, N., Lima, R., Anjos, E., & Matos, F. (2018). A Software Defined Wireless Networking Approach for Managing Handoff in IEEE 802.11 Networks. *Wireless Communications and Mobile Computing, 2018*, 1–11. <https://doi.org/10.1155/2018/9246824>
- Foukas, X., Marina, M. K., & Kontovasilis, K. (2015). Software Defined Networking Concepts. En M. Liyanage, A. Gurtov, & M. Ylianttila (Eds.), *Software Defined Mobile Networks (SDMN)* (pp. 21–44). John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781118900253.ch3>

- Kamma, P. K., Palla, C. R., Nelakuditi, U. R., & Yarrabothu, R. S. (2016). Design and implementation of 6LoWPAN border router. *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, 1–5. <https://doi.org/10.1109/WOCN.2016.7759025>
- Liang, L., Huang, L., Jiang, X., & Yao, Y. (2008). Design and implementation of wireless Smart-home sensor network based on ZigBee protocol. *2008 International Conference on Communications, Circuits and Systems*, 434–438. <https://doi.org/10.1109/ICCCAS.2008.4657808>
- Ma, K., & Sun, R. (2013). Introducing WebSocket-Based Real-Time Monitoring System for Remote Intelligent Buildings. *International Journal of Distributed Sensor Networks*, 9(12), 867693. <https://doi.org/10.1155/2013/867693>
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., & Turner, J. (2008). OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 69. <https://doi.org/10.1145/1355734.1355746>
- Najeem, J. S., & Krishnan, P. (2019). *International Journal of Recent Technology and Engineering*, 7(6), 686–690.
- Park, Y., Dinh, N.-T., & Kim, Y. (2012). A network monitoring system in 6LoWPAN networks. *2012 Fourth International Conference on Communications and Electronics (ICCE)*, 69–73. <https://doi.org/10.1109/CCE.2012.6315873>
- Pellerano, G., Falcitelli, M., Petracca, M., & Pagano, P. (2013). 6LoWPAN conform ITS-Station for non safety-critical services and applications. *2013 13th International Conference on ITS Telecommunications (ITST)*, 426–432. <https://doi.org/10.1109/ITST.2013.6685584>

Priya, A. V., & Radhika, N. (2019). Performance comparison of SDN OpenFlow controllers. *International Journal of Computer Aided Engineering and Technology*, 11(4/5), 467.  
<https://doi.org/10.1504/IJCAET.2019.100444>

Valdivieso C., A. L., Barona L., L. I., & Garcia V., L. J. (2013). Evolution and Challenges of Software Defined Networking. *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, 1–7.  
<https://doi.org/10.1109/SDN4FNS.2013.6702542>





## DECLARACIÓN Y AUTORIZACIÓN

Yo, **García Vásquez, Gabriel Santiago** con C.C: # 092025650-0 autor del trabajo de titulación: Evaluación de una red definida por software para redes de sensores inalámbricos utilizando el protocolo de descubrimiento de topología y manejo de paquetes, previo a la obtención del título de **Magister en Telecomunicaciones** en la Universidad Católica de Santiago de Guayaquil.

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

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

Guayaquil, 21 de febrero del 2020

f. \_\_\_\_\_

Nombre: **García Vásquez, Gabriel Santiago**

C.C: 092025650-0

<b>REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA</b>			
<b>FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN</b>			
<b>TÍTULO Y SUBTÍTULO:</b>	Evaluación de una red definida por software para redes de sensores inalámbricos utilizando el protocolo de descubrimiento de topología y manejo de paquetes.		
<b>AUTOR(ES)</b>	García Vásquez, Gabriel Santiago		
<b>REVISOR(ES)/TUTOR(ES)</b>	Córdova Rivadeneira, Luis Silvio MSc. Zamora Cedeño, Néstor MSc. Palacios Meléndez, Edwin Fernando MSc.		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Sistema de Posgrado		
<b>PROGRAMA:</b>	Maestría en Telecomunicaciones		
<b>TÍTULO OBTENIDO:</b>	Magister en Telecomunicaciones		
<b>FECHA DE PUBLICACIÓN:</b>	21 de febrero del 2020	<b>No. DE PÁGINAS:</b>	51
<b>ÁREAS TEMÁTICAS:</b>	Teoría de la Comunicación, Sistemas de Comunicaciones		
<b>PALABRAS CLAVES/KEYWORDS:</b>	Redes, Software, Protocolos, Topologías, Sensores, Inalámbrica.		
<b>RESUMEN/ABSTRACT:</b>	<p>El presente trabajo de examen complejo consiste en evaluar una red definida por software (SDN) para redes de sensores inalámbricos (WSN) utilizando los protocolos de descubrimiento de topología y manejo de paquetes. Actualmente, muchos profesionales realizan investigaciones orientadas a redes definidas por software, aunque existe otro grupo de investigaciones donde el enfoque son las redes inalámbricas definidas por software. En el capítulo 1, se presenta la definición y justificación del problema a investigar, así como, el objetivo general, objetivos específicos, la hipótesis y tipo de investigación utilizado en el presente trabajo investigativo. En el capítulo 2, se presenta los fundamentos teóricos de las redes definidas por software y de redes inalámbricas definidas por software. En el capítulo 3, se explica la red definida por software para redes de sensores y se presenta los resultados obtenidos de los escenarios realizados en OMNeT ++. Finalmente, se presentan las conclusiones y recomendaciones del presente trabajo.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> 0989678897	E-mail: <a href="mailto:gabogarcia64@gmail.com">gabogarcia64@gmail.com</a>	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):</b>	<b>Nombre:</b> Manuel Romero Paz		
	<b>Teléfono:</b> 0994606932		
	<b>E-mail:</b> <a href="mailto:mrromeropaz@yahoo.com">mrromeropaz@yahoo.com</a>		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			