UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Sensores inalámbricos en entornos industriales mediante SDN

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Telecomunicación

AUTOR/A: Planes Martínez, Ana

Tutor/a: Sempere Paya, Víctor Miguel

CURSO ACADÉMICO: 2021/2022

Resumen

La industria actual se encuentra en evolución constante, ya se habla de una Industria 4.0, la cual precisa una elevada necesidad de sensorización con requisitos temporales acotados. Por este motivo es necesario el uso de redes de sensores inalámbricos industriales IWSN *(Industrial Wireless Sensor Network*), los cuales permiten extender la funcionalidad y su uso en entornos productivos industriales.

El uso de este tipo redes permite almacenar y gestionar la información obtenida a un bajo coste y con alta flexibilidad. Mediante dichas redes se pueden desarrollar nuevas aplicaciones que requieran alta fiabilidad, baja latencia y cumplan requisitos de calidad de servicio. Además, el uso de nuevas tecnologías software permite el trabajo conjunto de las redes *wireless* y cableadas.

En este trabajo se estudia la interconexión de una red cableada definida por software SDN (*Software Defined Networking*) con una red de sensores inalámbrica. Se llevan a cabo distintas simulaciones y seguidamente el despliegue de la red en equipos reales, llevando a cabo una monitorización para ofrecer una alta calidad de servicio. Por último, se realiza el procesamiento y análisis de los resultados obtenidos.

Resum

La indústria actual es troba en evolució constant, ja es parla d'una Indústria 4.0, la qual precisa una elevada necessitat de sensorització amb requisits temporals delimitats. Per aquest motiu és necessari l'ús de xarxes de sensors inlàmbliques industrials IWSN (*Industrial Wireless Sensor Network*), els quals permeten estendre la funcionalitat i el seu ús en entorns productius industrials.

L'ús d'aquest tipus xarxes permet emmagatzemar i gestionar la informació obtinguda a un baix cost i amb alta flexibilitat. Mitjançant aquestes xarxes es poden desenvolupar noves aplicacions que requerisquen alta fiabilitat, baixa latència i complisquen requisits de qualitat de servei. A més, l'ús de noves tecnologies software permet el treball conjunt de les xarxes *wireless* i cablejades.

En aquest treball s'estudia la interconnexió d'una xarxa cablejada definida per software SDN (*Software Defined Networking*) amb una xarxa de sensors inalàmbrica. Es duen a terme diferents simulacions i seguidament el desplegament de la xarxa en equips reals, duent a terme un monitoratge per a oferir una alta qualitat de servei. Finalment, es realitza el processament i anàlisi dels resultats obtinguts.

Abstract

The current industry is in constant evolution, there is already talk of an Industry 4.0, which requires a high need for sensorization with limited time requirements. For this reason, it is necessary to use industrial wireless sensor networks IWSN (Industrial Wireless Sensor Network), which allow extending the functionality and its use in industrial production environments.

The use of this type of network allows the information obtained to be stored and managed at a low cost and with high flexibility. Through these networks, new applications that require high reliability, low latency and meet quality of service requirements can be developed. In addition, the use of new software technologies allows wireless and wired networks to work together.

In this work, the interconnection of a wired network defined by software SDN (Software Defined Networking) with a wireless sensor network is studied. Different simulations are carried out and then the deployment of the network in real equipment, carrying out monitoring to offer a high quality of service. Finally, the processing and analysis of the results obtained is carried out.

Índice

Índice de Figuras

Índice de Tablas

Capítulo 1. Introducción del trabajo

1.1 IoT

El termino IoT proviene del inglés de *Internet of Things*, también conocido en español como Internet de las cosas. IoT hace referencia a un número ilimitado de objetos interconectados mediante una red, siendo posible establecer una conexión con ellos. Este término recibe gran atención en la actualidad debido al uso masivo de dispositivos conectados, dado que actualmente se disponen de más de 25.000 millones de dispositivos.

Entrando en mayor profundidad, el término proviene debido a la concurrencia de tres tipos de tecnología:

- La tecnología de comunicación inalámbrica.
- La tecnología de los sistemas microelectrónicos.
- La tecnología de los microservicios de Internet.

Estos tres tipos de tecnología permiten establecer una comunicación máquina a máquina, ya sea mediante interacción humana o sin ella. Para ello se establece una conexión a una red y se emplea un gran número de sensores, como se ha realizado en este trabajo a una escala más reducida, para poder tener un proceso o un entono industrial bajo control.

Algunos aspectos sobre la historia de IoT, el término surgió a finales del siglo XX, más exactamente en los años 90, debido al primer objeto que se conectó a internet en 1990. Este objeto fue una tostadora y se considera el primer dispositivo IoT. En la actualidad es incalculable el número de dispositivos IoT existentes.

Figura 1. IoT. [1]

Cada vez es más normal poder establecer una conexión con los objetos, ya sean neveras o una mochila. Esta conexión permite obtener datos de interés para el usuario como, por ejemplo, en el caso de la nevera si se dispone de algún producto o no en ella. Esto permite ofrecer un gran número de facilidades a los usuarios. [2]

Dadas estas pinceladas de IoT, se ve que es necesario el uso de redes inalámbricas de sensores y más en el entorno industrial. Por este motivo se van a estudiar en este trabajo, dadas las grandes perspectivas de futuro que ofrecen, integrándolas en una red cableada SDN.

1.2 Industria

En el trabajo se ha tenido presente que el ambiente de trabajo es un entorno industrial, dónde el uso de sensores inalámbricos ofrece grandes ventajas a los trabajadores. La industria se encuentra constantemente en plena evolución, así pues, es importante tener unos conocimientos de la industria del pasado para tener una mayor percepción.

La primera revolución industrial, también llamada Industria 1.0, tuvo lugar a finales del siglo XIII dónde el agua tenía el papel principal, más concretamente el vapor de agua. La energía proveniente del vapor de agua ayudó a la industria a aumentar su producción y con ello provocó un crecimiento de la economía.

La segunda revolución industrial, también llamada Industria 2.0, tuvo lugar a finales del siglo XIX y principios del XX. En esta revolución industrial el papel principal lo tuvo la energía eléctrica. Con el aumento de la electricidad y el aumento de infraestructuras se pudieron desarrollar las primeras líneas de producción, como la más conocida de Henry Ford.

La tercera revolución industrial, también llamada Industria 3.0, tuvo lugar a finales del siglo XX. En esta revolución industrial el papel principal lo tuvo la automatización de los procesos industriales. Gracias al desarrollo de las computadoras y la programación de los dispositivos electrónicos. Actualmente se puede considerar que esta industria se está quedando desfasada, por consiguiente, ha surgido una nueva revolución industrial.

Desde hace pocos años ya se habla de una cuarta revolución industrial, también llamada Industria 4.0, la cual surgió a principios del siglo XXI. El papel principal lo tiene la hiperconectividad, permitiendo una producción inteligente. En esta cuarta revolución es necesario el uso de dispositivos IoT, la inteligencia artificial y otras tecnologías cómo la realidad aumentada. [3]

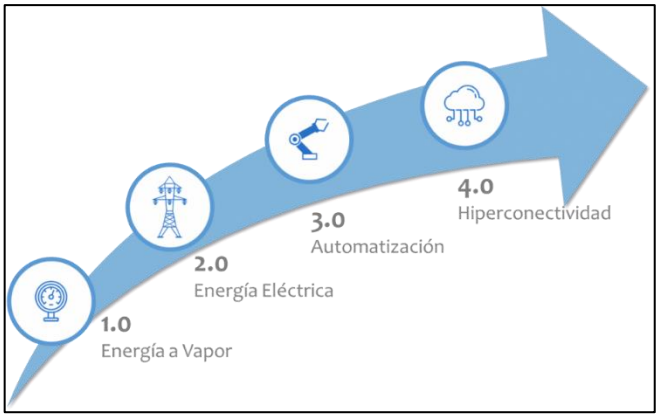


Figura 2. Historia de la industria. [4]

Mediante la sensorización se permite realizar la evolución a esta nueva industria, la cual requiere unos requisitos de temporales acotados. Por este motivo se emplean redes de sensores inalámbricos industriales IWSN (*Industrial Wireless Sensor Network*). Con estas redes se puede realizar una captura masiva de datos, permitiendo un mayor control y una mejora de los procesos industriales, consiguiendo una industria mucho más inteligente. La Industria 4.0 requiere de esta tecnología para cumplir con los requisitos de calidad de servicio que necesitan este tipo de entornos.

En el trabajo se ha tenido todo esto presente, ya que el uso de sensores inalámbricos es necesario para realizar la hiperconectividad, y para ello se va a emplear tecnología SDN tanto en redes cableadas como inalámbricas.

Capítulo 2. Objetivos

Las SDN se encuentran en un momento clave, se ha desarrollado como una tecnología para centros de procesamiento de datos que se ha extendido a todos los puntos de la red cómo WAN o LAN y ahora a las redes de sensores inalámbricos industriales IWSN. En este trabajo se pretende demostrar las ventajas que ofrece, centrándose en la capacidad de garantizar parámetros estrictos de calidad de servicio QoS (*Quality of Service*) entre extremos de la comunicación, realizando una monitorización de la red constante.

Para realizar esta demostración, se realiza una topología de red híbrida empleando redes inalámbricas y cableadas. En la siguiente figura se muestra el escenario que se ha planteado, se observan tanto una red cableada cómo una inalámbrica empleando SDN. Esto permite integrar múltiples tecnologías y tener un control constante de las redes. El control de las redes cableadas e inalámbricas se realiza mediante dos controladores, como se observa en la figura, los que se encargan de asignar los recursos dependiendo de las necesidades de la red y las prioridades.

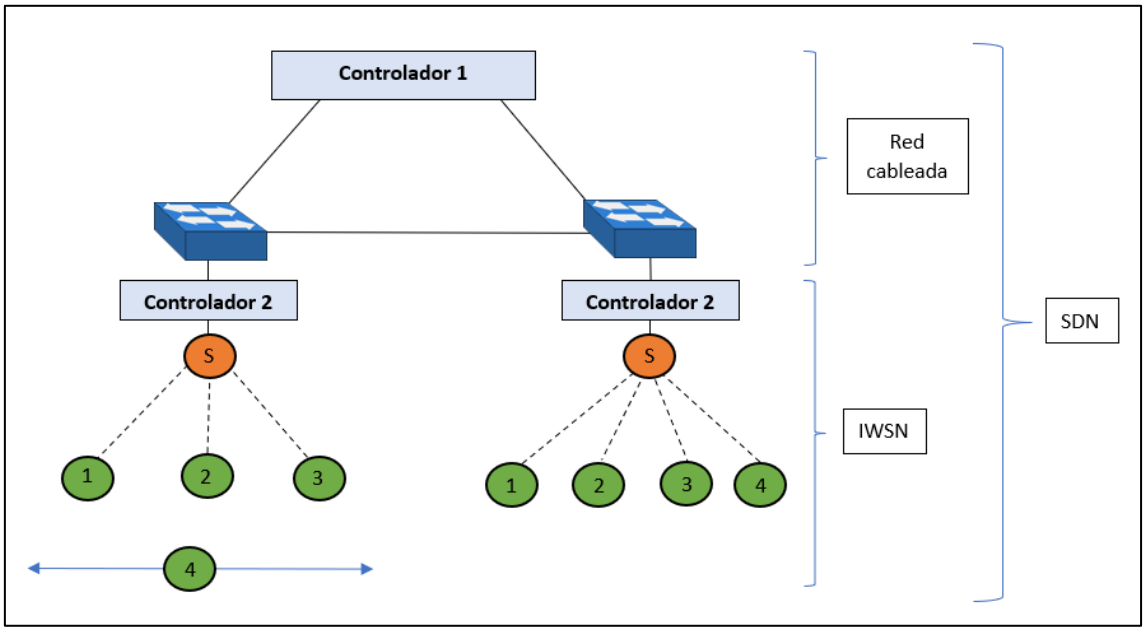


Figura 3. Escenario del trabajo.

Una vez descrito el escenario sobre el cual se va a trabajar, en primer lugar, se debe de realizar la elección del controlador de la red cableada y su correspondiente configuración. A continuación, se procede con la configuración de la red cableada. Para ello será necesario emplear dos switches, que se deben de configurar correctamente para establecer una conexión con el controlador. En cada uno de los switches se conectará una PLC para poder observar la saturación de la red. También dispondrán dos Raspberry que serán las encargadas de hacer la adaptación de la red inalámbrica a la cableada y de conectarse al controlador de la red inalámbrica. Una vez configurada la red cableada se procede a la configuración de la inalámbrica.

Se realiza un análisis del sistema operativo que emplean los nodos. Una vez escogido se procede a la programación de los nodos para que realicen las acciones correspondientes. A continuación, se analizan varios sensores para integrarlos en los nodos y se realiza una reconfiguración para incluir el sensor en el nodo y obtener los datos de interés. Como se observa en la anterior figura, existe un nodo móvil, que va montado en un robot que se configura para que circule sobre una línea azul.

Para finalizar, se realiza la configuración de la transición de la red inalámbrica a la red cableada. Este paso es muy importante y se establece en las Raspberrys, ya que de esta forma se

podrá tener un punto de control secundario para las redes inalámbricas y permite obtener los datos de los nodos gráficamente a través de la herramienta Grafana. De esta forma se obtiene el escenario descrito.

Una vez se tiene el escenario SDN, se realiza un análisis de los datos para obtener las conclusiones y obtener las ventajas que ofrece el empleo de SDN en las IWSN.

Para el desarrollo del trabajo se han establecido los siguientes objetivos:

- Implementación de la red cableada SDN.
- Establecer mecanismos de QoS en la red cableada mediante colas.
- Implementar la red inalámbrica IWSN e integrarla en la SDN.
- Establecer mecanismos de QoS en la red IWSN y monitorizar los sensores.
- Implementar la red SDN-WISE al completo y realizar su monitorización.

Capítulo 3. Metodología

3.1 Gestión del proyecto

Se van a realizar las siguientes pruebas para un correcto montaje del escenario y realizar un mayor aprovechamiento de las herramientas y tecnologías.

- Estudio teórico de las redes SDN y IWSN.
- Simulación del escenario en la herramienta Mininet para la configuración del controlador de la red cableada (ONOS).
- Configuración de los switches para establecer la conexión con el controlador.
- Configuración de las PLCs.
- Montaje completo de la red cableada SDN y obtener las mediciones sin y con QoS.
- Configuración del controlador secundario en las Raspberrys y agregar nuevas funcionalidades (SDN-WISE).
- Pruebas y configuración de los nodos.
- Pruebas e integración de los sensores.
- Configuración del robot móvil.
- Configuración de las herramientas empleadas, como Grafana.
- Montaje completo de la red IWSN y obtener las mediciones.
- Implementar la red SDN-WISE al completo.
- Estudio de los resultados obtenidos.

3.2 Distribución de tareas

Nombre de tarea	Duración	Comienzo	Fin
Estudio teórico	19 días	lun 20/09/21	jue 14/10/21
Configuración controlador ONOS	17 días	vie 15/10/21	dom 07/11/21
Primeras simulaciones Mininet y ONOS	6 días	lun 08/11/21	dom 14/11/21
Configuraión switches físicos y ONOS	6 días	lun 15/11/21	dom 21/11/21
Configuración PLCs	6 días	lun 22/11/21	dom 28/11/21
Montaje red cableada y añadir QoS	11 días	lun 29/11/21	dom 12/12/21
Pruebas e intgración de los sensores	11 días	lun 13/12/21	dom 26/12/21
Pruebas y configuración de los nodos	91 días	lun 27/12/21	dom 01/05/22
Configuración controlador SDN-WISE	101 días	lun 27/12/21	dom 15/05/22
Configuració del robot móvil	6 días	lun 13/12/21	lun 20/12/21
Configuración herramienta Grafana	26 días	mar 21/12/21	mar 25/01/22
Montaje de la red SDN-WISE y pruebas	16 días	lun 16/05/22	dom 05/06/22
Estudio de los resultados obtenidos	21 días	lun 16/05/22	dom 12/06/22
Redacción del trabajo	172 días	vie 15/10/21	dom 12/06/22

Figura 4. Distribución de las tareas.

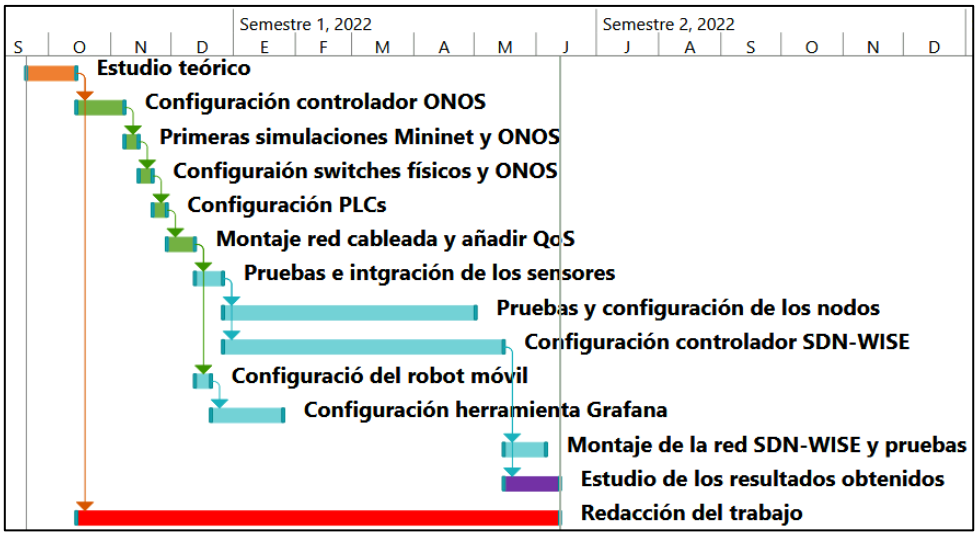


Figura 5. Diagrama de las tareas realizadas durante el trabajo.

Primero, se realiza un estudio teórico de las tecnologías que se van a emplear para tener una mayor noción acerca de ellas al igual que las herramientas que se van a emplear para llevar a cabo el trabajo. Esta tarea se observa en el diagrama al inicio de color naranja.

Una vez realizado el estudio se empieza primero con la red cableada SDN, seleccionando el controlador y su configuración. Se realizan varias pruebas mediante la herramienta de simulación Mininet. A continuación, se configuran los switches físicos y las PLCs y se integran a la red. Una vez finalizado el montaje de la red cableada se testea y se añade calidad de servicio para obtener y analizar los resultados. Las tareas correspondientes a la red cableada SDN se pueden observar en el diagrama de color verde.

Acto seguido, se continua con la red de sensores inalámbrica SDN, empezando seleccionando los sensores a emplear y testeándolos. Una vez seleccionados, se empieza tanto con la configuración del controlador SDN-WISE como con la configuración de los nodos. Al mismo tiempo, se programa el robot que se va a emplear en las pruebas y la herramienta Grafana para visualizar los datos. Una vez se tiene todo listo, se monta la red inalámbrica, se testea y se aplican mejoras. Todas estas tareas relacionadas con la red SDN inalámbrica se pueden observar en el diagrama de color azul.

Por último, se realiza un estudio de todos los resultados obtenidos, como se observa en el diagrama la tarea de color morado. Durante el trabajo se ha ido redactando la memoria con todas las pruebas realizadas, como se observa en el diagrama la tarea de color rojo.

Capítulo 4. Estado del Arte SDN

4.1 SDN (*Software Defined Networking*)

SDN son redes definidas por software, como bien indica su nombre, son redes programables y automatizadas. La principal característica de estas redes es la separación del plano de control y el plano de datos. Las redes SDN pueden ser tanto virtuales como físicas, permitiendo la administración y configuración de los dispositivos de una forma más centralizada debido a dicha separación de planos. Permiten una mayor escalabilidad, ya que ofrecen una infraestructura más sencilla, una mayor velocidad, una red inteligente abierta más flexible y permite una reprogramación. [5]

Durante estos años las plataformas basadas en la nube han permitido un mayor impulso de las redes SDN, ya que ofrecen una sencillez y una rapidez a la hora de gestionar redes de gran tamaño. Todas las facilidades que nos ofrece SDN se deben principalmente a la separación del plano de datos y el de control. Las principales características de SDN son las siguientes [6]:

- Permite una centralización de la administración de la red.
- Ofrece una mayor agilidad debido a la separación del plano de control y el plano de datos.
- Permite la programación de la infraestructura.
- Permite una mayor automatización debido a la programación.

Vistas las principales características de las redes SDN, es inevitable no pensar en su uso en los dispositivos IoT. El creciente uso de la tecnología IoT hace que las redes SDN estén en auge ya que permite una gestión más centralizada de los dispositivos y reduce los embotellamientos producidos por IoT, algo muy positivo en entornos industriales.

La infraestructura es la principal diferencia entre las redes tradicionales y las redes SDN, pero no solo existe esa diferencia, otra es la seguridad. Las redes tradicionales se basan en hardware y las redes SDN se basan en software. Los principales objetivos de las redes SDN son los siguientes:

- Conexiones con gran velocidad.
- Gran agilidad debido a la infraestructura.
- Ofrecer a los clientes mayor calidad.
- Ofrecer gran cobertura.
- Reducir los costes de CAPEX y OPEX al interesado.

Una vez repasadas las cualidades de las redes SDN, se va a hablar más en profundidad sobre la arquitectura y los protocolos.

4.1.1 Arquitectura

En SDN la arquitectura está formada por tres capas: la de aplicación, la de control y la de infraestructura. Como ya se ha comentado anteriormente, existe una separación del plano de control y del de datos. El plano de control está formado por la capa de aplicación y la capa de control; el plano de datos está formado por la capa de infraestructura. A continuación, se detalla con mayor profundidad cada capa:

- Plano de control: permite la gestión y el control de la red. Dentro del plano de control se encuentran las siguientes capas:
  - Capa de aplicación: informa de los requisitos a la red. Existen múltiples aplicaciones como la de mantenimiento o la de seguridad. Esta capa se comunica con la capa de control mediante APIs, permitiendo el intercambio de mensajes.
  - Capa de control: se puede considerar el núcleo central de la red, donde se localiza el controlador SDN. La capa de control es la encargada de la gestión de la red y de configurarla, permitiendo añadir flujos de tráfico. La capa de control es la

encargada de comunicarse con la capa de infraestructura, el plano de datos. La comunicación con los elementos de la red se debe realizar para la monitorización de la red, y no solo con elementos de red, sino también es necesaria una comunicación con otros controladores SDN. Esta comunicación es muy importante, ya que se trata de un elemento delicado en la red.

- Plano de datos: es donde se ubican los elementos hardware de la red. Dentro del plano de datos se encuentra la capa de infraestructura:
  - Capa de infraestructura: encamina y comunica los dispositivos, como los nodos de la red. Existe una comunicación de la capa de infraestructura con la capa de control, el plano de datos con el plano de control, mediante varios protocolos SDN, como OpenFlow.

En la siguiente figura se muestra de forma más detallada los planos y las capas que se han explicado anteriormente, así como la comunicación entre ellas. [7] [8]

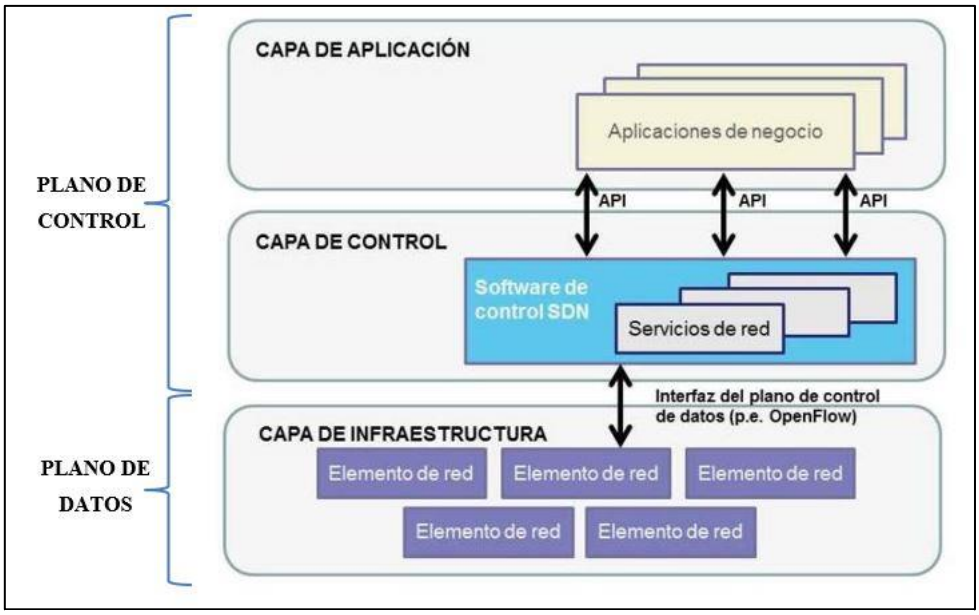


Figura 6. Arquitectura SDN.

4.1.2 Protocolos SDN

Existen varios protocolos de infraestructura SDN conocidos, como los que se van a nombrar a continuación, pero en el trabajo solo se va a emplear uno de ellos [9]:

- POF es una tecnología nueva la cual permite liberar de funciones al elemento de reenvío del plano de datos. De esta forma el elemento de reenvío no necesita saber el formato del paquete y es el elemento de control el encargado de realizar el análisis del paquete.
- ForCES permite una gestión más flexible en cuanto a la planteada tradicionalmente, pero manteniendo la arquitectura. Se permite que el elemento de control sea un elemento más de la red, pero con los planos separados.
- OpFlex tiene como objetivo mejorar la escalabilidad y para ello reparte tareas de administración complejas a los elementos de reenvío del plano de datos.
- OpenFlow es el protocolo que se va a emplear en este trabajo, por este motivo se le va a dedicar el siguiente punto de la memoria para explicarlo con mayor profundidad.

4.1.2.1 OpenFlow

OpenFlow surgió a raíz de una investigación de la Universidad de Stanford. Se estaba estudiando que protocolo emplear en la red del campus diariamente. Principalmente se basa en un switch OpenFlow, las tablas internas de flujo y una interfaz que supervisa los flujos.

En este trabajo se ha escogido el protocolo OpenFlow ya que es el protocolo predominante en las SDN y también uno de los primeros. Se emplea para comunicar el plano de datos y el plano de control, como se ha comentado en puntos anteriores.

La primera versión de OpenFlow surgió en 2011, fue la versión inicial del grupo de investigación. En el mismo año surgió la versión 1.2, esta versión ya a manos de la ONF, la cual se encarga del control y supervisión del protocolo. A partir de entonces se han ido desarrollando más versiones, hasta la 1.5.1. Esta última versión fue lanzada en 2015 y es la última lanzada a fecha de hoy. A continuación, se va a mostrar una tabla con las distintas versiones disponibles, el año de lanzamiento y un breve resumen.

Versión	Año de lanzamiento	Resumen
1.1	2011	La versión más completa al ser la primera. También cuenta con las tablas de grupo y el uso de múltiples tablas de flujo en un conmutador.
1.2	2011	Se añade el uso de direcciones IPv6 y permite la conexión a diversos controladores por parte del conmutador.
1.3	2012	Medidores de flujo para añadir funciones de supervisión y calidad de la red.
1.4	2013	Incluye sincronización en las tablas de flujo.
1.5	2014	Aporta mayor flexibilidad, pero más complejo, como un mayor soporte a parte de Ethernet.

Tabla 1. Resumen versiones OpenFlow.

OpenFlow ofrece múltiples ventajas, una de las más importantes es poder tener separado el plano de control en el ordenador y a parte el plano de datos, esto permite que en caso de que surja algún imprevisto que los dispositivos no sepan gestionar, el plano de control ubicado en el ordenador sea quien tome las decisiones oportunas. Esto permite tener una separación entre el software y el hardware. Esta es la principal ventaja que ofrece este protocolo, pero no la única, a continuación se van a señalar los beneficios que ofrece:

- El control centralizado permite configurar la red de forma más rápida y sencilla.
- Una topología física puede contener distintas capas lógicas.
- El coste de una configuración distribuida es más elevado que el de la configuración centralizada.
- Los dispositivos hardware tienen un menor precio al requerir menos capacidades.
- Reducción del CAPEX y OPEX.
- Tiempos de implementación más reducidos.
- Aportan una mayor fluidez, dinamismo y rapidez.

Para el desarrollo del trabajo se ha tenido presente que los switches deben de ser compatibles con OpenFlow. Esto es un aspecto importante a tener en cuenta, de forma resumida, deben de permitir separa el tráfico. La peculiaridad de este tipo de switches es que están formados por los siguientes 3 bloques, como se muestra en la siguiente figura:

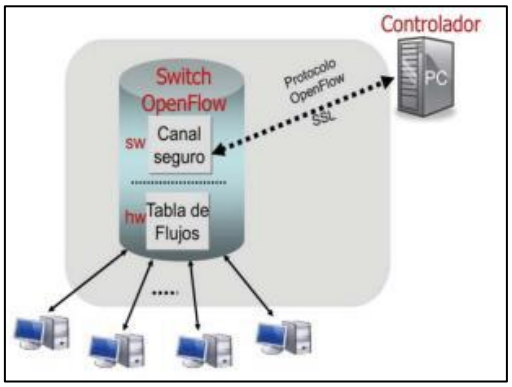


Figura 7. Switch OpenFlow. [10]

- Tablas de flujo: tienen la capacidad de gestionar el tráfico entrante y tratar cada uno de ellos de forma distinta.
- Canal seguro: se ubican en el interior del switch con la finalidad de conectarlo con el controlador mediante una conexión que suele ser habitualmente TLS o TCP. De esta forma, cuando llega un paquete desconocido el controlador puede indicarle al switch como tratarlo.
- El controlador: responsable de las tablas de flujo. Es el encargado de administrar y controlar la red.

En el mercado existen dos tipos de switches:

- Los OpenFlow-*only*: este tipo de switches disponen de una o más tablas de flujo, cuantas más tablas, mayor procesamiento del paquete. Funcionan de la siguiente forma, al llegar un paquete se analiza la cabecera. Seguidamente se comprueba si existe en las tablas de flujo una entrada que coincida con dicho paquete. En caso de que sí exista, se realiza la operación indicada. En caso de no existir, se comprueba en la tabla *Miss-Table* si se debe de mandar el paquete al controlador y si no el paquete se elimina.
- Los híbridos: este tipo de switches incluyen también el enrutamiento Ethernet o IP. Es necesario emplear un mecanismo para diferenciar ambas gestiones del paquete.

Dado que existen dos clases de dispositivos, existen dos arquitecturas posibles. En ambos casos existe la figura del controlador SDN, el switch OpenFlow y los dispositivos finales.

En la siguiente figura se puede observar, en primer lugar, la arquitectura empleando únicamente switch OpenFlow-*only*. En segundo lugar, se observa la arquitectura empleando tanto switches híbridos como tradicionales.

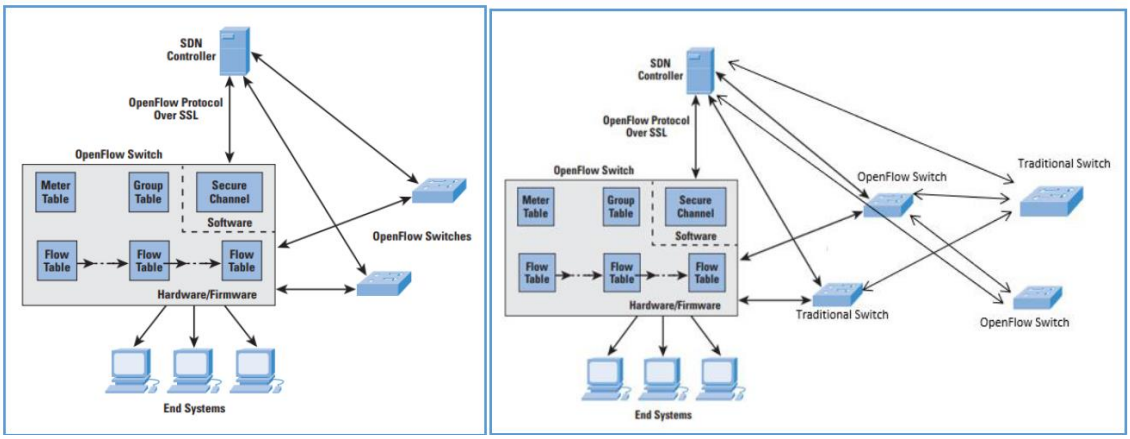


Figura 8. Uso de switches OpenFlow y uso de switches OpenFlow y tradicionales. [11]

En este trabajo se van a emplear switches OpenFlow-*only* ya que no se van a emplear switches tradicionales.

Capítulo 5. Estado del Arte WSN

5.1 WSN *(Wireless Sensor Networks)*

Las WSN se pueden determinar cómo redes inalámbricas autoconfiguradas y sin infraestructura. Se emplean para la observación de condiciones físicas y ambientales (temperatura, movimiento, presión, humedad…). Dicha información se envía a un centro de control, encargado de realizar el análisis y monitoreo. Este tipo de redes suelen estar formadas por un gran número de nodos con sensores. [12]

Las WSN tienen sus orígenes en 1980 a través del programa DSNs (*Distributed Sensor Networks*) llevado a cabo por el departamento de defensa americano.

Los nodos están formados sobre todo por 4 bloques:

- El sensor.
- Transceptor de radio.
- Dispositivo informático.
- Componente de potencia.

La principal desventaja de este tipo de redes es la limitación de recursos, de velocidad de procesamiento y la limitación del ancho de banda. Así pues, es necesario realizar una correcta organización y configuración de la infraestructura para aprovechar los recursos disponibles al máximo.

Otro de los puntos a tener en cuenta es la batería, en las WSN la batería suele ser limitada. Por esta razón es necesario realizar operaciones con un ciclo de trabajo reducido, realizar un enrutamiento óptimo…

Por otro lado, los sensores incorporados tienen la posibilidad de enviar datos de forma seguida o cada cierto periodo de tiempo. También existe la posibilidad de que el nodo actúe cuando se cumplen ciertas condiciones.

5.1.1 Características, ventajas y desventajas

Las principales características que presentan las WSN son las siguientes:

- Se tiene presente en todo momento el consumo de la batería, algo muy importante a tener en cuenta en IoT. Se realizan restricciones en la batería.
- Las WSN deben de ser capaces de actuar frente a fallos en los nodos.
- Permite una escalabilidad.
- Sencillas de usar.

Estas son las principales características de las WSN, pero también es importante tener en cuenta las principales ventajas y desventajas que ofrece.

Ventajas:

- Se pueden emplear en sitios de difícil acceso.
- Fácil implementación e instalación, ya que no es necesario extender cable.
- Permite agregar nuevos nodos mediante la reconfiguración remota.
- Permite una centralización de la red con el uso de un nodo central, *sink*.
- Precio de implementación reducido.

Desventajas:

- La velocidad de comunicación es menor que en el caso de las redes cableadas.
- Presenta una mayor complejidad que las redes cableadas.
- Puede sufrir interferencias procedentes del entorno, y más en entornos industriales.

5.1.2 Estructura de las redes

Existen varias topologías de red, permitiendo que el control de la red sea centralizado, descentralizado o distribuido. A continuación, se van a describir las distintas estructuras para las WSN:

- Topología en estrella: también conocido como punto a multipunto. La gran ventaja que ofrecen este tipo de redes es el ahorro de la batería en los dispositivos finales. Esto se debe a que solo existe un enlace de comunicación con el punto de control. La gran desventaja de esta topología es la necesidad de que todos los dispositivos se encuentren dentro de la zona de cobertura del nodo central, ya que solo tienen un punto de conexión. Este tipo de topología presenta un control centralizado.

- Topología en malla: permite la transmisión entre nodos, ampliando la zona de cobertura al no existir un único punto de conexión. Esta topología permite realizar comunicaciones multisalto. Ofrece una gran escalabilidad y una mayor redundancia. La desventaja en este caso es el mayor consumo de batería. En este caso el control puede ser distribuido.

- Topología híbrida: este tipo de topología es una combinación de estrellas distribuidas en una malla. Esto permite una comunicación más sólida y un ahorro en el consumo de la batería. El control en este caso suele ser de forma descentralizada, ya que en cada agrupación en estrella puede ubicarse un nodo de control.

- Topología en árbol: esta topología tiene forma jerárquica, ya que en la parte superior se encuentra el nodo central. En la parte inferior se encuentran distintas capas con el resto de los dispositivos. En este caso el control suele ser centralizado.

En la siguiente figura se muestran las distintas topologías nombradas anteriormente.

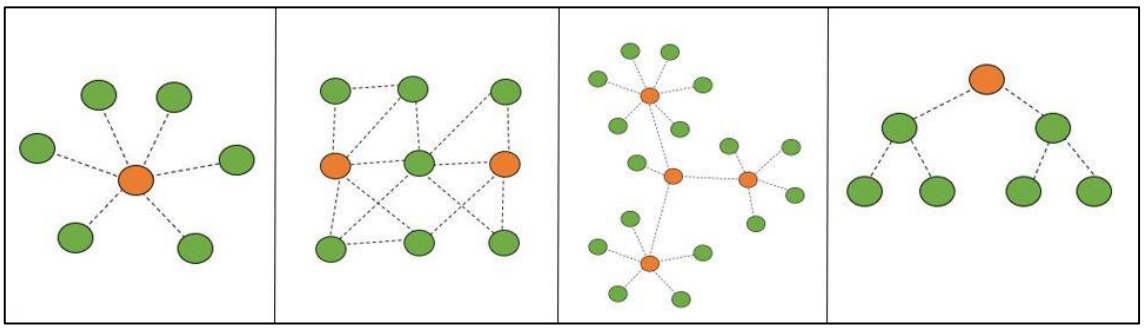


Figura 9. Estructuras de las WSN: a) topología en estrella, b) topología en malla, c) topología híbrida, d) topología en árbol.

5.1.3 Aplicaciones

Las redes WSN se emplean para distintas aplicaciones por su gran utilidad, las aplicaciones más destacadas son las siguientes:

- Aplicaciones militares: en el mundo militar las redes WSN son de gran utilidad ya que permiten un control y monitorización de una zona. El ámbito militar fue donde surgió el origen de las redes WSN debido a su gran necesidad. Este tipo de redes permite mejorar

las maniobras militares ya sea en cuanto a ataque o defensa, por ejemplo, mediante sensores de movimiento.

- Aplicaciones médicas: permiten monitorizar a los pacientes y tener un mayor control de las constantes delicadas. Una gran ayuda para el personal médico ya que facilita el trabajo y ofrece una mayor atención al paciente pudiendo crear alertas en caso de que algún parámetro salga de los límites prescritos por el personal médico.

- Aplicaciones industriales: este es un gran campo a tener en cuenta, ya que es uno de los sitios donde más se emplean las WSN actualmente, o también conocidas como IWSN (*Industrial Wireless Sensor Network*). Este tipo de redes han ido en aumento debido a las prestaciones que ofrece, como la calidad de servicio, la robustez y la optimización del consumo de energía.

El trabajo se ha centrado en aplicaciones industriales, ya que las IWSN ofrecen un gran abanico de posibilidades a desarrollar.

5.1.4 Protocolos

Dado que el principal objetivo de las redes WSN es reducir el consumo de batería, existen distintos protocolos que permiten su optimización. Los principales protocolos empleados en WSN son los siguientes:

- ZigBee: hace referencia a un conjunto de protocolos basado en IEEE 802.15.4. Se emplea para realizar comunicaciones inalámbricas como en las redes WSN con el objetivo de reducir el consumo de energía de los dispositivos.
- IEEE 802.15.4: hace referencia a los protocolos tanto de nivel MAC como de nivel físico. El principal objetivo también es reducir el consumo energético en aquellas redes con una carga de tráfico baja. Más concretamente, se va a emplear el estándar IEEE 802.15.4e, el cual es una evolución de este. En el siguiente punto se hablará más en detalle.
- WirelessHart: también es una tecnología basada en IEEE 802.15.4, diseñada para la autoorganización y la autoreparación en una red de tipo malla. Fue aprobado en 2007 y está pensada para la optimización de los recursos en las redes inalámbricas.
- 6LowPAN (IPv6 *Low-power wireless Personal Area Network*): permite emplear IPv6 sobre redes basadas en IEEE 802.15.4 ya que tiene libertad en la capa física y en la banda de frecuencia. En este caso, no es de interés ya que están más enfocadas en el hogar y en el ámbito personal, y no en el industrial. [13]

5.1.4.1 IEEE 802.15.4

El estándar IEEE 802.15.4 está ideado para redes inalámbricas con una baja tasa de tráfico ofreciendo un consumo bajo y una flexibilidad en la red a un coste reducido. No está ideado para redes de largo alcance, ya que ofrece una cobertura de 10-30 metros. Aunque su idea inicial era emplearse en redes inalámbricas de área personal WPAN (*Wireless Personal Area Network*), se ha demostrado que es totalmente compatible con las WSN al ofrecer los recursos necesarios para ellas. [14]

Este estándar fue desarrollado en octubre de 2003 por el IEEE (*Institute of Electrical and Electronics Engineers*) cubriendo los protocolos de nivel físico y nivel MAC.

Existen dos clases de dispositivos:

- Dispositivo de función completa FFD (*Full Function Device*): este tipo de dispositivo puede ser empleado como coordinador o como un nodo simple.
- Dispositivo de función reducida RFD (*Reduced Function Device*): estos dispositivos únicamente pueden actuar como nodos simples.

Principalmente existen dos clases de redes:

- Las redes punto a punto: en este tipo de redes solo hay un coordinador, siendo la red más centralizada pero más reducida ya que el resto de los nodos deben de estar dentro de la zona de cobertura del coordinador. Este tipo de redes tienen un gran rendimiento y una alta carga de trabajo por parte del coordinador.
- Las redes en estrella: son redes más aconsejadas para el despliegue de WSN. No existe un nodo central, ya que el nodo coordinador puede ubicarse en un extremo de la red. El resto de los nodos pueden comunicase entre ellos, reduciendo la carga de tráfico en el nodo coordinador.

También es posible elaborar redes en malla, formando un árbol de clústeres, con un coordinador en cada clúster y uno principal.

Las principales características son las siguientes:

- Existen tres rangos de transmisión: a 868 MHz 20 Kb/s, a 915 MHz 40 Kb/s y a 2,4 GHz 250 Kb/s.
- Permite un alcance de hasta 30 metros.
- La latencia se encuentra por debajo de los 15 ms.
- Existen varios canales: en el caso de 868 MHz hay 1 canal, en 915 MHz hay 10 canales y en 2,4 GHz 16 canales.
- El canal de acceso puede ser CSMA/CA y CSMA/CA ranurado.
- Permite una temperatura mínima de -40ºC y una máxima de 85ºC en entornos industriales.

En cuanto a la capa física, se puede observar en la siguiente figura las tres bandas de frecuencia en las que opera. En el caso de la banda de 868 MHz solo hay un canal, el canal 0. En la banda de 915 MHz se dispone de 10 canales con una separación de 2 MHz entre los canales. En la banda de 2,4 GHz hay 16 canales de 2 MHz y al ser una banda libre la separación entre ellos aumenta a 5 MHz, de esta forma se evitan interferencias.

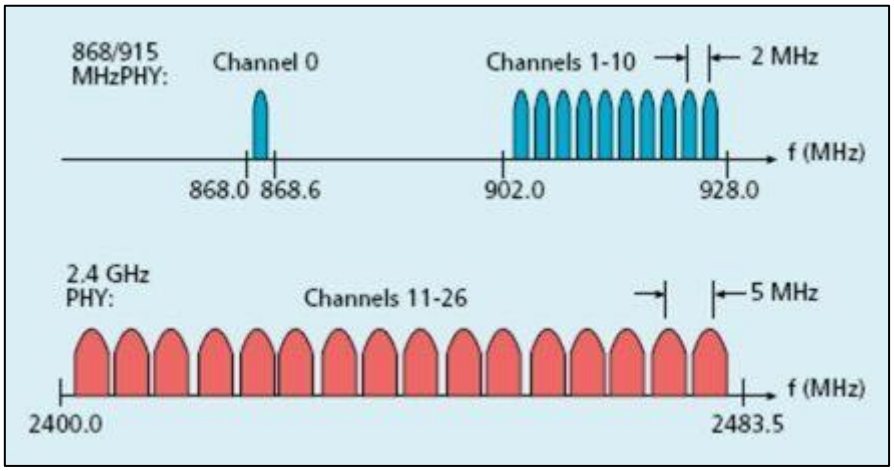


Figura 10. Bandas de frecuencia IEEE 802.15.4. [15]

En la siguiente tabla se muestra de forma más resumida las principales características de cada banda de frecuencia. Se muestra también la técnica de ensanchado en cada banda para evitar interferencias al igual que la modulación empleada.

Banda de frecuencia	Número de canales	Técnica de ensanchado	Modulación	Tasa de símbolos por canal (kbaud)	Tasa de bits por canal (kb/s)
868 MHz	1	Binary DSSS	BPSK	20	20
915 MHz	10	Binary DSSS	BPSK	40	40
2,4 GHz	16	16-array DSSS	O-QPSK	62,5	250

Tabla 2. Características IEEE 802.15.4.

Por otro lado, en la capa de control de acceso al medio MAC (*Media Access Control*) existe la posibilidad de emplear o no mensajes de ACK (*Acknowledgement*) para una mayor fiabilidad del canal. Su uso es optativo ya que hay escenarios donde su uso se puede prescindir para obtener un mayor ahorro de la batería al reducir las comunicaciones. En la capa MAC existen dos modos de ejecución [16]:

- Empleando Beacons (*Beacon enabled*): de esta forma existe una sincronización de los dispositivos de la red ya que existe un tiempo de transmisión exclusivo para cada uno. Para ello se emplea el algoritmo de planificación como CSMA/CA ranurado. Los nodos solo podrán transmitir en el periodo asignado, el resto del tiempo permanecen a la espera del siguiente periodo reduciendo así su consumo de batería.
- Sin emplear Beacons (*Beaconless*): en este caso no existe una asignación temporal para acceder al canal. Se emplea el algoritmo CSMA/CA y cada dispositivo disputa por transmitir.

Para el empleo de las WSN en el ámbito industrial, se desarrolló una actualización del estándar surgiendo así una nueva versión en 2012, el estándar IEEE 802.15.4e.

El nuevo estándar incorpora varios protocolos MAC pensados para un ambiente industrial cómo LLDN, DSME y TSCH. Estos nuevos protocolos permiten un ahorro en la batería de los dispositivos, obtener transmisiones más confiables y redes más deterministas permitiendo una mejor adaptación en aplicaciones industriales. En este trabajo se va a emplear el protocolo TSCH, el cual se comenta a continuación, ya que ofrece un mayor rendimiento.

5.1.4.1.1 TSCH (*Time Slotted Channel Hopping*)

TSCH es un protocolo MAC que incorpora el estándar IEEE 802.15.4e. Este protocolo permite programar tanto la transmisión como la recepción de los datos, evitando que se produzca pérdida de información debido a las colisiones e interferencias. De esta forma, ofrece comunicaciones con una mayor confiabilidad y un mayor rendimiento en cuanto a consumo de batería y latencia, ofreciendo una mejora en la QoS. [17]

El protocolo realiza una planificación mediante una matriz llamada *slotframe*, en la cual las filas representan los desplazamientos del canal (*Channel Offset*), evitando las interferencias entre las comunicaciones, y las columnas representan las ranuras temporales (*Timeslot*). Las ranuras temporales suelen tener una duración de 10 ms, tiempo suficiente para transmitir una trama de hasta 127 bytes, recibir el ACK y procesarlo. Dada dicha matriz, se puede planificar cuando un nodo va a transmitir o recibir un paquete y cuando puede apagar la radio. Para ello, los dispositivos se deben de sincronizar mediante el paquete EB (*Enhanced Beacon*). Dicho paquete les permite a los dispositivos conocer el ASN (*Absolute Sequence Number*) para obtener una referencia temporal y el intervalo de tiempo actual. El ASN se emplea juntamente con el *Channel Offset* para determinar en qué canal físico real se debe realizar la comunicación, como se observa en la siguiente ecuación, siendo un valor entre 0 y 15 ya que se dispone de hasta 16 canales útiles en la banda de 2,4 GHz.

$$Canal\ físico = F\{(ASN + ChannelOffset) mod\ N_c\} \quad (1)$$

Donde:

- $F\{\ \}$ es una función biyectiva.
- ASN representa el número de *Timeslot* absoluto.
- N_c representa el número de canales físicos disponibles.

A continuación, se muestra en la siguiente figura, la matriz descrita anteriormente, en este caso se van a emplear 4 *Cahannel Offset* y 11 *timeslots*. Dentro de cada *timeslot* se pueden producir varias transmisiones ya que se disponen de hasta 4 canales, al igual que dentro de cada canal pueden producirse varias transmisiones en varios *timeslots*. Para garantizar que exista una rotación en todos los canales de los intervalos de tiempo, el número de *timeslots* debe de ser primo.

		Timeslots										
		0	1	2	3	4	5	6	7	8	9	10
Channel Offset	0											
	1					**1->3**						
	2										**1->3**	
	3											

Figura 11. *Slotframe* TSCH.

Las celdas del *slotframe* pueden ser de dos tipos, siempre teniendo presente que solo puede transmitir un dispositivo:

- Celda exclusiva: este tipo de celda se emplea para cuando la transmisión es *unicast*, un dispositivo transmite y otro recibe.
- Celda compartida: este tipo de celda se emplea para cuando la transmisión es *broadcast*, un dispositivo transmite y reciben varios.

Capítulo 6. Entorno de trabajo

Dada la breve explicación que se ha comentado en el apartado de objetivos, se van a comentar los distintos elementos y herramientas que se emplean para llevar a cabo el trabajo.

6.1 Controlador ONOS

ONOS (*Open Network Operating System*) es un controlador SDN que permite tener un control de la red, pudiendo tener un control en los enlaces, nodos y demás. También admite la transición de redes heredadas.

ONOS surgió como un proyecto en 2012 en ON.Lab (*Open Networking Lab*), en el cual se presentó el primer prototipo en 2013. ONOS se lanzó en 2014 por ON.Lab y otros socios del mundo de la industria. En 2015 se produjo un cambio ya que la Fundación Linux adjuntó ONOS como una colaboración. [18]

El controlador ONOS ofrece:

- Escalabilidad.
- Alto rendimiento.
- Resistencia.
- Compatibilidad con los dispositivos.

Se ha empleado el controlador SDN ONOS, el cual es de código abierto, como controlador de la red cableada. Está diseñado para que sea estable, escalable y distribuido con una orientación a las redes de proveedor de servicio [19] [20]. Para la correcta instalación se deben de seguir los pasos que se indican en su página web oficial. [21]

El concepto de la flexibilidad se debe al mecanismo que incorpora para permitir la conexión y desconexión de dispositivos mientras se encuentra en funcionamiento. También tiene disponibles fuentes de telemetría como Grafana.

La arquitectura de ONOS está compuesta por 3 niveles:

- Nivel 1: protocolos de comunicación, la capa más baja de la figura.
- Nivel 2: núcleo de ONOS, las capas centrales de la figura.
- Nivel 3: aplicaciones, capa superior de la figura.

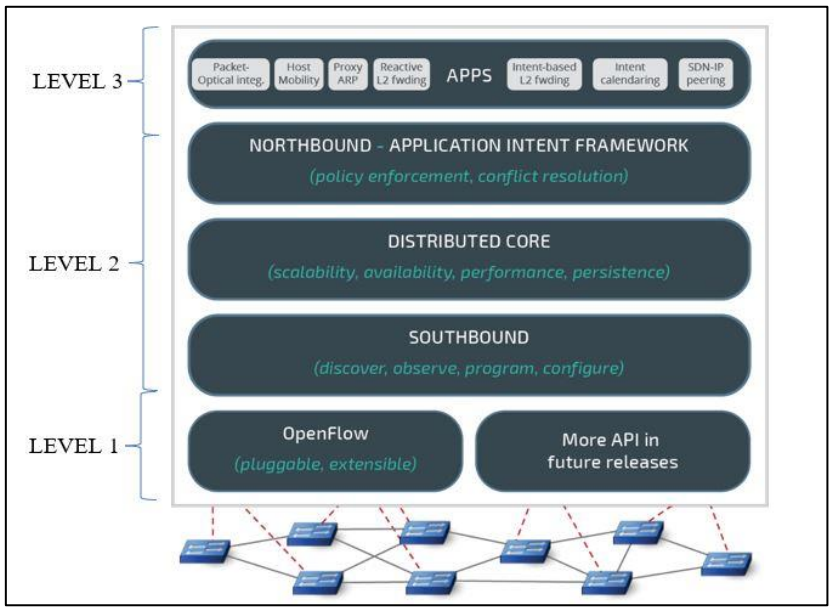


Figura 12. Arquitectura ONOS.

En la figura se puede observar de forma más detallada los tres niveles de ONOS mencionados. Por todas estas características y funcionalidades se ha escogido ONOS como controlador de la red cableada para este trabajo.

6.2 SDN-WISE

En el trabajo se va a emplear SDN-WISE, es una solución pensada para integrar las WSN en las SDN, por ello se ha escogido su uso en este trabajo. Está disponible online y es de libre acceso. [22]

Las redes SDN aportan grandes beneficios a las WSN como la versatilidad, una gestión más ágil y la flexibilidad que aporta. Todo ello es importante en las WSN, ya que permite programar los flujos desde el controlador. SDN-WISE ayuda a la transición entre ambas redes, ya que el empleo de OpenFlow no está pensado para redes inalámbricas y requiere una gran cantidad de modificaciones. Esta herramienta permite reducir el tráfico de control y así poder garantizar un ancho de banda, un parámetro crítico en las WSN llegando a reducir el rendimiento de la red.

La arquitectura está planteada tanto para el plano de control, como para los s*ink* y los nodos, la cual se muestra a continuación. La diferencia entre los nodos y el s*ink* es principalmente que los paquetes de control deben de llegar al controlador a través de un s*ink* para poder abandonar la red inalámbrica.

Como se muestra en la siguiente figura, el *sink* está formado por una capa de adaptación que será la encargada de realizar la transición de la red cableada a la inalámbrica. En el trabajo el *sink* está formado por una Raspberry que va conectada al switch, encargada de realizar la parte de adaptación, y por un nodo conectado a dicha Raspberry. Por otro lado, el plano de control se encuentra ubicado en el ordenador como segundo punto de control de la red inalámbrica SDN.

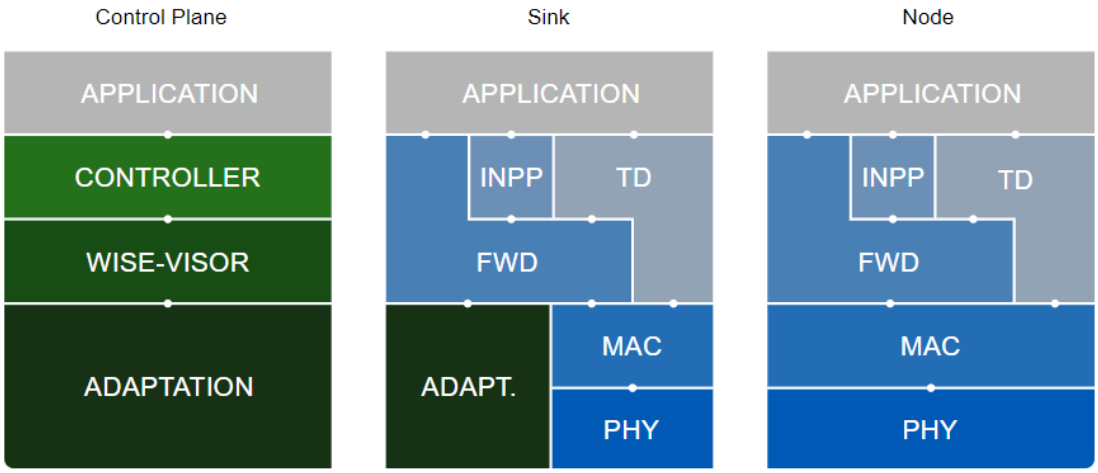


Figura 13. Arquitectura SDN-WISE. [22]

Entrando más en profundidad, SDN-WISE está formado principalmente por 5 capas:

- PHY (802.15.4 estandarizado).
- MAC (802.15.4e estandarizado).
- FWD (*Forwarding*): se reenvían los paquetes obedeciendo la tabla de flujo enviada desde el controlador. Para realizar dicha acción se debe de tener en cuenta el ID (identificador) del nodo, el cual tiene una longitud de 2 bytes.
- TD (*Topology Discovery*): tiene la función de descubrir los nodos vecinos e informar al controlador. Esta acción se lleva a cabo debido a que los nodos transmiten paquetes de *beacon* de forma periódica indicando el rango de cobertura. Cuando un nodo vecino recibe dicho paquete lo añade a la tabla de vecinos. A continuación, los nodos informan

al controlador mediante un paquete de *report* para que pueda construir la topología y tener una visión de la red.

- INNP (*In-Network Packet Processing*): esta capa se encarga del procesamiento de los paquetes dentro de la red, juntando aquellos paquetes de tamaño reducido con un mismo destino para reducir la congestión. El tamaño máximo de los paquetes es de 116 bytes.
- Aplicación.

Los paquetes SDN-WISE tienen un encabezado de 10 bytes común formado por: 1 byte para indicar la longitud del paquete, 1 byte para identificar la red, 2 bytes para indicar el nodo origen, 2 bytes para indicar el nodo destino, 1 byte para indicar el tipo de paquete, 1 byte para indicar el tiempo de vida TTL (*Time To Live*) y 2 bytes para identificar el nodo del siguiente salto.

Como se ha indicado, hay un byte exclusivo para identificar el tipo de paquete. Existen 8 tipos de paquete:

- Paquete de tipo 0: paquete de datos.
- Paquete de tipo 1: paquete de *beacon*.
- Paquete de tipo 2: paquete de *report*.
- Paquete de tipo 3: paquete de *request*.
- Paquete de tipo 4: paquete de *response*.
- Paquete de tipo 5: paquete de *OpenPath*.
- Paquete de tipo 6: paquete de configuración.
- Paquete de tipo 7: paquete de *RegProxy*.

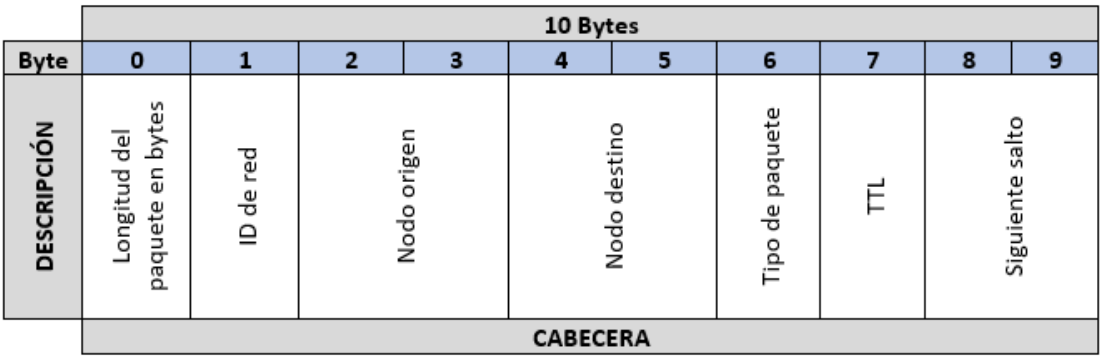


Figura 14. Cabecera paquete SDN-WISE.

En el trabajo se han realizado varias modificaciones a dicha herramienta para poder ofrecer varios servicios que no van incluidos, más adelante se comentarán.

6.3 Mininet

Mininet es una herramienta que permite una emulación de una red al completo. Esta herramienta incorpora varios ejemplos de redes, pero también permite personalizar una red, y aportar un respaldo, ya que es pública con licencia de código abierto. [23] [24]

La herramienta es muy útil ya que permite realizar tanto pruebas y demostraciones de la red, sobre todo para redes SDN empleando OpenFlow. Mininet ofrece las siguientes características:

- Permite realizar pruebas de red mediante un banco de pruebas de forma económica y sencilla.
- Permite que varios usuarios trabajen a la vez de forma simultánea.
- Permite realizar pruebas de regresión a un nivel de sistema.
- Se pueden realizar diseños y pruebas de topologías complejas personalizadas.

- Proporciona una API de Python para la creación de las redes.

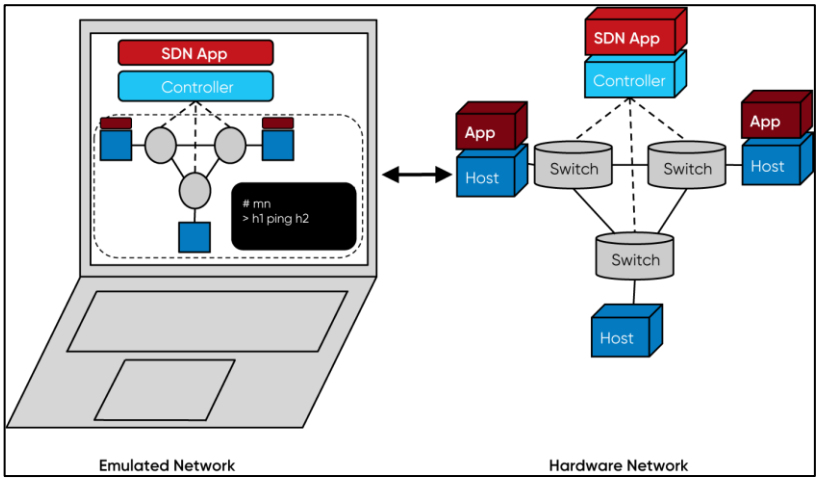


Figura 15. Herramienta Mininet. [25]

Mediante esta herramienta se puede realizar una comprobación del comportamiento de la red que se quiere desarrollar y experimentar. No todo son ventajas, también tiene algunas limitaciones:

- No se puede exceder la CPU o el ancho de banda disponible en un solo servidor.
- No permite ejecutar switches y aplicaciones OpenFlow no compatibles con Linux.

En el trabajo se va a emplear dicha herramienta para realizar las pruebas de la red cableada SDN por las ventajas que aporta.

6.4 Cooja

Cooja es una herramienta de simulación para RPL en WSN. Por este motivo se emplea en redes de sensores inalámbricos, como en el caso de este trabajo. Es de código abierto y de uso gratuito. [26] [27]

Mediante esta herramienta basada en Java se puede simular una red WSN con el sistema operativo de cada nodo. Al emplear el sistema operativo Contiki-NG en los nodos, este simulador es una buena elección. Cooja ofrece las siguientes ventajas:

- Flexible: permite sustituir y añadir funcionalidades.
- Uso gratuito.
- Permite simular redes WSN a varios niveles.
- Incorpora la herramienta *collect-view*: permite obtener información de la red y visualizarla de forma gráfica.

Dada esta breve introducción a la herramienta Cooja, se ha podido observar que se trata de una herramienta muy desarrollada para WSN y por ello se ha empleado en el trabajo.

6.5 Contiki-NG

Contiki es un sistema operativo que se emplea en pequeños controladores de bajo consumo. Permite desarrollar aplicaciones que permiten hacer un uso adecuado del hardware y a la vez permite una comunicación inalámbrica de bajo consumo. [28] [29]

A mediados de 2017 surgió una nueva variante del sistema operativo Contiki, la cual se llamó Contiki-NG. Esta nueva variante surgió con el objetivo de:

- Comunicación IPv6 confiable y segura.
- Realizar una modernización del sistema.

- Mejora de la documentación.
- Mayor agilidad.
- Emplearse en IoT.

Contiki-NG es un sistema operativo pensado para dispositivos IoT. Dados los objetivos anteriores, permite una comunicación de bajo consumo algo muy importante a tener en cuenta en dispositivos que se emplean en comunicaciones inalámbricas.

6.6 Grafana

En este trabajo también se ha empleado la herramienta Grafana, un software de libre disposición. Mediante esta herramienta se puede tanto consultar, visualizar y realizar un tratamiento de los datos recibidos de una fuente en los paneles de visualización. La gran utilidad de Grafana es que el tratamiento de los datos se puede realizar desde un único panel, juntando todas las gráficas, aunque provengan de varios destinos.

Alguna de las principales características que ofrece Grafana son las siguientes: [30]

- Visualización de los datos de forma más gráfica en paneles. Permite modificar los gráficos con multitud de opciones.
- Agrupación de los gráficos en paneles, pudiendo ver gráficos de distintos orígenes en un mismo panel.
- Existe una autenticación al iniciar la herramienta, por lo que ofrece cierta privacidad.
- Permite entre varios equipos intercambio de información, como pueden ser los datos o los paneles creados ya que se pueden exportar.

Para su correcta instalación se deben de seguir las indicaciones que se indican en su página web oficial. [31]

6.7 Dispositivos empleados

6.7.1 Switch

Para el desarrollo de este trabajo se han empleado dos switches HPE Aruba 2930F (JL259A) y se emplean para la red cableada SDN. Este dispositivo tiene un precio de 725 € por unidad.

Se ha escogido este switch ya que contiene 24 puertos, puertos suficientes para el escenario que se quiere desplegar. También soporta el protocolo OpenFlow.

Para el trabajo es necesario el empleo de 2 dispositivos que formarán parte de la red cableada y establecerán la conexión con el controlador ONOS, el punto de control ubicado en las Raspberrys y entre ambos switches.

6.7.2 Nodos

Los nodos que se van a emplear en la WSN son OpenMote B fabricados por *Industrial Shields*. Los dispositivos tienen un precio de 100,75 € por unidad.

Se han escogido estos dispositivos ya que el consumo es muy reducido, algo muy importante en dispositivos IoT. También soporta todas las modulaciones IEEE802.15.4g y es compatible con Contiki. Dispone de 4 indicadores LED y la posibilidad de conectar una antena de 2,4 GHz, que no va incluida. La antena escogida es Antena WiFi 2,4 GHz, cuyo precio es de 10,50 € por unidad.

6.7.3 Robot

Para la movilidad del nodo se emplea un robot, RoboMaster S1 del fabricante DJI. Este robot se va a programar para que siga una línea azul y el nodo móvil se ubicará encima del robot. El recio del robot es de 549 €.

Se ha escogido este robot ya que tiene una gran variedad de funciones gracias al reconocimiento de objetos y permite generar múltiples aplicaciones.

6.7.4 Raspberry Pi

Se van a emplear varias Raspberrys, en especial el modelo Raspberry Pi RPI4-MODBP 4GB-BULK, con un precio de 58,75 € por unidad.

Estos dispositivos son los que se conectan a los switches y se encargan de la adaptación de la red WSN con la SDN. También se emplean como segundo punto de control para la WSN.

6.7.5 Sensores

Se van a emplear los sensores de la placa OpenMote B Sensors con un precio de 36,24 € por unidad. Esta placa se emplea para los nodos OpenMote B e incluye sensores de humedad, temperatura, presión y luminosidad.

6.7.6 PLC

Se emplean dos PLCs de la marca Siemens, en especial el modelo PLC LOGO! 230RCE-Siemens, con un precio de 160 € por unidad.

Estos dispositivos cuentan con un puerto Ethernet, dato de interés, ya que se va a emplear para observar el estado de la red cableada. Permite realizar múltiples programas, lo que permitirá asignarle varias funciones a cada una de ellas.

6.7.7 Ordenador central

El ordenador central es el que dispondrá de los controladores. Para ello se emplea un ordenador con un procesador Intel Core i7-10700 CPU 3,80 GHz x 16. El sistema operativo que se emplea es Ubuntu 20.04.3 LTS y con una capacidad suficiente como para poder trabajar sin problemas.

Capítulo 7. Trabajo previo

7.1 Simulación del escenario base

Para tener un primer contacto con las redes definidas por software, se ha realizado una emulación de una topología similar a la planteada a través de Mininet y se ha empleado ONOS como controlador SDN.

Para ello, se ha creado un *script* en Python donde se establece una conexión con el controlador ONOS. Seguidamente se han creado los hosts, los switches con el protocolo OpenFlow v1.3 y los enlaces para la comunicación con una limitación de 1000 Mbits/sec.

Para el desarrollo del *script* se han empleado las siguientes funciones:

- Añadir el controlador ONOS SDN, indicando la dirección IP en la que se aloja y el puerto para establecer la conexión:

```
c0=net.addController('c0', controller=RemoteController, ip='127.0.0.1', port=6653)
```

- Añadir los switches, donde la x hace referencia al número de switch. Como se observa a continuación emplean el protocolo OpenFlow v1.3:

```
sx=net.addSwitch('sx', protocols="OpenFlow13")
```

- Añadir los hosts, dónde la x hace referencia al número de host. También se les asigna una dirección IP y una máscara de red:

```
hx=net.addHost('hx')
hx.setIP("XXX.XXX.XXX.XXX",YY)
```

- Añadir los enlaces entre los dispositivos, también se asigna el ancho de banda del enlace:

```
net.addLink(dispositivo1,dispositivo2,cls=TCLink,bw=1000)
```

Una vez configurado el *script* de Python, se ejecuta el controlador ONOS y para ello se emplea el siguiente comando:

```
sudo bazel run onos-local [-- debug]
```

También para poder visualizar la interfaz web de ONOS se debe ejecutar el siguiente comando:

```
sudo ./tools/test/bin/onos localhost
```

Por último, se ejecuta el *script* de Python para poder visualizar el escenario confeccionado en la interfaz web del controlador ONOS mediante el siguiente comando:

```
sudo mn --custom=/TopoSDN/escenario.py
```

Una vez todo ejecutado es el momento de comprobar que la configuración realizada en el controlador ONOS es la correcta, y por lo tanto se puede visualizar el escenario. Para ello se deben de activar las aplicaciones de ONOS pertinentes y se obtiene el siguiente escenario en pantalla.

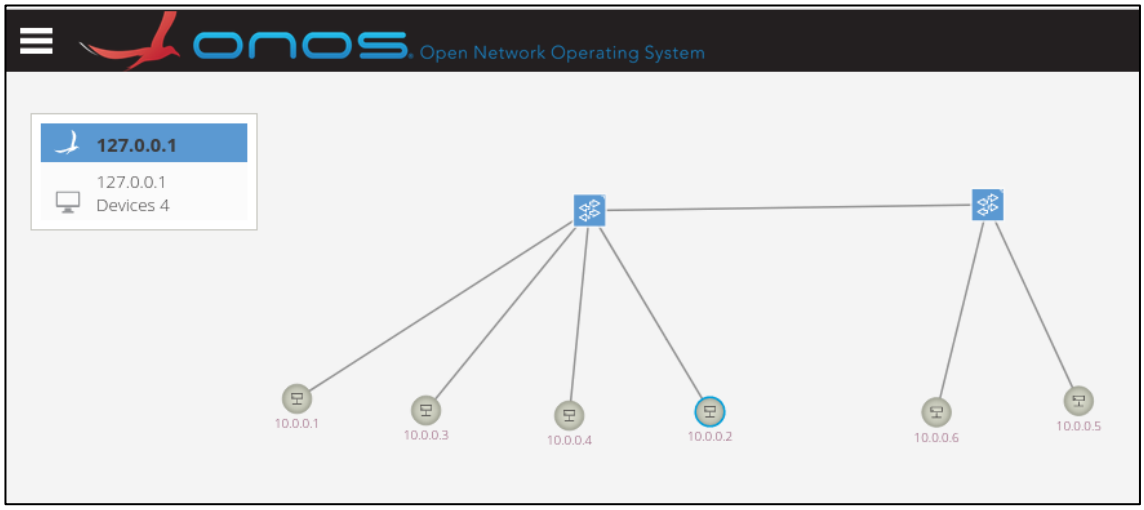


Figura 16. Escenario simulado.

Como se observa en la figura, se dispone de dos switches conectados al controlador y a su vez a los hosts correspondientes. Para verificar el correcto funcionamiento se realiza un *pingall* para verificar la comunicación entre los dispositivos.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
```

Figura 17. *Pingall* con el *Forwarding* activado.

A continuación, se va a estudiar el funcionamiento de los *intents*. En primer lugar, se debe desactivar la aplicación de *Reactive Forwarding*. Esta aplicación permite instalar los flujos en el plano de datos necesarios para cada paquete perdido que llega al controlador. De esta forma el tráfico se reenvía de manera correcta y se puede hacer *ping* como se ha observado anteriormente. Al desactivarla, todos los flujos instalados desaparecen y se debe de crear manualmente los flujos de interés. Como se observa, ahora ya no se reciben los paquetes al realizar un *pingall*.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X X X X
*** Results: 100% dropped (0/30 received)
```

Figura 18. *Pingall* con el *Forwarding* desactivado.

Ahora se van a crear los flujos manualmente mediante *intents*. Los *intents* permiten especificar las voluntades de control de red como políticas, en lugar de mecanismos. Esta acción se realiza mediante un proceso de instalación de los *intents*, las políticas, proporcionado las reglas de flujo necesarias, la reserva de recursos necesarios y los cambios pertinentes en el entorno de trabajo. Los *intents* se pueden realizar mediante la interfaz web del controlador ONOS o mediante los siguientes comandos:

add-host-intent idHostOrigen idHostDestino (*intent* entre hosts)
add-point-intent idLinkOrigen idLinkDestino (*intent* entre links)

En este caso se van a realizar de forma gráfica, se puede visualizar que el *intent* se ha instalado de forma correcta entre el host 1 y el 6.

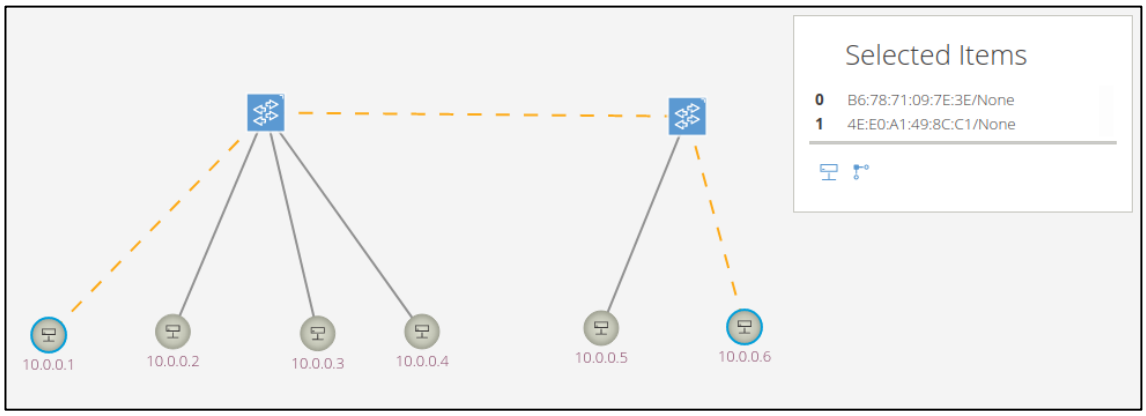


Figura 19. *Intent* entre el dispositivo 1 y el 6.

El *intent* se ha instalado correctamente como se muestra en las siguientes figuras, tanto en el interfaz web como en la pantalla de comandos.

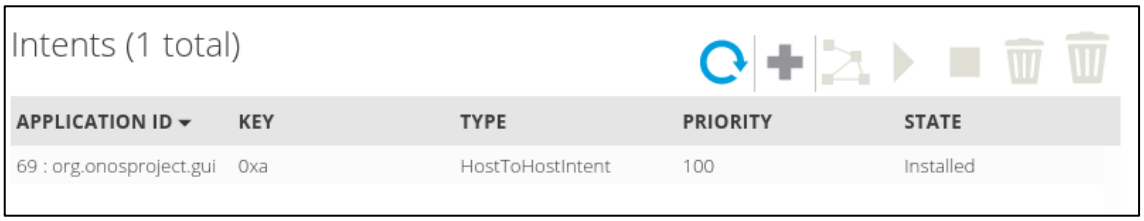


Figura 20. Instalación del *intent* en la interfaz web.

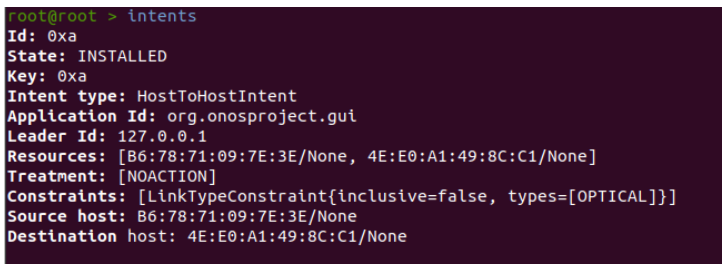


Figura 21. Instalación del *intent* en la pantalla de comandos.

Ahora al realizar un *pingall* se puede observar como si se establece una comunicación con el dispositivo 1 y 6 al tener un *intent* entre ellos. El controlador, además, tiene la capacidad de que en caso de que se caiga algún enlace del *intent* busque si existe un camino alternativo para que no se pierda la comunicación.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X X h6
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X X X
h5 -> X X X X X
h6 -> h1 X X X X
*** Results: 93% dropped (2/30 received)
```

Figura 22. *Pingall* con el *intent* instalado.

En el siguiente capítulo se van a desplegar las redes físicamente, en primer lugar la red cableada y a continuación las redes inalámbricas con SDN-WISE.

7.2 Configuración de los switches

A continuación, se realiza la configuración de los dos switches físicos que se van a emplear en el escenario. Para ello se debe de tener presente que deben de ser compatibles con el protocolo OpenFlow, más en concreto la versión 1.3.

Para realizar la configuración de los switches se debe de crear en primer lugar la configuración del protocolo OpenFlow. Se debe indicar la ubicación del controlador indicando la dirección IP, el puerto y la VLAN de la interfaz del controlador. También se deberá de crear dos VLAN, ya que, como se ha comentado en el estado del arte de SDN, existe una separación del plano de datos y el plano de control. Para ello se asignan unos puertos a la VLAN 1 y otros a la VLAN 2. Hay un puerto que debe ser común, ya que el segundo switch se conectará al primero a dicho puerto para mostrar la configuración de la red en el controlador ONOS. A continuación, se puede visualizar la configuración de los switches.

```
Running configuration:

; JL259A Configuration Editor; Created on release #WC.16.11.0002
; Ver #14:67.6f.f8.1d.9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:44
hostname "Aruba-2930F-24G-4SFP"
module 1 type jl259a
sflow 1 destination 192.168.1.5
sflow 1 polling 1-28 20
sflow 1 sampling 1-28 50
snmp-server community "public" unrestricted
openflow
   controller-id 1 ip 192.168.10.5 port 6653 controller-interface vlan 10
   instance "ofsdn"
      member vlan 20
      controller-id 1
      software-flow-table 4
      version 1.3 only
      connection-interruption-mode fail-standalone
      pipeline-model standard-match
      enable
      exit
   enable
   exit
vlan 1
   name "DEFAULT_VLAN"
   no untagged 1-28
   no ip address
   exit
vlan 10
   name "VLAN10"
   untagged 4-22,24-28
   tagged 23
   ip address 192.168.10.200 255.255.255.0
   exit
vlan 20
   name "VLAN20"
   untagged 1-3
   tagged 23
   ip address 10.0.0.200 255.255.255.0
   exit
```

Figura 23. Configuración switch 1.

```
Running configuration:

; JL259A Configuration Editor; Created on release #WC.16.09.0001
; Ver #14:27.6f.f8.1d.9b.3f.bf.bb.ef.7c.59.fc.6b.fb.9f.fc.ff.ff.37.ef:04
hostname "Aruba-2930F-24G-4SFP"
module 1 type jl259a
sflow 1 destination 192.168.1.5
sflow 1 polling 1-28 20
sflow 1 sampling 1-28 50
snmp-server community "public" unrestricted
openflow
   controller-id 1 ip 192.168.1.5 port 6653 controller-interface vlan 1
   instance "ofsdn"
      member vlan 2
      controller-id 1
      version 1.3
      connection-interruption-mode fail-standalone
      pipeline-model standard-match
      enable
      exit
   enable
   exit
vlan 1
   name "DEFAULT_VLAN"
   no untagged 1-3
   untagged 4-23,25-28
   tagged 24
   ip address 192.168.1.252 255.255.255.0
   ipv6 enable
   ipv6 address dhcp full
   exit
vlan 2
   name "VLAN2"
   untagged 1-3
   tagged 24
   ip address 10.0.0.199 255.255.255.0
   exit
```

Figura 24. Configuración switch 2.

En las figuras anteriores se muestran las dos VLAN creadas, estas VLAN son distintas para cada switch como se puede observar, pero las estructuras de configuración son las mismas.

Una vez realizada la configuración de ambos switches y realizada su correcta conexión entre ellos y el controlador ONOS, se debe modificar el *script* de Python de la simulación en Mininet. En él, se añaden las interfaces por los que se han conectado los switches al controlador (switch 1 por la VLAN 20 y switch 2 por la VLAN 2) y realizar una conexión a uno de los switches simulados. Para ello se hace uso de la siguiente función:

```
_intf1 = Intf("interfaz", node=sx)
```

En este punto ya se tiene la configuración de los switches y se han añadido a la simulación. Por la parte del controlador ONOS, se deben de activar las aplicaciones pertinentes para que el switch pueda conectarse al controlador. Otra acción a tener en cuenta es la configuración del número de tablas para una tabla de flujo, ya que si no los flujos que se agreguen ocasionarán problemas.

7.3 Configuración de las PLCs

En el proyecto se van a emplear dos PLCs, se les debe configurar una IP para conectarlas a las VLAN correspondientes de cada switch. Mediante el programa LOGO! se van a programar las PLCs para que una de ellas envíe un pulso a la otra.

- PLC 1: se va a configurar con la IP 10.0.0.201 y máscara 255.255.255.0 e irá conectada al switch 1. El programa que se ha configurado es el que se muestra en la siguiente imagen.

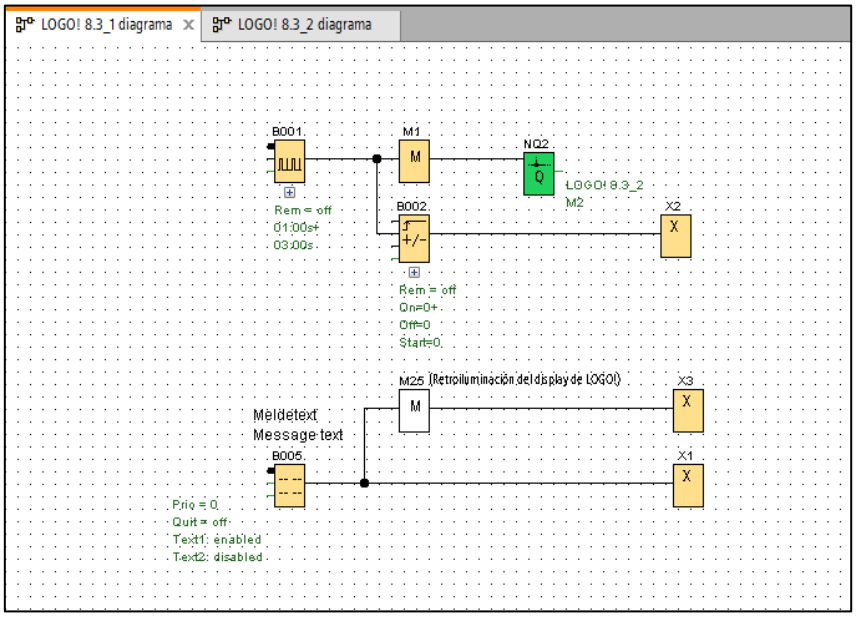


Figura 25. Configuración PLC 1.

El programa genera pulsos cada 3 segundos, una vez generado el pulso lo manda a la otra PLC. Mientras por pantalla muestra el siguiente mensaje, donde se indican también los pulsos enviados.

Figura 26. Mensaje PLC 1.

- PLC2: se va a configurar con la IP 10.0.0.202 y máscara 255.255.255.0 e irá conectada al switch 2. El programa que se ha configurado es el que se muestra en la siguiente imagen.

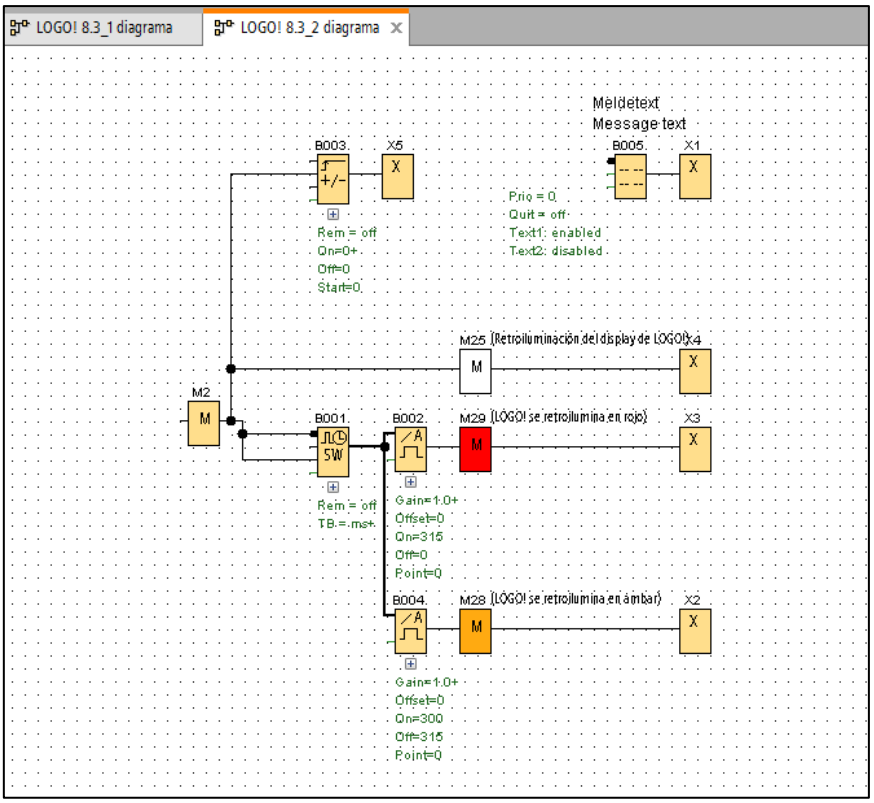


Figura 27. Configuración PLC 2.

El programa cada vez que reciba un pulso iluminará la pantalla de color blanco e incrementará el contador de pulsos recibidos. Cuando está a la espera del pulso inicia un contador que muestra el tiempo entre paquetes, si no ha recibido un pulso en el periodo indicado la pantalla se iluminará de color naranja. Si el contador sigue incrementando y supera los 3,15 segundos la pantalla se iluminará de color rojo. Este proceso se detendrá cuando reciba un pulso y reinicie el contador. A continuación, se muestran los estados de transición, en la figura a) cuando se encuentra a la espera del pulso (durante los 3 primeros segundos). En la figura b) cuando recibe un pulso, en la figura c) cuando el contador se encuentra entre los 3 y los 3,15 segundos y en la figura d) cuando el contador supera los 3,15 segundos.

Figura 28. Transiciones PLC 2: a) a la espera del pulso durante los 3 primeros segundos, b) al recibir el pulso, c) contador entre los 3 y 3,15 segundos, d) contador por encima de los 3,15 segundos.

7.4 Configuración de los nodos

Los nodos emplean el sistema operativo Contiki-NG. Se va a emplear la aplicación disponible de código abierto de SDN-WISE-CONTIKI que permite la transición entre redes SDN e IWSN mediante el empleo del protocolo SDN-WISE. También se han añadido nuevas funcionalidades, como un indicador en los leds del dispositivo para indicar la cobertura, añadir los datos de los sensores en el paquete de datos y añadir el protocolo TSCH para ofrecer una mayor calidad de servicio.

7.5 Configuración del controlador SDN-WISE.

El controlador SDN-WISE es de código abierto e incorpora varias funciones básicas. Para ampliar dichas funciones se han añadido nuevas aplicaciones, como visualizar y distinguir los nodos de forma gráfica o añadir flujos entre los dispositivos inalámbricos. A continuación, se nombran las nuevas funcionalidades que se han incorporado:

- Se ha incorporado una nueva pestaña de control, permite configurar un flujo entre dos dispositivos, asignando una prioridad y un *deadline*. También permite eliminar flujos existentes y visualizarlos en una tabla.

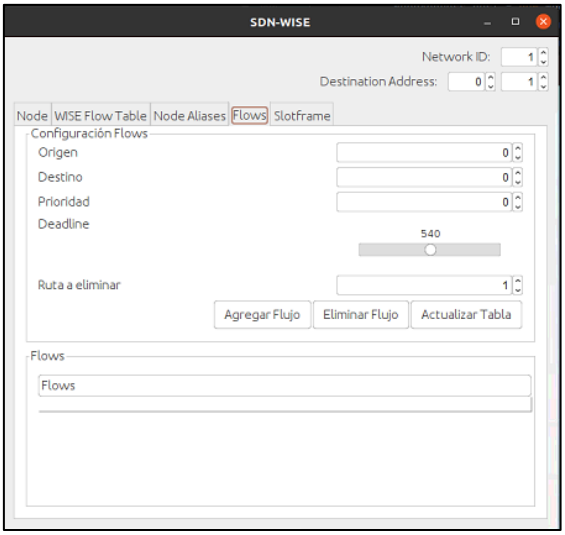


Figura 29. Configuración pestaña *Flows*.

- Se ha incorporado una nueva pestaña, la cual permite visualizar el *slotframe*, la matriz de planificación de TSCH. En el trabajo se van a emplear 67 *timeslots*, reservando los 6 primeros para el tráfico de control SDN, y 4 *Channle Offsets*, ya que si se emplea un mayor número es necesario un mayor tiempo de sincronismo.

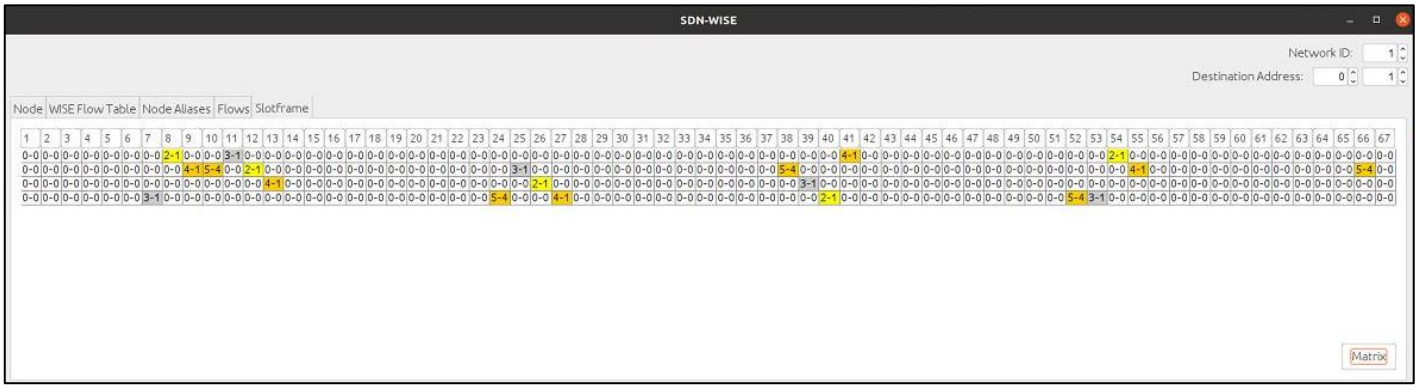


Figura 30. Configuración pestaña *Slotframe*.

- Creación de una aplicación que envíe las estadísticas a Grafana mediante un cliente MQTT. Se envían varias estadísticas, como el PDR, el DSR y la información de los sensores. En la siguiente figura se puede visualizar uno de los paneles configurados.

Figura 31. Panel de visualización de la herramienta Grafana.

- Modificación de la interfaz gráfica, cómo distinguir los nodos fijos (de color verde) el *sink* (de color gris) y el nodo móvil (de color azul).

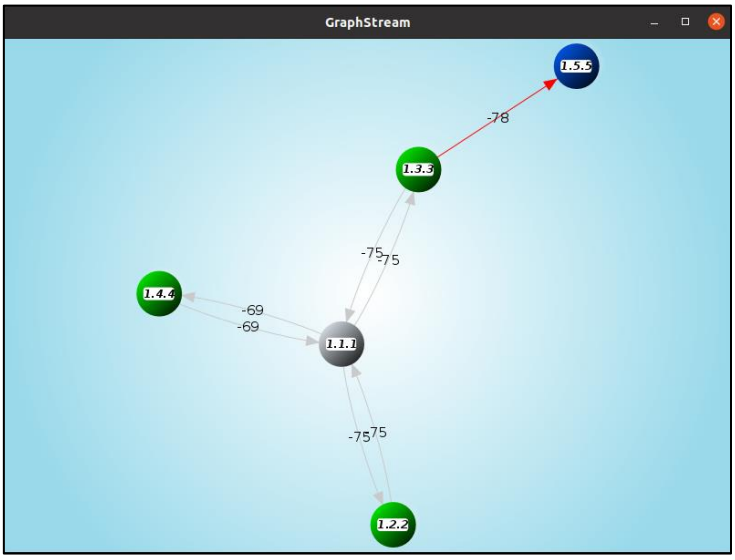


Figura 32. Interfaz gráfica controlador SDN-WISE.

Estas nuevas funciones añadidas permiten tener un segundo punto de control para verificar que la red funciona correctamente y poder gestionarla.

Capítulo 8. Desarrollo

8.1 Red cableada SDN

Para realizar el montaje de la red cableada SDN se han configurado dos switches como se ha explicado en el capítulo anterior. Cada uno de ellos tiene conectado una PLC para observar la saturación de la red con la configuración descrita en el capítulo anterior, una Raspberry para añadir tráfico de saturación y otra Raspberry conectada a un *sink* de la SDN-WISE. En la siguiente figura se puede observar las conexiones a uno de los switches.

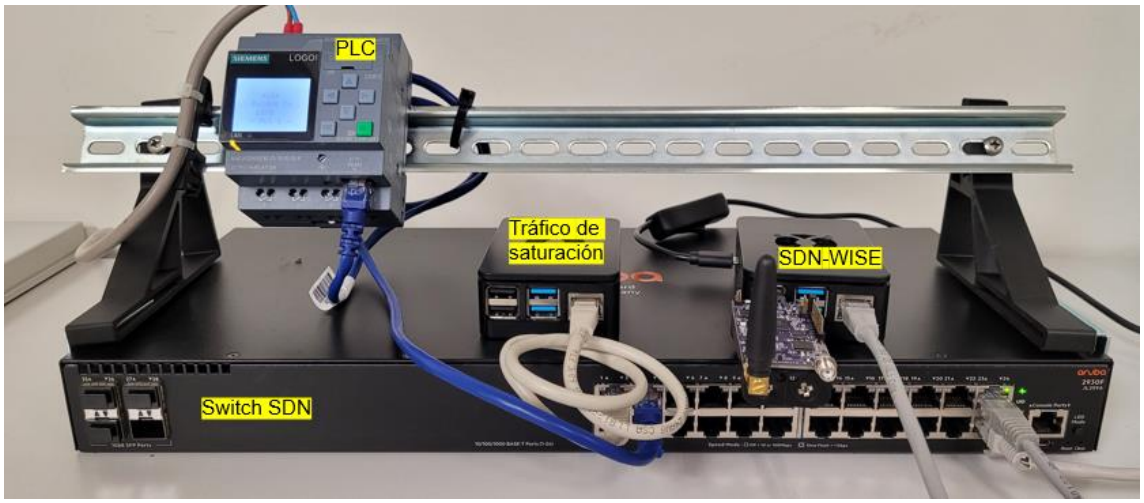


Figura 33. Conexiones del switch.

Para realizar las pruebas con el controlador ONOS, también se ha añadido la red cableada simulada con Mininet. A continuación, se puede observar la topología descrita.

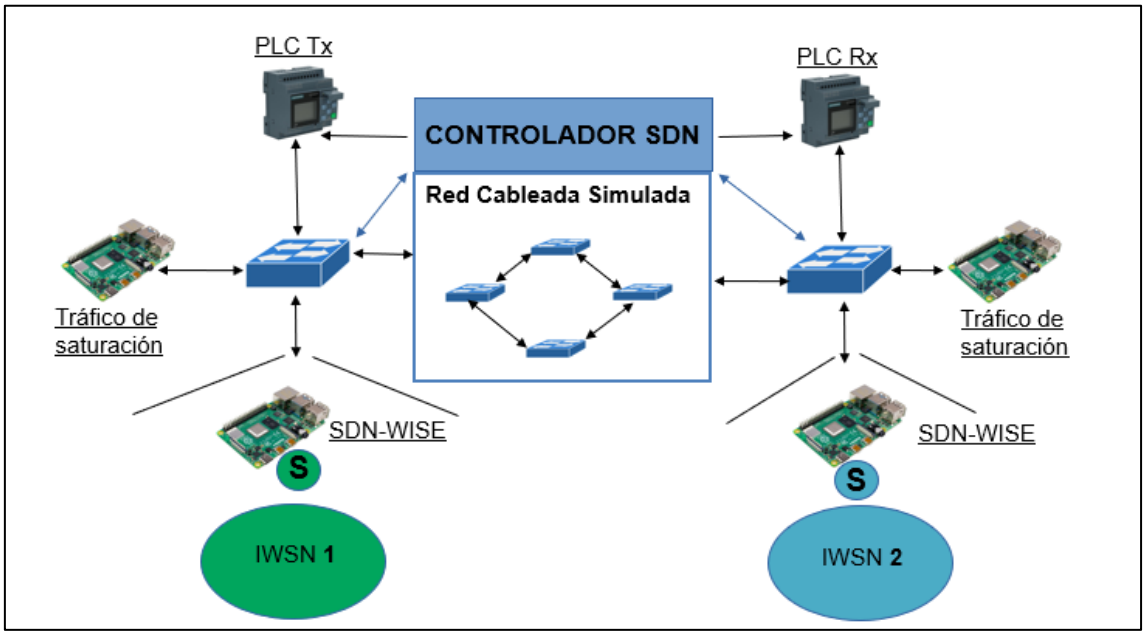


Figura 34. Escenario de la red cableada.

8.1.1 Sin calidad de servicio

Una vez descrito el escenario de la red cableada SDN se procede a su evaluación. Para ello se tienen dos PLC, en las que se ha decidido modificar el tiempo de envío de los pulsos a 1,02 segundos.

Mediante el controlador ONOS se desactiva la aplicación de *Reactive Forwarding* y se añaden dos *intents*, uno entre las dos Raspberrys de saturación y otro entre las dos PLCs. En la siguiente figura se puede observar el escenario en el controlador ONOS.

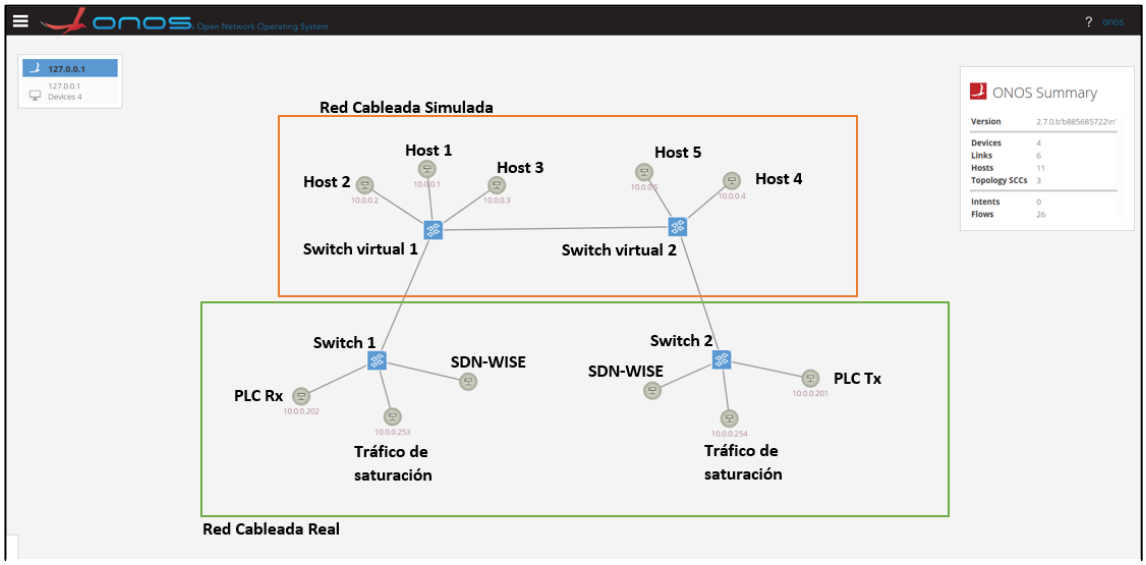
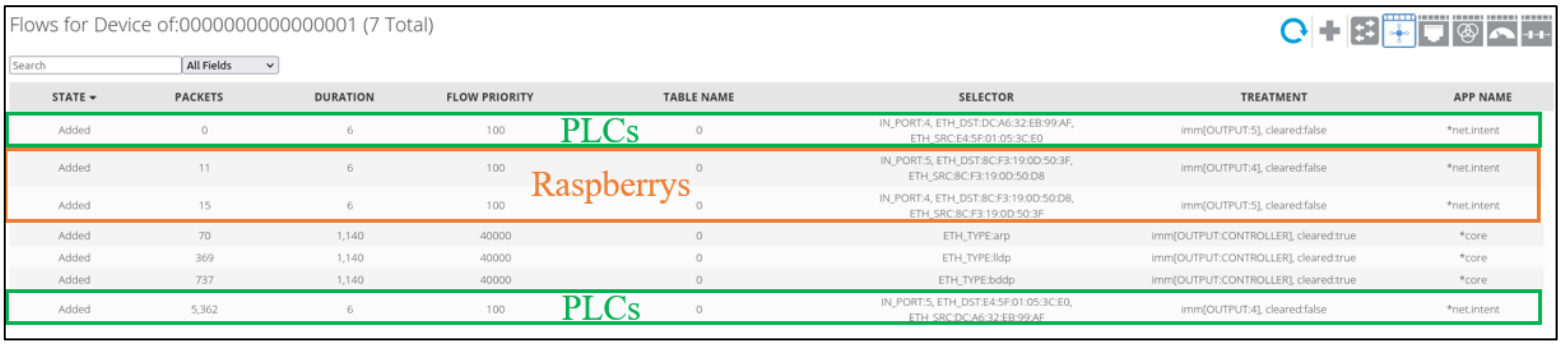


Figura 35. Escenario controlador ONOS.

En este momento, los flujos instalados en el switch virtual 1 son los siguientes, los del cuadro naranja son flujos que se han creado para los *intents* entre las Raspberrys y los flujos de los cuadros verde para los *intents* de las PLCs. Se observa como las prioridades son iguales para los 4 flujos, una prioridad de 100.



STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT	APP NAME
Added	0	6	100	0	IN_PORT:4, ETH_DST:DC:A6:32:EB:99:AF, ETH_SRC:E4:5F:01:05:3C:E0	imm[OUTPUT:5], cleared:false	*net.intent
Added	11	6	100	0	IN_PORT:5, ETH_DST:8C:F3:19:0D:50:3F, ETH_SRC:8C:F3:19:0D:50:D8	imm[OUTPUT:4], cleared:false	*net.intent
Added	15	6	100	0	IN_PORT:4, ETH_DST:8C:F3:19:0D:50:D8, ETH_SRC:8C:F3:19:0D:50:3F	imm[OUTPUT:5], cleared:false	*net.intent
Added	70	1,140	40000	0	ETH_TYPE:arp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	369	1,140	40000	0	ETH_TYPE:lldp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	737	1,140	40000	0	ETH_TYPE:bddp	imm[OUTPUT:CONTROLLER], cleared:true	*core
Added	5,362	6	100	0	IN_PORT:5, ETH_DST:E4:5F:01:05:3C:E0, ETH_SRC:DC:A6:32:EB:99:AF	imm[OUTPUT:4], cleared:false	*net.intent

Figura 36. Flujos switch virtual 1 sin QoS.

Para comprobar el estado de la red y observar la saturación de la red y su rendimiento se va a emplear la herramienta *iperf*, con ella se pueden crear flujos de datos TCP y UDP entre un cliente y un servidor. Se va a configurar una de las Raspberrys como servidor y la otra como cliente. La Raspberry que actúa como cliente se le va a indicar que el ancho de banda de la comunicación con el servidor sea de 10 Mbits/s y que tenga una duración de 1 minuto y 40 segundos la comunicación.

Al finalizar la comunicación entre las Raspberrys se obtienen los siguientes datos, se han transferido un total de 125 MBytes pero solo se han recibido 110 MBytes y el ancho de banda se ha respetado en todo momento. Esto implica que el 88% de los datos se han recibido y el 12% se han perdido.

En el caso de la comunicación entre las PLCs, esta ha sido afectada por la comunicación de las Raspberrys al ser muy superior. De los 54 pulsos que ha transmitido la PLC 1 durante el periodo que ha durado la comunicación entre las Raspberrys, solo se han recibido 38 pulsos en la PLC 2. Esto implica que el 70,37% de los datos se han recibido y el 29,63% se han perdido.

Dados estos resultados se puede obtener el PDR (*Packet Delivery Ratio*) que se mide teniendo en cuenta la relación entre los paquetes enviados y los paquetes recibidos entre dos dispositivos. En el caso de las PLCs el PDR obtenido es del 70,37%. También se puede obtener el DSR (*Deadline Satisfaction Ratio*) que se mide teniendo en cuenta la siguiente fórmula.

$$DSR = \frac{PaquetesRecibidos - PaquetesRecibidos\ fuera\ de\ tiempo}{PaquetesRecibidos} \quad (2)$$

En este caso se obtiene un DSR de 65,79%, ya que de los 38 paquetes recibidos hay 13 que han llegado fuera del tiempo establecido.

8.1.2 Con calidad de servicio por defecto

Dado el escenario del punto anterior, se agrega una cola al switch virtual 1. Dicha cola se llama cola 0 y se configura para todos los paquetes salientes al switch virtual 2, delimitándolos con un ancho de banda máximo de 7 Mbits/s.

A continuación, se procede a realizar la misma prueba entre las dos Raspberrys, una será el servidor y la otra el cliente a la cual se le indica que el ancho de banda de la comunicación con el servidor sea de 10 Mbits/s y que tenga una duración de 1 minuto y 40 segundos la comunicación.

Al finalizar la comunicación entre las Raspberrys se obtienen los siguientes datos, se han transferido un total de 125 MBytes pero solo se han recibido 71,7 MBytes. El ancho de banda se ha visto afectado ya que este ha sido de 5,91 Mbits/s, y no de 10 Mbits/s como se le había indicado al cliente. Esto implica que el 57,36% de los datos se han recibido y el 42,64% se han perdido. Como se puede observar, la limitación de la cola ha afectado a la comunicación.

En el caso de la comunicación entre las PLCs, esta ha sido afectada por la comunicación de las Raspberrys al ser muy superior, ya que ambos tráficos tenían designados la misma cola por defecto. De los 54 pulsos que ha transmitido la PLC 1 durante el periodo que ha durado la comunicación entre las Raspberrys, solo se han recibido 13 pulsos en la PLC 2. Esto implica que el 24,07% de los datos se han recibido y el 75,93% se han perdido.

Si se analiza el PDR de las PLCs, este ahora toma un valor inferior al caso anterior, siendo de 24,07%. En el caso del DSR se obtiene un valor de 46,15%, ya que de los 13 paquetes recibidos hay 7 que han llegado fuera del tiempo establecido.

Como se puede observar, en este caso, aunque se haya instalado una cola por defecto no ofrece ninguna calidad de servicio. Esto se debe a que al no tener más colas y asignar un tráfico a cada cola, todos los tráficos tienen asignados la cola 0 por defecto. Esto provoca que se delimiten más las comunicaciones porque se está asignado un ancho de banda más reducido y esto conlleva un mayor número de pérdidas.

8.1.3 Asignación de colas en las Raspberrys

En este caso, se parte del escenario anterior, pero se ha añadido una cola nueva la cual se llamará cola 1 y delimita el tráfico a un ancho de banda de 3 Mbits/s. En el controlador se va a instalar una nueva regla de fujo entre las Raspberrys. Esta nueva regla tendrá una prioridad de 130, ya que los *intents* tienen una prioridad de 100 y se quiere que las colas tengan una mayor prioridad. Ahora el switch virtual 1 cuenta con los siguientes flujos, destacando el del cuadro rojo que es el de la cola de las Raspberrys.

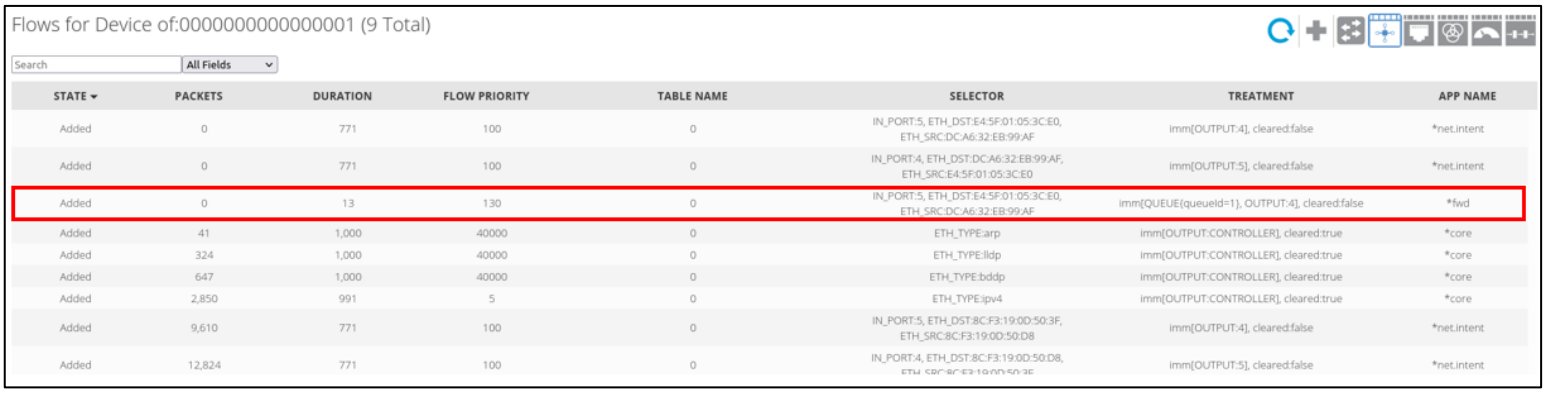


Figura 37. Flujos del switch virtual 1 con cola asignada.

A continuación, se va a realizar la misma prueba entre las Raspberrys, una será el servidor y la otra el cliente a la cual se le indica que el ancho de banda de la comunicación con el servidor sea de 10 Mbits/s y que tenga una duración de 1 minuto y 40 segundos la comunicación.

Al finalizar la comunicación entre las Raspberrys se obtienen los siguientes datos, se han transferido un total de 125 MBytes pero solo se han recibido 36,2 MBytes. El ancho de banda se ha visto afectado ya que este ha sido de 2,92 Mbits/s, y no de 10 Mbits/s como se le había indicado al cliente, debido a la limitación de la cola 1. Esto implica que el 28,96% de los datos se han recibido y el 71,04% se han perdido. Como se puede observar, la limitación de la cola ha afectado a la comunicación de forma muy agresiva.

En el caso de la comunicación entre las PLCs, esta no ha sido afectada por la comunicación de las Raspberrys al estar en la cola 0 por defecto, y por lo tanto existe una separación de tráficos. De los 54 pulsos que ha transmitido la PLC 1 durante el periodo que ha durado la comunicación entre las Raspberrys, se han recibido los 54 pulsos en la PLC 2. Esto implica que el 100% de los datos se han recibido y no se han producido pérdidas.

Si se analiza el PDR de las PLCs, este ahora toma un valor de 100%. En el caso del DSR se obtiene un valor del 100%, ya que la comunicación entre las PLCs tiene un ancho de banda suficiente para que no se vea afectada la comunicación.

8.1.4 Asignación de colas en las PLCs

En este punto se va a eliminar la asignación de la cola 1 a las Raspberrys y se va a eliminar el flujo pertinente en el switch virtual 1. Ahora se va a asignar la cola 1 a la comunicación entre las PLCs y se va a añadir el flujo pertinente en el controlador ONOS con una prioridad de 130.

A continuación, se va a realizar la misma prueba entre las Raspberrys, una será el servidor y la otra el cliente a la cual se le indica que el ancho de banda de la comunicación con el servidor sea de 10 Mbits/s y que tenga una duración de 1 minuto y 40 segundos la comunicación.

Al finalizar la comunicación entre las Raspberrys se obtienen los siguientes datos, se han transferido un total de 125 MBytes pero solo se han recibido 82,5 MBytes. El ancho de banda se ha visto afectado ya que este ha sido de 6,8 Mbits/s, y no de 10 Mbits/s como se le había indicado al cliente, debido a la limitación de la cola 0 donde se encuentran por defecto. Esto implica que el 66% de los datos se han recibido y el 34% se han perdido. Como se puede observar, en este caso al estar solo la comunicación entre las Raspberrys en la cola 0 se han producido menos pérdidas que en caso de estar ambas comunicaciones en la cola 0. Aun así, se han producido algunas pérdidas ya que el ancho de banda se ha limitado.

En el caso de la comunicación entre las PLCs, esta no ha sido afectada por la comunicación de las Raspberrys al estar en la cola 1 y tener ancho de banda suficiente. De los 54 pulsos que ha transmitido la PLC 1 durante el periodo que ha durado la comunicación entre las

Raspberrys, se han recibido los 54 pulsos en la PLC 2. Esto implica que el 100% de los datos se han recibido y no se han producido pérdidas.

Si se analiza el PDR de las PLCs, este ahora toma un valor de 100%. En el caso del DSR se obtiene un valor del 100%, ya que la comunicación entre las PLCs tiene un ancho de banda más que suficiente para que no se vea afectada la comunicación.

En la siguiente figura se puede observar de forma resumida los paquetes enviados y recibidos en las PLCs en las 4 pruebas realizadas.

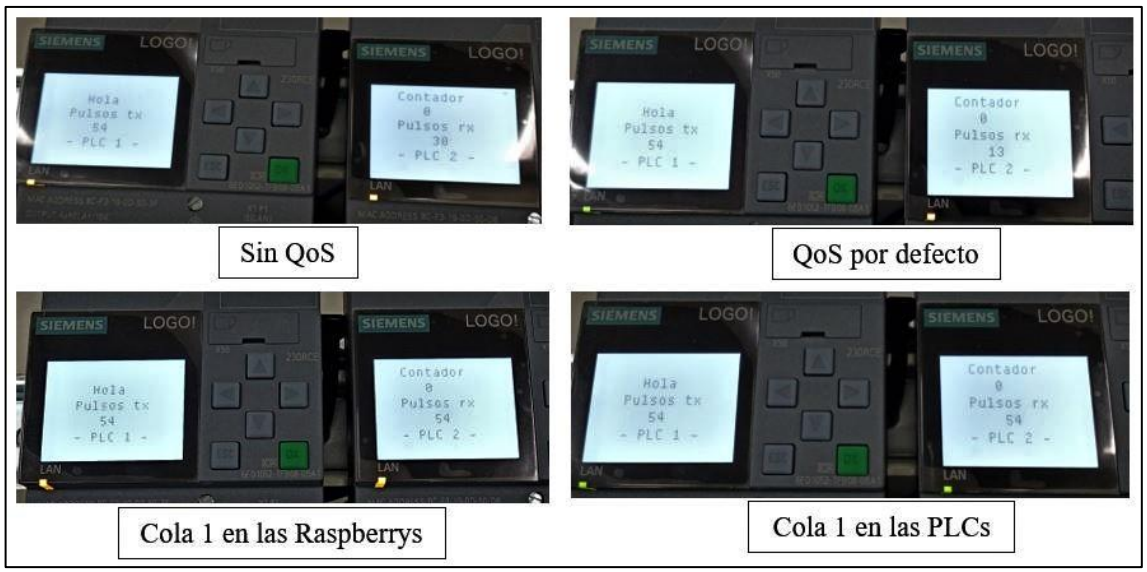


Figura 38. Paquetes transmitidos y recibidos en las PLCs.

8.2 Red inalámbrica SDN-WISE

En esta parte del proyecto se ha empleado SDN-WISE para realizar la integración de las IWSN en la SDN. Como se ha comentado anteriormente, en la arquitectura SDN-WISE existen 3 bloques. En esta parte del trabajo el bloque de control se ubica en la Raspberry, quien también dispondrá del *sink*. La Raspberry permite tener un segundo punto de control en la red SDN, mediante el controlador SDN-WISE. Los nodos se han configurado previamente como se ha comentado. En esta parte el escenario que se va a llevar a cabo es el que se muestra en la siguiente figura, incluyendo pequeñas modificaciones en algunos de los puntos.

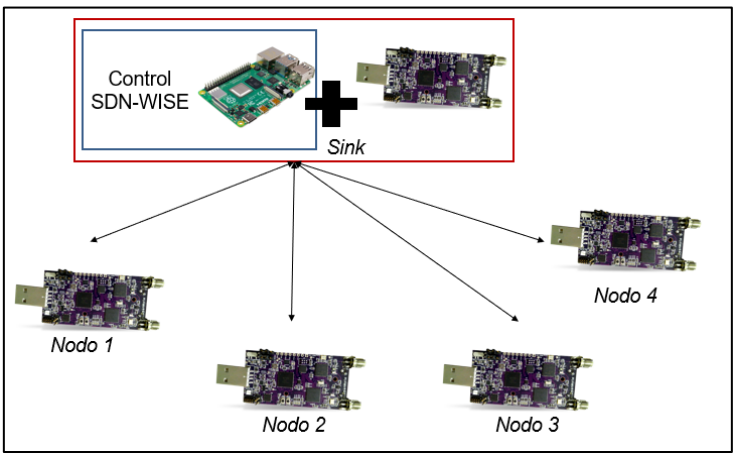


Figura 39. Escenario base de la red SDN-WISE.

8.2.1 Simulación SDN-WISE

Para verificar que el código de los nodos se ha realizado correctamente y las configuraciones añadidas al controlador SDN-WISE son correctas, se ha realizado una simulación de la red inalámbrica mediante la herramienta Cooja. Para ello se ha cargado un dispositivo con el código del *sink* y cuatro dispositivos con el código de los nodos, la distancia entre los dispositivos no alcanza los 10 metros ya que en las pruebas reales no se va a superar dicha distancia. Esta simulación se realiza en un entorno ideal, sin interferencias, por lo que se espera que los valores a obtener sean muy buenos y mejores a los valores reales.

Una vez cargado el escenario de simulación en Cooja se realiza la conexión del *sink* con el punto de control SDN-WISE y se empieza la simulación. Se puede verificar en la interfaz gráfica que se han cargado todos los nodos y el valor del RSSI (*Received Signal Strength Indicator*) entre los dispositivos. En la siguiente figura se puede observar en primer lugar el escenario simulado en Cooja, el *sink* es el dispositivo ubicado dentro de un círculo naranja, y al lado el escenario que se interpreta en la interfaz gráfica del punto de control.

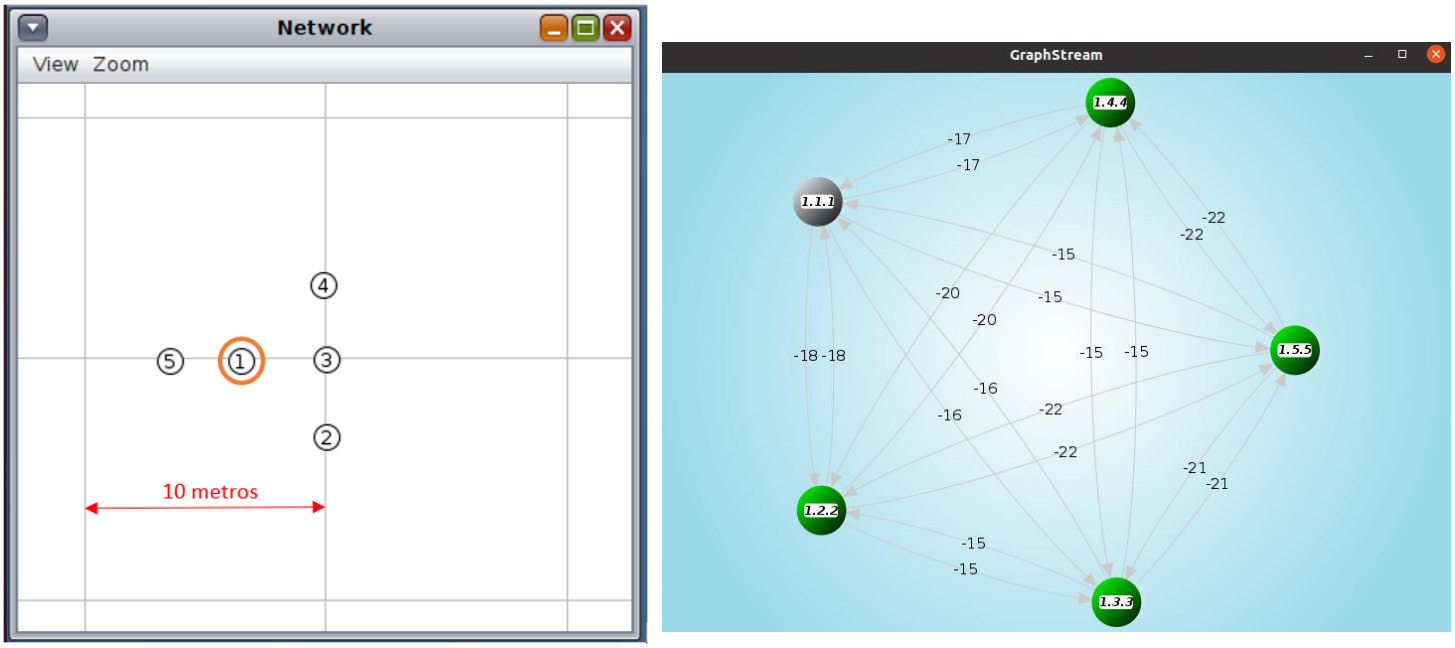


Figura 40. Escenario: a) simulador Cooja, b) gráfico del punto de control.

En primer lugar, se va a analizar los paquetes de *report*, los cuales se han configurado para mandarse cada 8 segundos ya que son paquetes que emplea el controlador para crear la red. Estos paquetes disponen del valor de RSSI y de esta forma se puede saber la robustez de la señal recibida. Este dato es de interés ya que en el caso de los nodos fijos se emplea el valor de RSSI y el número de saltos al *sink* para determinar el camino más fiable y realizar el envío de los datos.

Mediante la herramienta Grafana se pueden visualizar las métricas de interés, como el valor de RSSI, la diferencia de tiempo en milisegundos entre los paquetes de *report* de un mismo dispositivo o el DSR. Un ejemplo del panel descrito se puede observar en la siguiente figura. En ella se puede ver de forma clara cómo la diferencia en milisegundos entre los paquetes de *report* de un mismo dispositivo es aproximadamente de 8000 ms, lo que equivale a los 8 segundos configurados.

Figura 41. Panel de las métricas de los mensajes de *report* en Grafana.

A continuación, se va a analizar el valor de RSSI de los dispositivos con cada uno de los dispositivos vecinos, Figura 42.

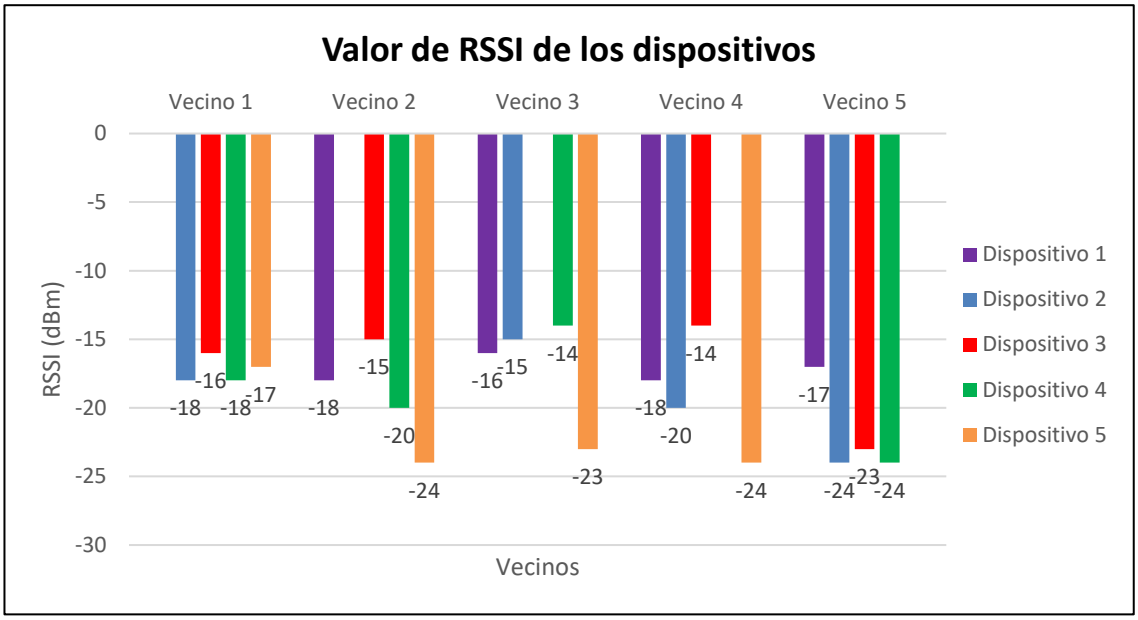


Figura 42. Valor de RSSI de los dispositivos en el simulador Cooja.

Como se observa en la figura, los valores de RSSI son muy buenos, hay que tener presente que se trata de un entorno ideal por lo que en un entono real estos valores empeorarán. Los principales valores a tener en cuenta son los correspondientes al dispositivo 1, el *sink*, ya que es el dispositivo que debe de mandar los *reports* al controlador y por lo tanto es bueno tener un buen nivel de señal a dicho dispositivo.

En cuanto al valor del DSR de los paquetes de *report*, para todos los dispositivos prácticamente el valor es del 100% ya que al ser un entorno simulado prácticamente todos los paquetes llegan en el tiempo establecido.

En segundo lugar, se va a analizar los paquetes de datos que se reciben. Se van a configurar 2 flujos desde el punto de control para todos los nodos, en primer lugar se creará un flujo de 100 ms para todos los nodos y a continuación otro de 400 ms. Para todos estos flujos se va a establecer un marcador con un tiempo de 100 ms para determinar el DSR, teniendo en cuenta un margen del 10%.

Se va a analizar el PDR y el DSR para el primer flujo de 100 ms. En la siguiente figura se pueden observar los flujos que se han creado con destino en el *sink*, seguidamente de la matriz donde se muestra la planificación para cada uno de los dispositivos.

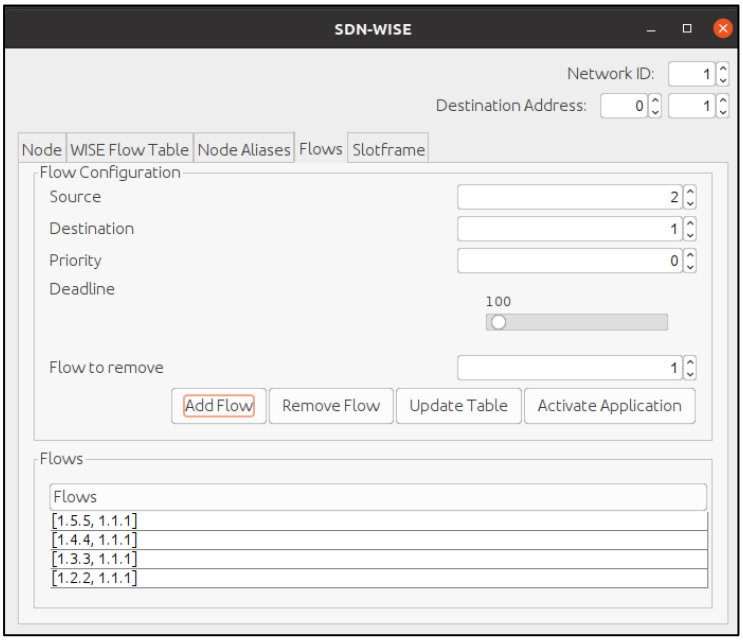


Figura 43. Configuración de los flujos en el punto de control.

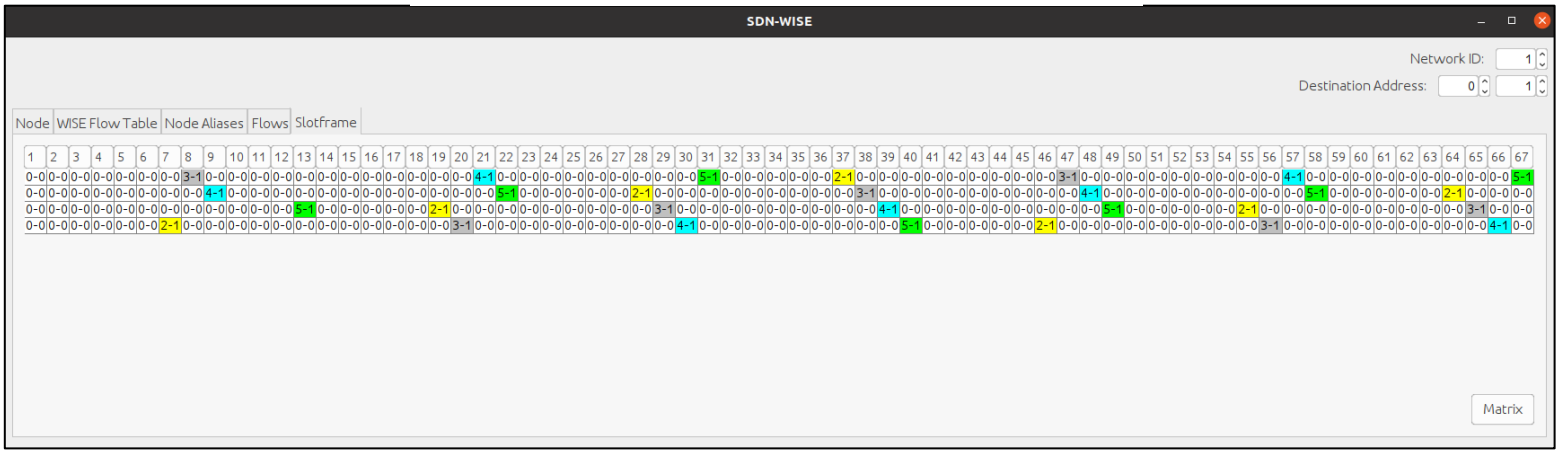


Figura 44. *Slotframe* de la planificación con un *deadline* de 100 ms.

Una vez configurado el flujo para los 4 nodos se va a obtener el valor del PDR, permite observar la relación entre los paquetes enviados y los recibidos. Se ha establecido un flujo de 100 ms y la tasa de transmisión de paquetes configurada en los nodos es de 100 ms, por lo que al tratarse de un entorno ideal se recibirán todos los paquetes que genere el nodo. A continuación, se puede observar, Figura 45, la evolución del PDR en cada uno de los dispositivos, que en todo momento el valor es del 100%.

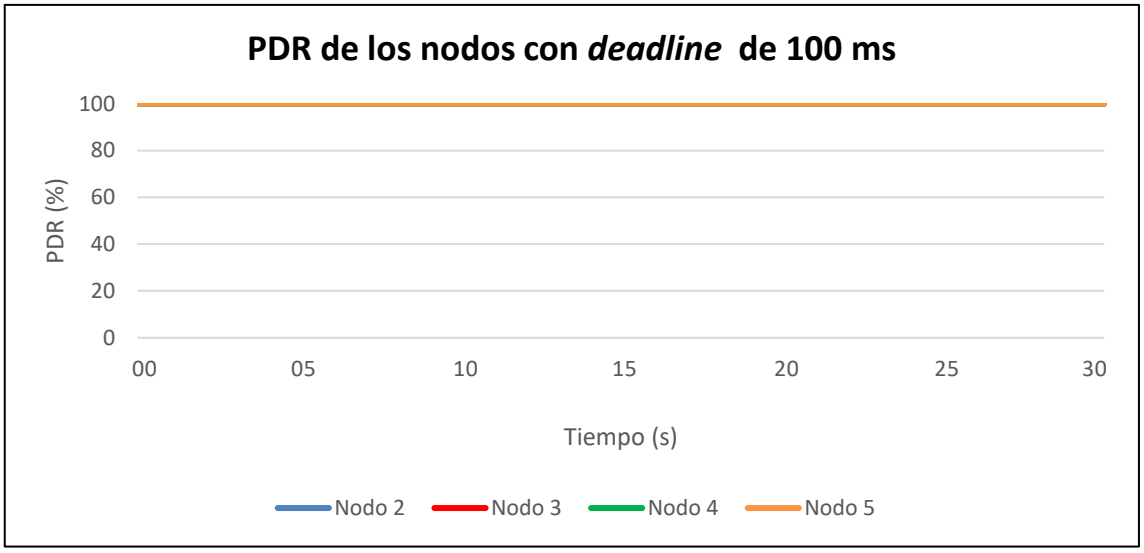


Figura 45. Evolución del valor de PDR de los paquetes de datos con un *deadline* de 100 ms.

En cuanto al DSR, como el límite de tiempo establecido es de 100 ms, el DSR toma un valor próximo al 100% en casi todos los nodos. El valor no es del 100% porque se debe tener en cuenta que también existe un tráfico de control que hace que los tiempos no sean precisos al 100%. En la siguiente figura, Figura 46, se puede observar con mayor profundidad como el valor en todos los nodos se encuentra en torno al 96,5%.

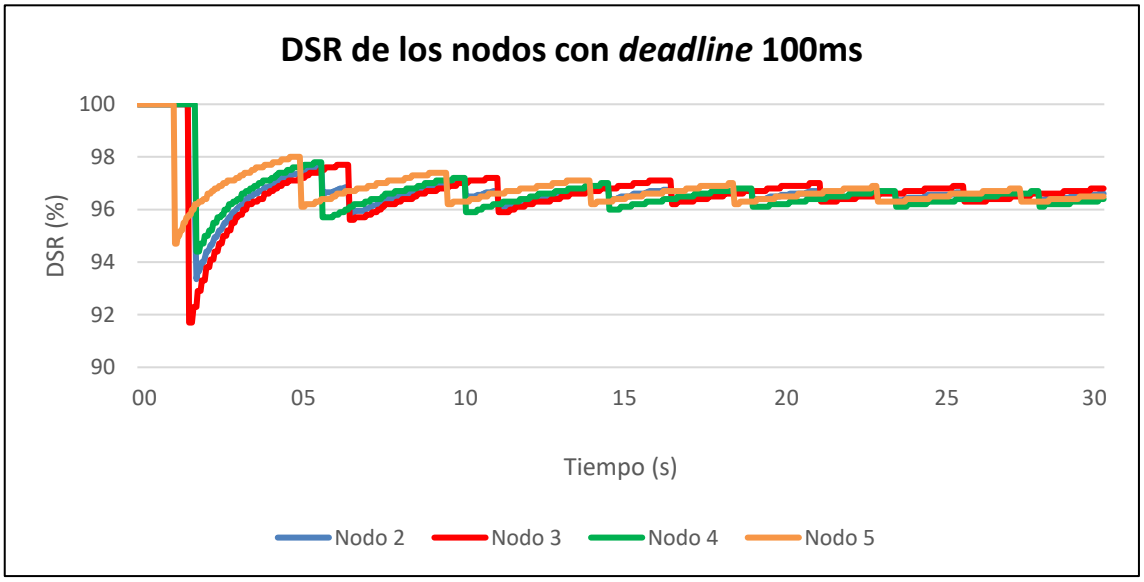


Figura 46. Evolución del valor de DSR de los paquetes de datos con un *deadline* de 100 ms.

Seguidamente, se va a analizar el PDR y el DSR para el segundo flujo de 400 ms. En la siguiente figura se puede observar la matriz donde se muestra la planificación para cada uno de los dispositivos.

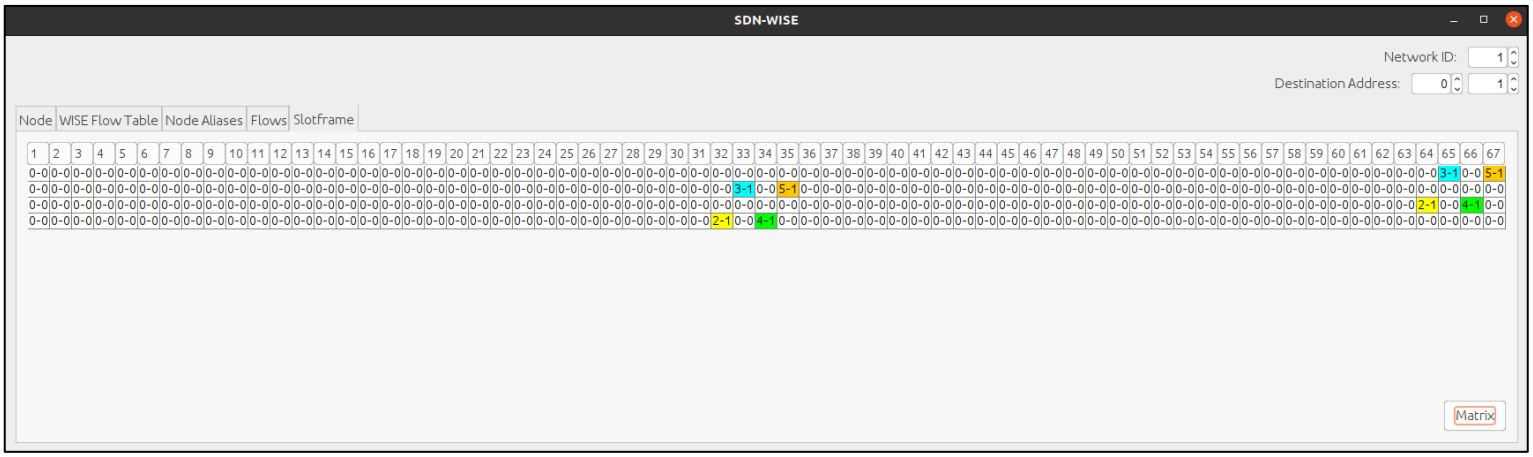


Figura 47. *Slotframe* de la planificación con un *deadline* de 400 ms.

En cuanto al valor del PDR, se puede observar, Figura 48, la evolución del PDR en cada uno de los dispositivos. Se detecta que el valor disminuye hasta alcanzar aproximadamente un valor entorno al 35%. Como se observa los valores han disminuido considerablemente, pero son valores razonables recordando que el nodo se ha configurado para que genere los paquetes de datos cada 100 ms. En este caso los nodos van almacenando los paquetes de datos que faltan por transmitir en colas para ir transmitiendo los paquetes en el intervalo establecido. Las colas se han establecido con un tamaño máximo de 8 paquetes de datos, una vez se alcanza dicho valor se empiezan a descartar paquetes y por ese motivo se producen las pérdidas que se ven reflejadas en el PDR.

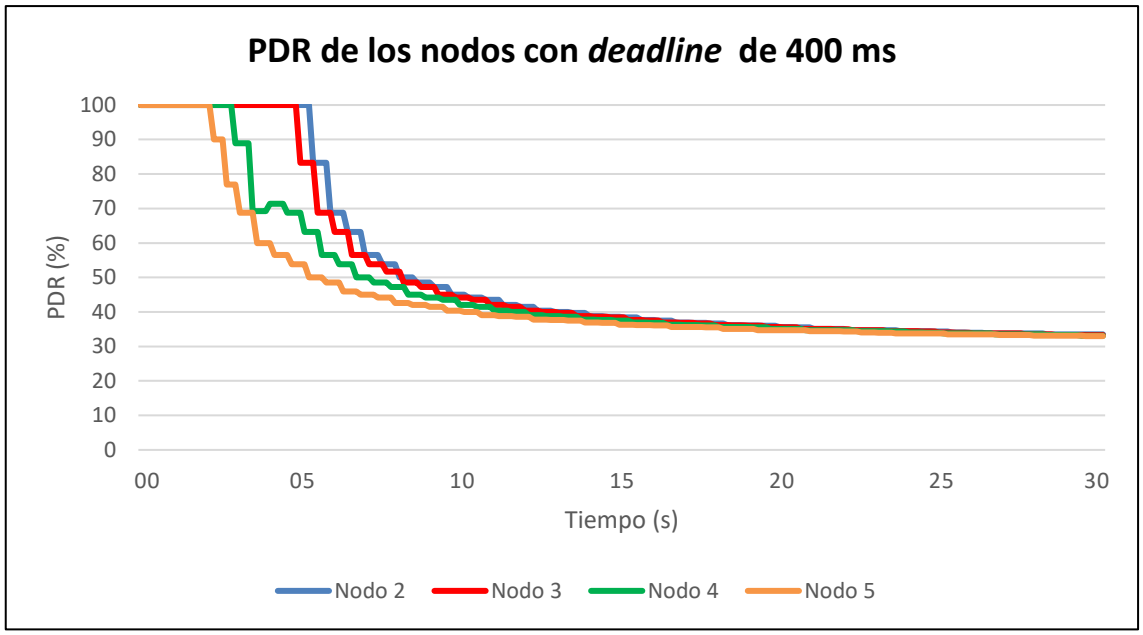


Figura 48. Evolución del valor de PDR de los paquetes de datos con un *deadline* de 400 ms.

En cuanto al DSR, este valor ya no corresponde al 100% ya que el límite temporal está establecido en 100 ms, pero los tiempos entre los paquetes es superior debido a la configuración del flujo. Como se observa, Figura 49, la tendencia es el 0% ya que los datos llegan sobre los 400 ms.

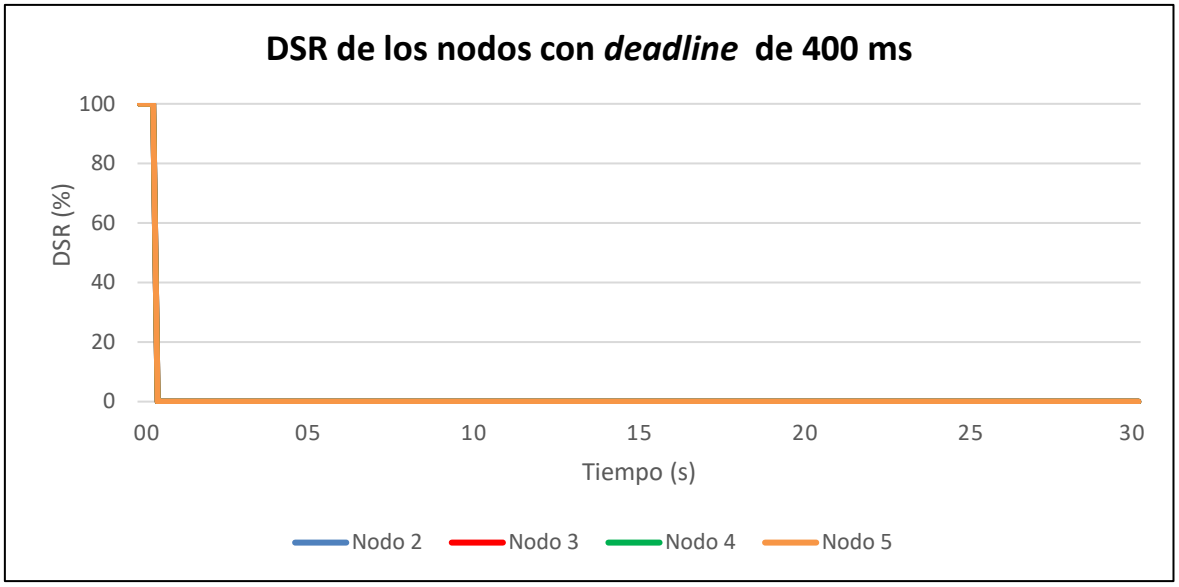


Figura 49. Evolución del valor de DSR de los paquetes de datos con un *deadline* de 400 ms.

Mediante estas simulaciones se ha podido verificar que las herramientas configuradas y los dispositivos programados se ha realizado de forma correcta. Los resultados obtenidos hay que recordar que se han obtenido en un entorno ideal, pero sirven para observar el comportamiento que va a tener la red en el ámbito real.

8.2.2 SDN-WISE con 4 nodos fijos en un entorno real

A continuación, se va a realizar el montaje de la misma red que se ha simulado anteriormente en Cooja pero en un entorno real. La ubicación de los dispositivos es la que se muestra a continuación, con una separación inferior a los 10 metros entre ellos.

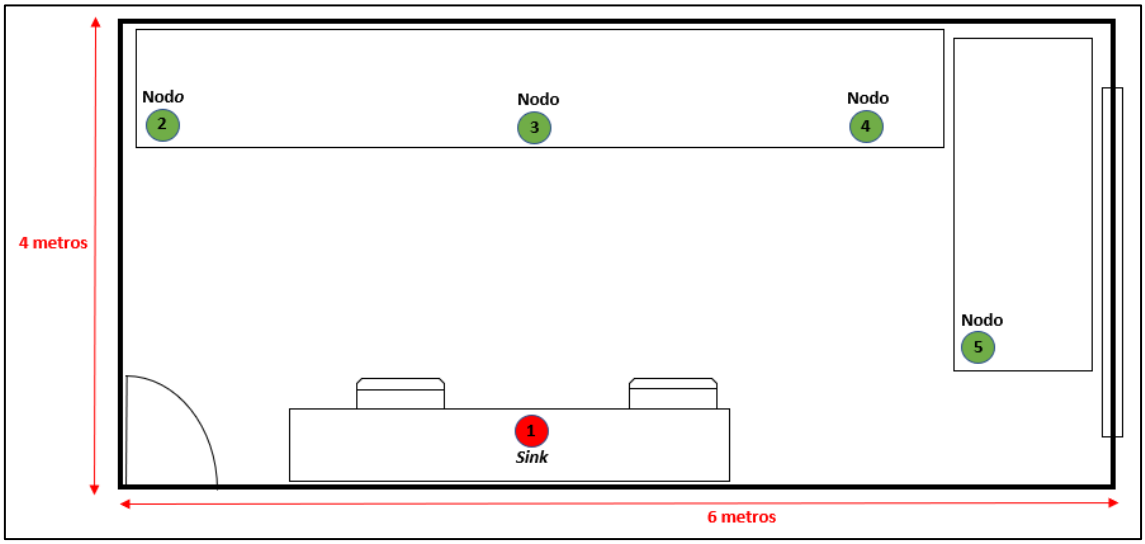


Figura 50. Escenario con los 4 nodos fijos en el entorno real.

Se va a analizar los paquetes de *report* para observar la variación del RSSI de un entorno simulado a uno real.

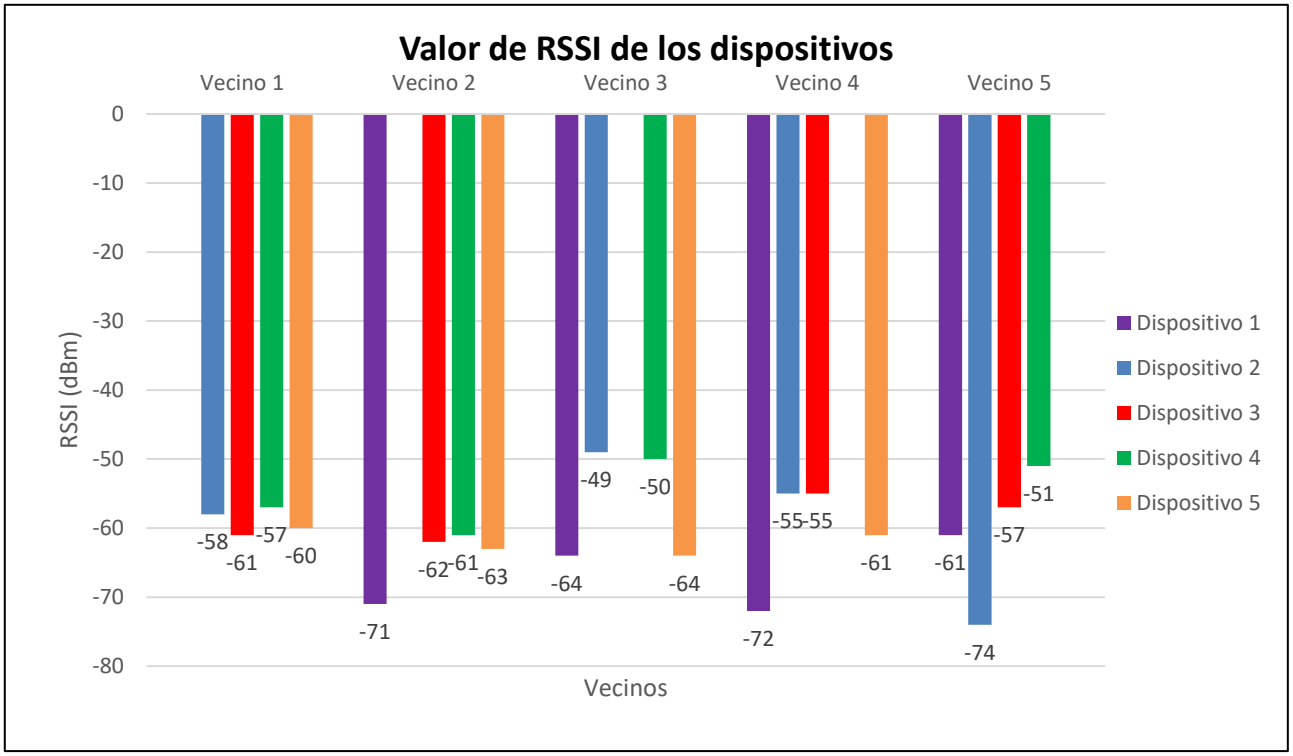


Figura 51. RSSI en un entorno real y los 4 nodos fijos.

Se puede observar, Figura 51, que existe una gran variación del entorno simulado al entorno real, obteniendo una diferencia de unos 42 dBm. Los valores son altos, pero son valores que se consideran aceptables debido al gran número de interferencias que existe en el ambiente. Todos los valores rondan un valor de -60 dBm, el valor que se toma como un buen nivel de señal.

A continuación, se va a observar la evolución del DSR para observar la diferencia de tiempos entre paquetes de *report*. Hay que recordar que el tiempo configurado para el envío de dichos paquetes es de 8 segundos. Para el cálculo de dicho valor se ha establecido un límite temporal de 8000 ms, con un margen del 1%.

Como se observa, Figura 52, dicho valor deja de valer 100% como ocurría en el entorno simulado, ya que los tiempos ahora tienen en cuenta un mayor número de factores como la saturación que pueda tener el nodo. Se observa que el dispositivo 1 toma un valor del 100% ya que es el único conectado directamente al controlador, el resto llega a disminuir hasta un 40% hasta llegar a estabilizarse.

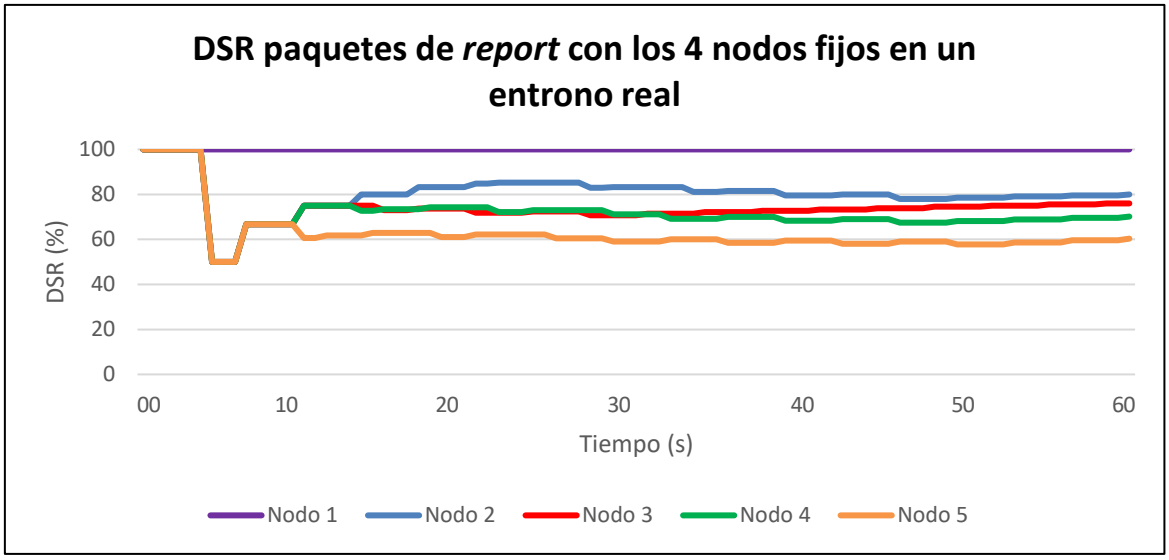


Figura 52. Valor de DSR para los paquetes de *report* con 4 nodos fijos en un entorno real.

Se van a añadir los flujos para los paquetes de datos al igual que en la simulación, un flujo de 100 ms y otro de 400 ms con una tasa de generación de paquetes configurada en los nodos de 100 ms.

Primero se va a analizar el flujo de 100 ms, más concretamente el valor de PDR de los nodos. En la Figura 53, se puede observar cómo los nodos actúan correctamente en cuanto al envío de los datos, ya que no se producen pérdidas y se obtiene un PDR del 100% en todos los nodos en todo momento.

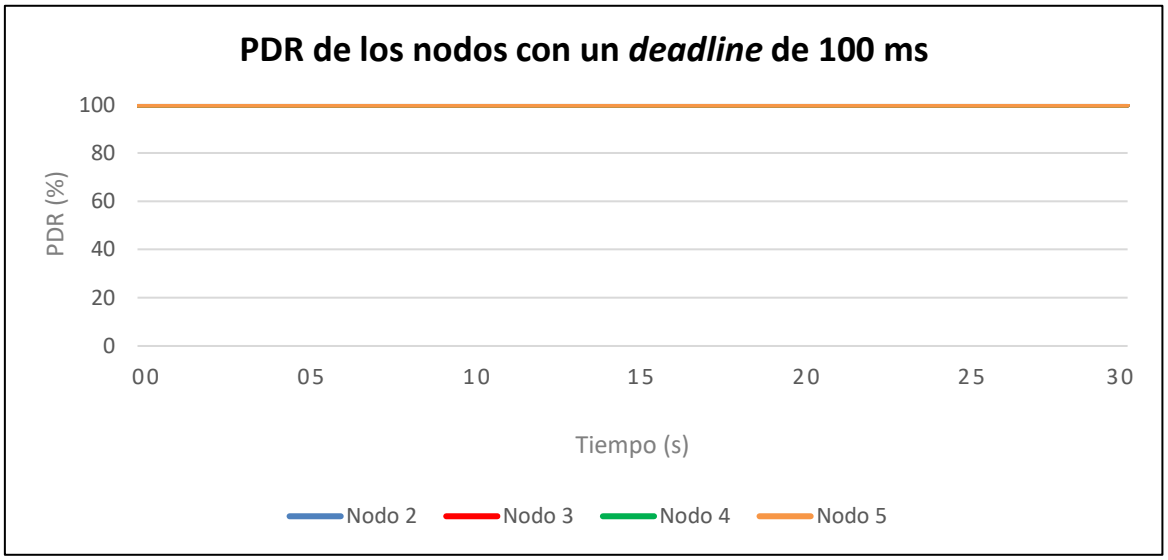


Figura 53. PDR de los paquetes de datos de los nodos fijos con un *deadline* de 100 ms en un entorno real.

En cuanto al DSR, este valor si ha empeorado considerablemente, llegando a reducirse un 20% como se observa en la Figura 54. Esto se debe a que en la transmisión de los paquetes de datos se produce un pequeño retraso debido a que se puede saltar alguno de los slots asignados y enviarlo por el siguiente, pero no en todos ellos. Era de esperar que las pruebas en el entorno real perjudicaran dicho valor, reduciéndose hasta alcanzar un valor del 75% aproximadamente en casi todos los nodos al estabilizarse.

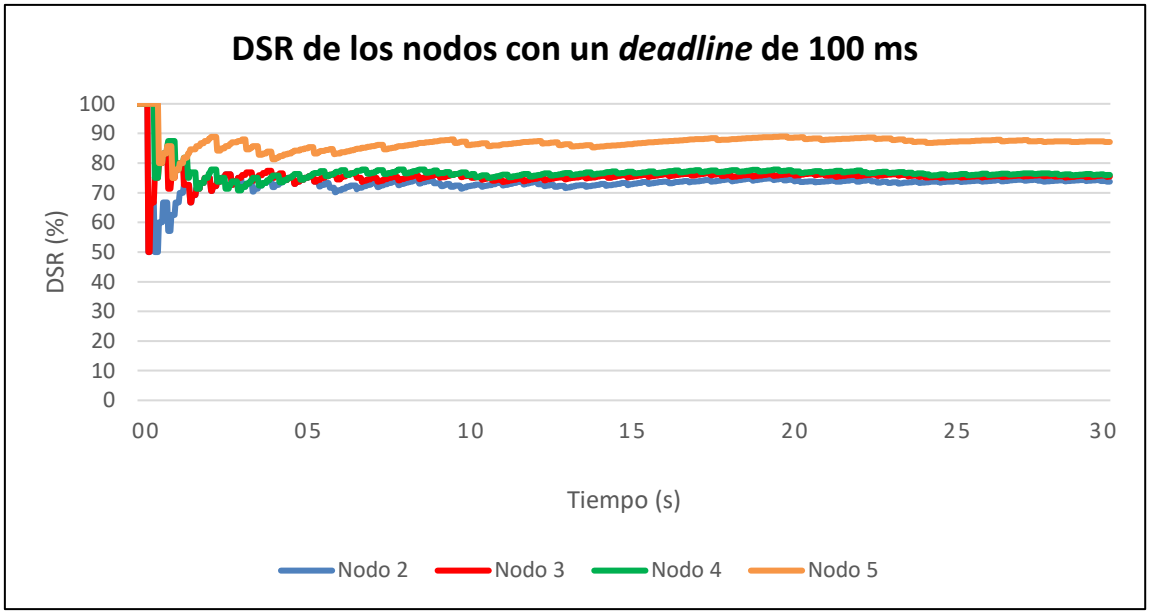


Figura 54. DSR de los paquetes de datos de los nodos fijos con un *deadline* de 100 ms en un entorno real.

En segundo lugar, se va a analizar el flujo de 400 ms, empezando por el valor de PDR. En la Figura 55, se puede observar cómo dicho valor se asemeja mucho al obtenido en las simulaciones, no llegando a superar un margen del 7% de los resultados obtenidos en el punto anterior. Este es un buen indicador, ya que refleja que el comportamiento de los nodos es el esperado.

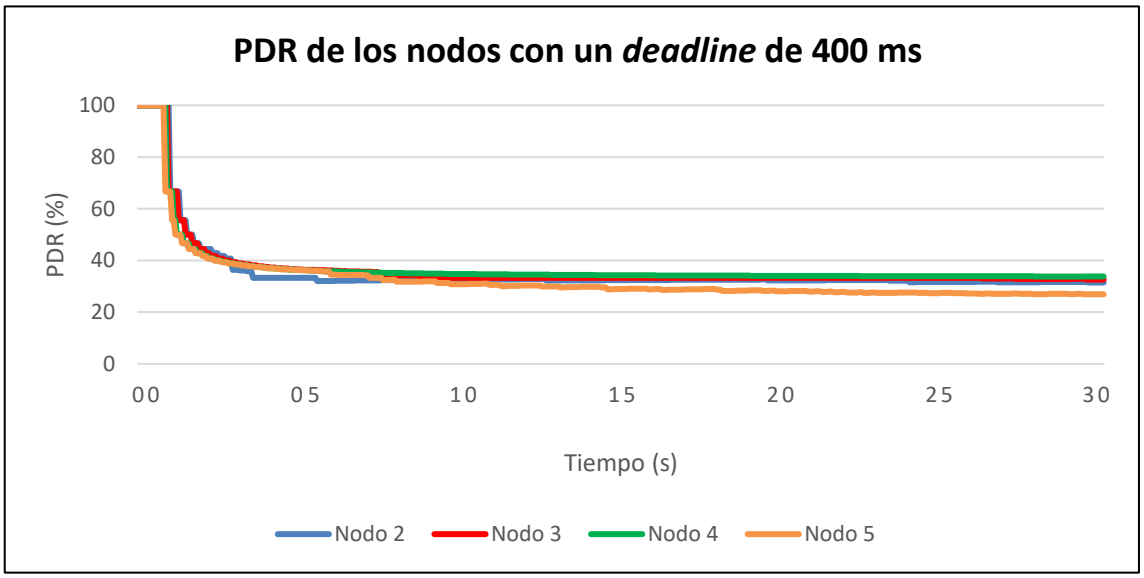


Figura 55. PDR de los paquetes de datos de los nodos fijos con un *deadline* de 400 ms en un entorno real.

El DSR en este caso presenta los mismos resultados que en la simulación, Figura 49, debido a que el límite establecido es de 100 ms y con el flujo establecido todas las trasmisiones superan dicho valor alcanzando un valor del 0% de DSR en todos los nodos.

8.2.3 SDN-WISE con 3 nodos fijos y uno móvil

En la siguiente prueba se va a modificar el escenario anterior, se va a realizar una red inalámbrica, pero uno de los nodos irá montado en un robot para ofrecer movilidad al nodo móvil. De esta forma, permite estudiar el impacto de la movilidad en los nodos. Como se puede observar, ahora el nodo 5 irá moviéndose a lo largo de un circuito establecido por una línea azul, girando cada vez que llegue al final de la línea.

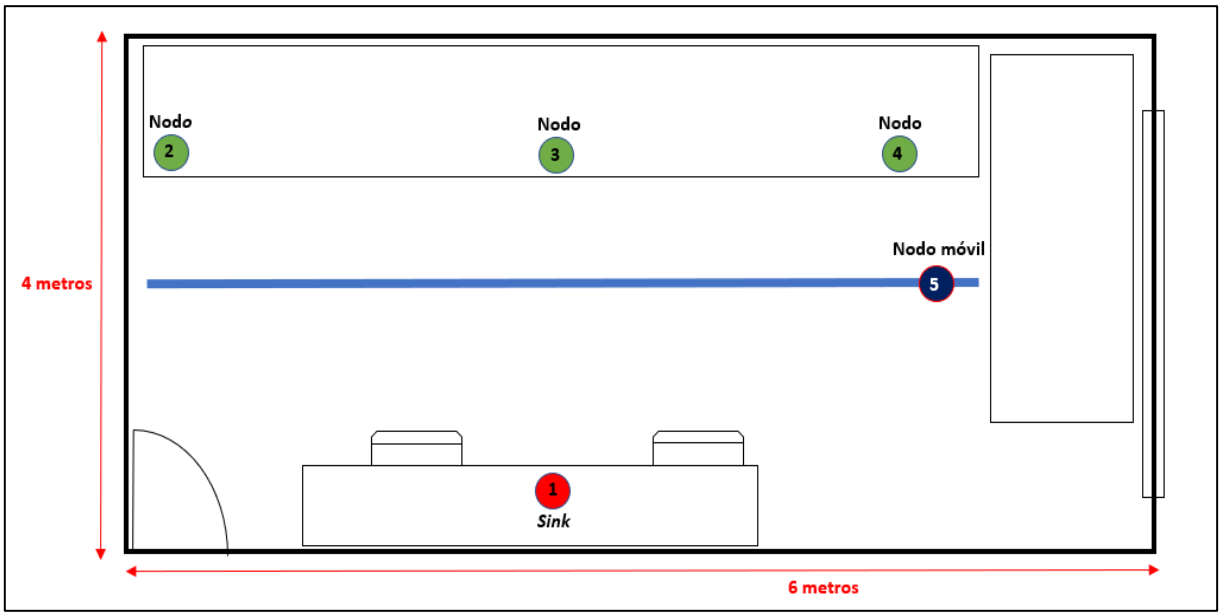


Figura 56. Escenario con los 3 nodos fijos y 1 móvil.

Figura 57. Movilidad del nodo mediante el robot.

Para realizar el estudio primero se evalúa el impacto en los paquetes de *report*, ya que ofrecen el valor de RSSI. Como el resto de los nodos se mantienen fijos se va a prestar una mayor atención al valor de RSSI del nodo 5 que es el móvil y es el que presenta las mayores variaciones. En la siguiente figura, Figura 58, se puede observar el valor de RSSI que disponen los dispositivos con el nodo 5.

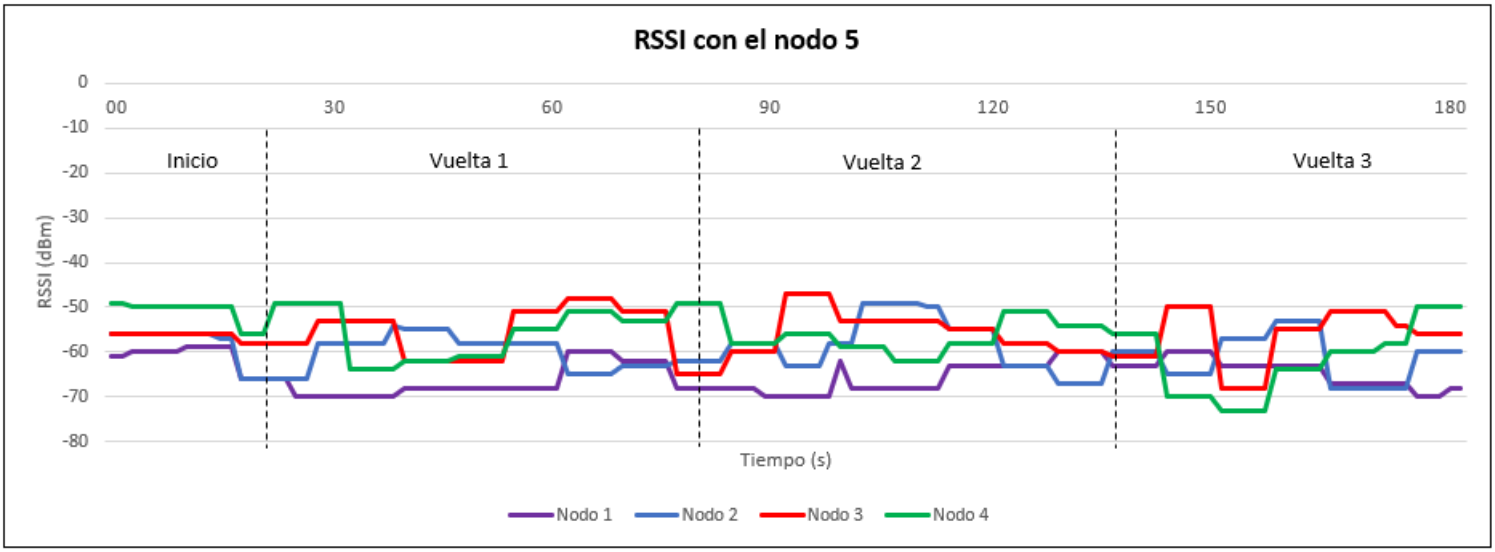


Figura 58. RSSI de los dispositivos con el nodo 5.

En este instante si se crea un flujo de datos del nodo 5 al *sink*, se puede observar cómo el camino se establecería del 1 al 5 al ser el camino con menor número de saltos. Pero el nivel de RSSI es el peor de todos los caminos. Se puede observar que conforme el robot va moviéndose y acercando el nodo 5 al resto de los nodos el valor de RSSI mejora considerablemente. Por lo tanto, el camino actual es el menos óptimo en cuanto a calidad de servicio. Por otro lado, la movilidad del nodo también aporta una modificación constante de la red, lo que hace que sea más inestable. Esto se puede observar en el valor del DSR, Figura 59, el cual ha empeorado considerablemente llegando a alcanzar valores del 10% al estabilizarse.

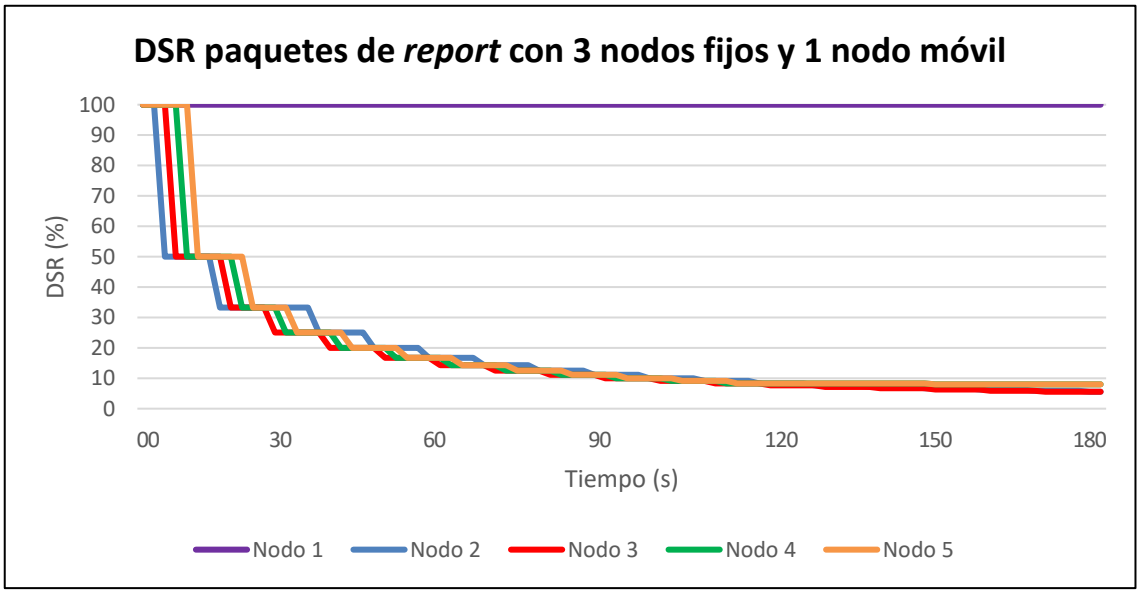


Figura 59. DSR de los paquetes de *report* con 3 nodos fijos y 1 nodo móvil.

A continuación, se va a estudiar cómo afecta la movilidad a los paquetes de datos. Para esa prueba se ha decidido modificar la tasa de creación de paquetes a 250 ms y añadiendo un único flujo de 250 ms a todos los nodos.

Primero se va a evaluar el valor de PDR, en la siguiente figura se puede observar cómo dicho valor ha empeorado en algunos dispositivos. La diferencia es pequeña ya que no llega a ser superior al 20%, pero ya se ha producido una pérdida de paquetes. Este hecho es de gran importancia ya que en anteriores pruebas siempre había sido del 100%. En el ámbito industrial hay tráficos sensibles a las pérdidas, llevando a detener procesos industriales por fallos como este. Este hecho hace ver que la movilidad del nodo 5 si ha tenido una repercusión en el tráfico.

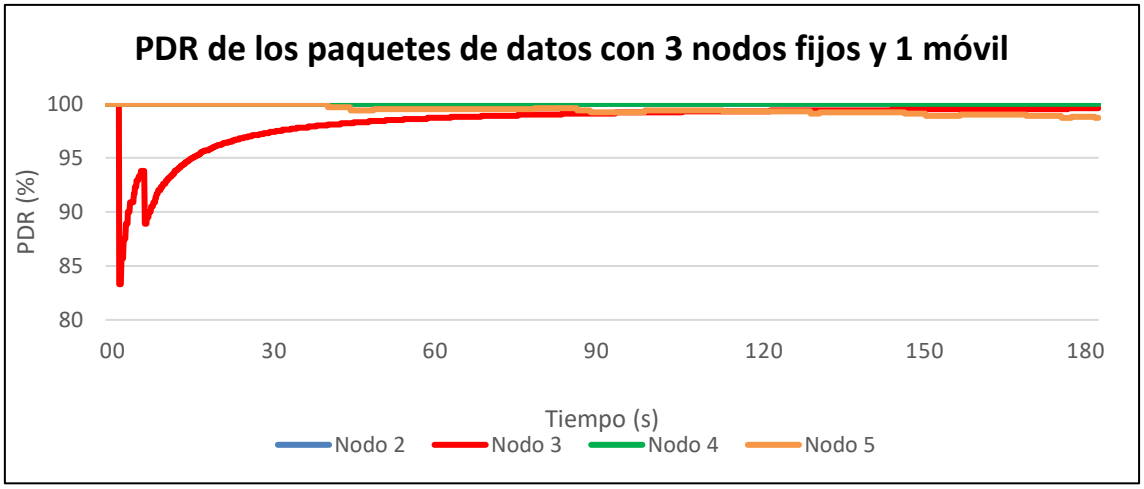


Figura 60. PDR de los paquetes de datos con 3 nodos fijos y 1 móvil.

Si se observa el valor de DSR, Figura 61, este ha empeorado también en todos los dispositivos. Ahora no llega a estabilizarse hasta alcanzar un valor de 50%, por lo que se ha reducido más de un 20% que en las pruebas anteriores. Esto quiere decir que dentro de los paquetes que se han recibido, la mitad se han recibido superados los 250 ms, asumiendo un margen del 10%.

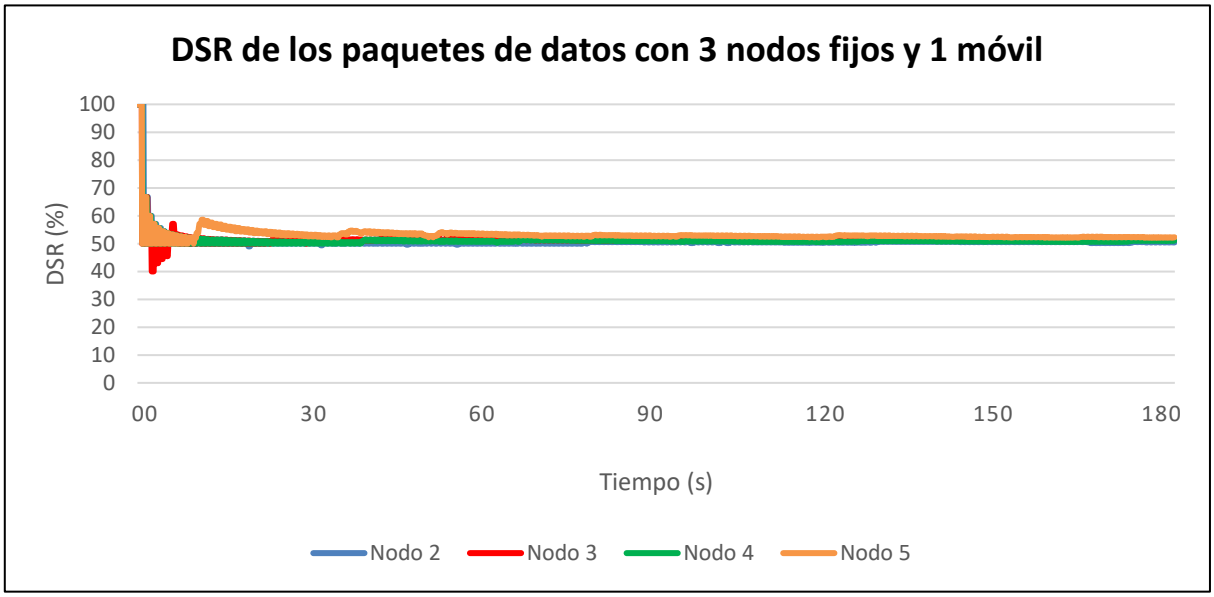


Figura 61. DSR de los paquetes de datos con 3 nodos fijos y 1 móvil.

Por lo tanto, se ha podido observar que la movilidad en los nodos afecta a las comunicaciones. Por este motivo, se ha decidido crear una aplicación en el controlador que permita tratar los nodos móviles de forma distinta. En el siguiente punto se comentarán las mejoras efectuadas.

8.2.4 SDN-WISE con 3 nodos fijos y uno móvil con nuevas mejoras

En este punto se ha añadido una nueva mejora en los nodos para mejorar la movilidad. Primero se ha configurado el nodo móvil para que el camino lo establezca con aquel dispositivo que tiene un mejor nivel de RSSI, sin tener en cuenta el número de saltos como ocurría en el caso anterior. También se ha modificado la transmisión de los dispositivos móviles, estos ahora realizarán transmisiones *broadcast*.

A continuación, en el controlador se ha añadido la posibilidad de añadir nodos con la especialidad de ser móviles. Esta especialidad se encarga de tratarlos de forma distinta, como por ejemplo añadir un flujo de datos cada vez que se reciba un paquete de *report* por un camino nuevo. A la hora de crear el flujo se ha creado un nuevo procesamiento que permite en el *slotframe* tener celdas compartidas.

Este nuevo procesamiento permite que los nodos móviles tengan asignados siempre los mismos slots, de forma que, cuando se establezca un flujo de datos con un nuevo nodo los slots se asignarán de forma normal, pero si se establece otro flujo con otro dispositivo este asignará los mismos slots al nodo móvil y se agregarán los flujos intermedios entre el nodo móvil y el *sink* dependiendo de estos slots. Este mecanismo permite que el nodo móvil transmita de forma *broadcast*, y el resto de los nodos sepan por dónde van a recibir los datos provenientes de dicho nodo.

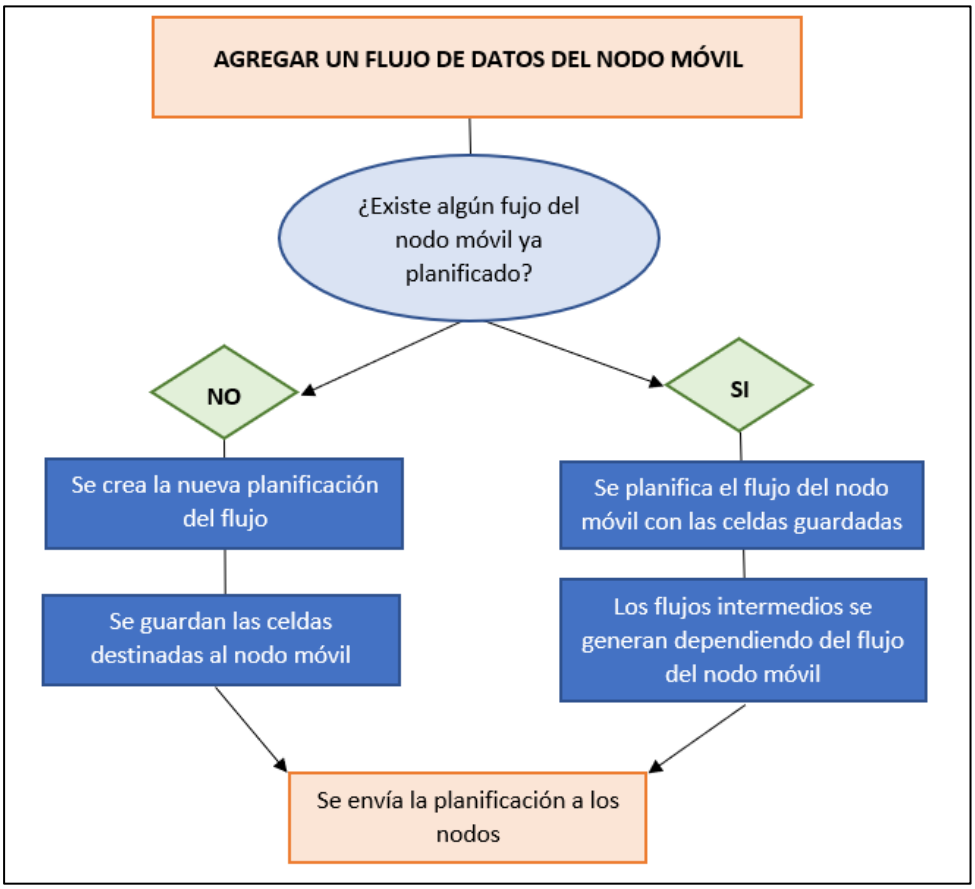


Figura 62. Nuevo proceso implementado en el controlador SDN-WISE.

En el diagrama, Figura 62, se muestra de forma resumida el procesamiento que realiza el controlador para asignar los flujos de los nodos móviles. En la siguiente figura se puede ver de forma clara cómo se va formando el *slotframe* conforme el nodo móvil se va moviendo y se van creando flujos con el nuevo procesamiento. Esto permite que el nodo móvil se vaya desplazando y vaya transmitiendo por aquel nodo que tenga un mejor nivel de RSSI.

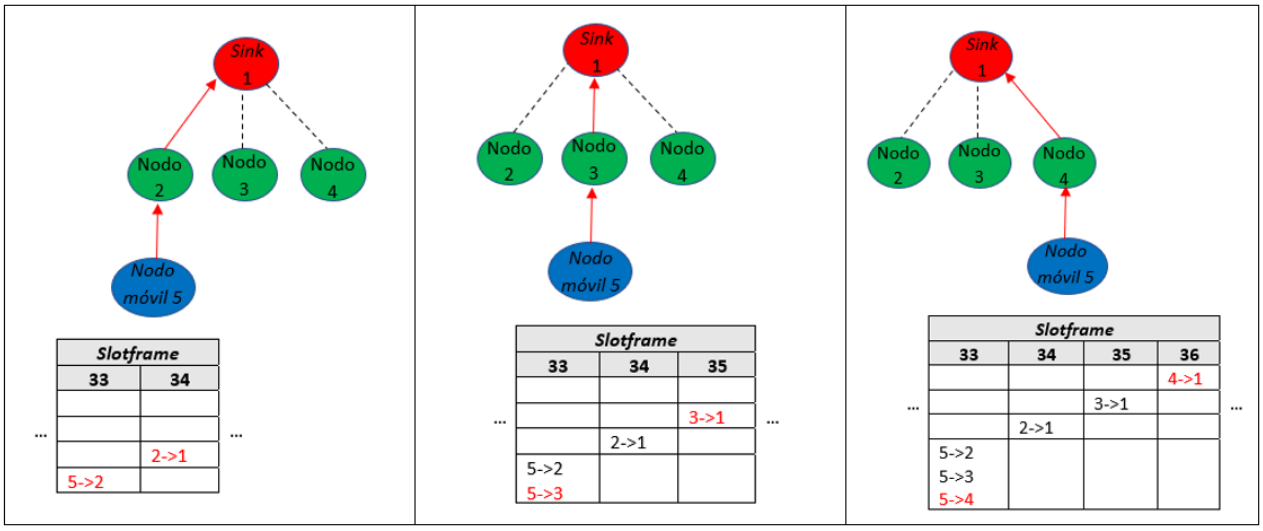


Figura 63. Asignación en el *slotframe* con el nuevo proceso implementado.

A continuación, Figura 64, se puede observar cómo se muestra en el *slotframe* la asignación de slots para el nodo 5, el nodo móvil, para el *deadline* establecido de 250 ms para los paquetes de datos. Se puede observar en un cuadro rojo aquellos slots compartidos, en los cuales únicamente transmite el nodo 5. El resto de los slots se van asignando teniendo en cuenta la ubicación de los slots móviles.

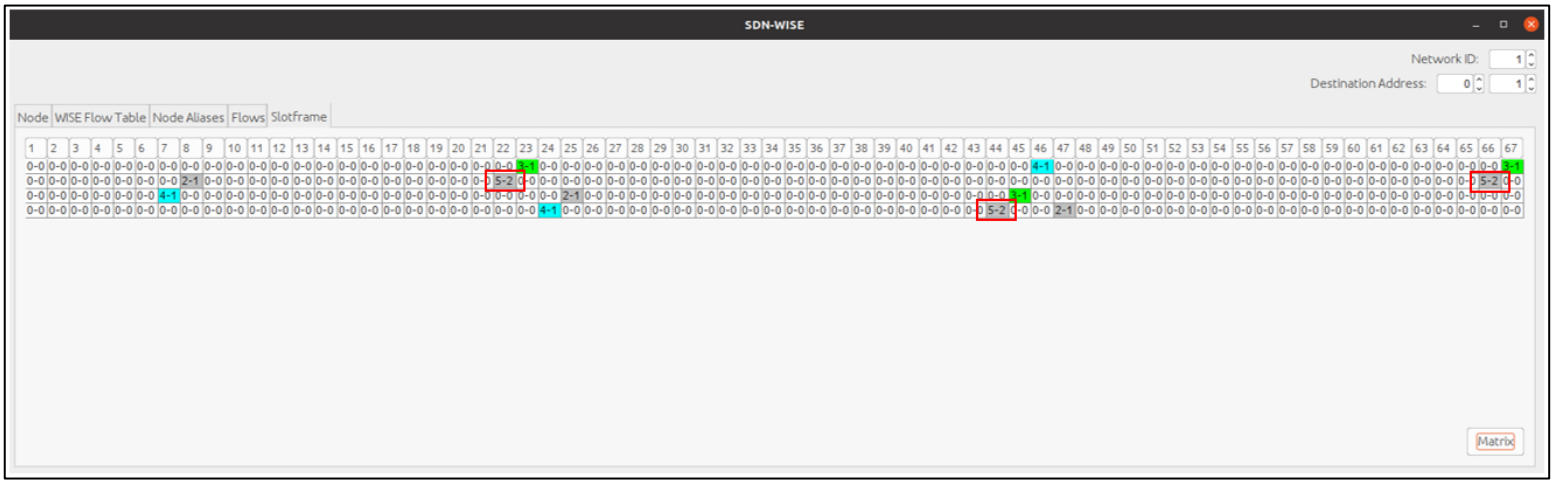


Figura 64. *Slotframe* con las mejoras de movilidad.

Una vez implementada dicha mejora, se van a realizar las mismas pruebas que en el punto anterior para observar las mejoras que ofrece. En primer lugar, los resultados obtenidos de los paquetes de *report*.

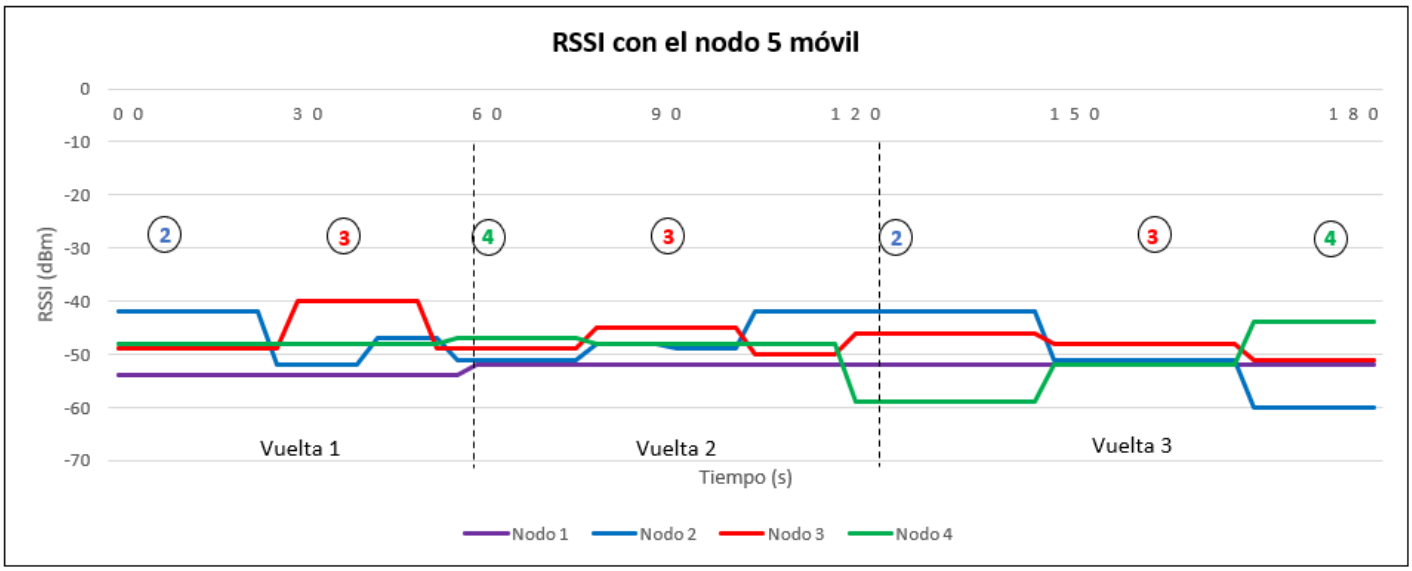


Figura 65. RSSI con el nodo 5 móvil mejorado.

Se va a analizar el valor del RSSI, el cual va a indicar qué camino va a tomar el nodo 5 conforme se va moviendo. Se observa, Figura 65, que según el nodo 5 se va moviendo por el circuito, este va modificando su ruta conectándose a aquel nodo con mejor señal. En la gráfica se puede observar el número de vueltas que se realizan y el nodo del círculo es al cual se va conectando según se va moviendo.

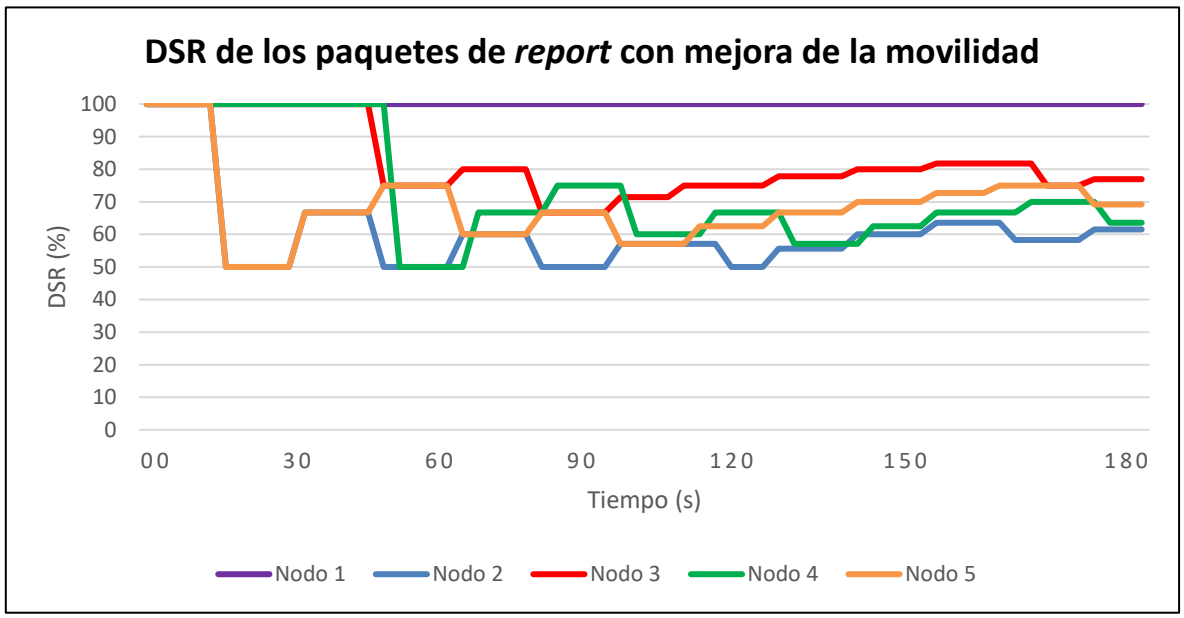


Figura 66. DSR de los paquetes de *report* con la mejora en la movilidad.

Si se observa la gráfica del DSR de los paquetes de *report*, Figura 66, se puede observar cómo existe una mejora considerable. En este caso los valores son superiores al 50% en todo

momento, lo que supone una mejora considerable al punto anterior. Los nodos tienen una menor carga permitiendo que los tiempos sean más correctos y exista una menor saturación.

A continuación, se van a analizar los paquetes de datos. Se va a observar los valores de PDR, Figura 67, los cuales tienen un valor del 100% en todo momento. Esto es una gran mejora, ya que implica que no se ha perdido ningún paquete durante la transmisión.

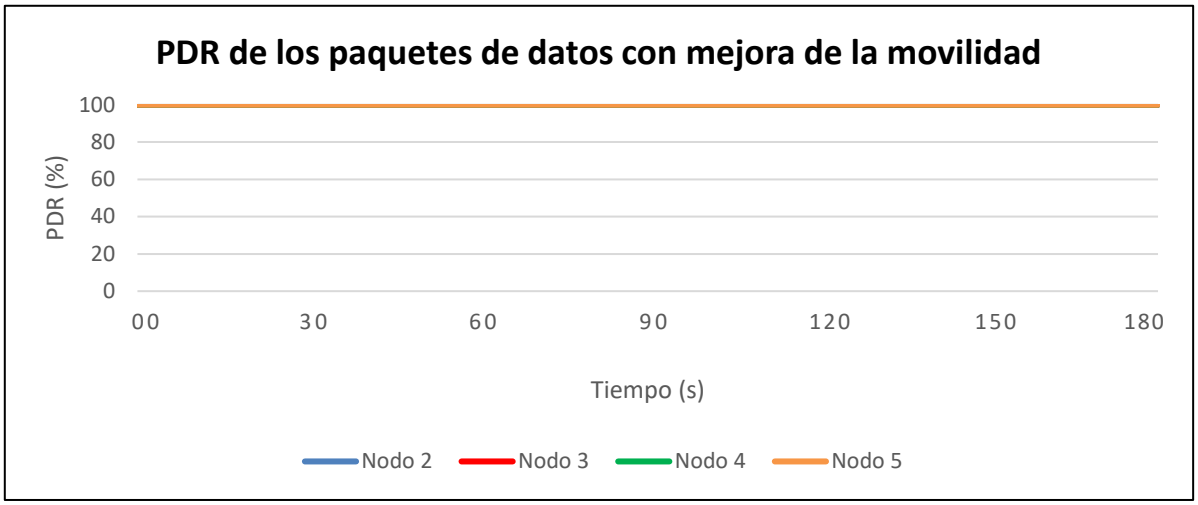


Figura 67. PDR de los paquetes de datos con la mejora en la movilidad.

En cuanto a los valores de DSR, Figura 68, estos han mejorado en ciertos dispositivos, pero no han empeorado en ninguno de ellos. En la anterior prueba los valores de DSR no llegaban a estabilizarse hasta alcanzar el 50%, actualmente se estabilizan al alcanzar valores del 60%.

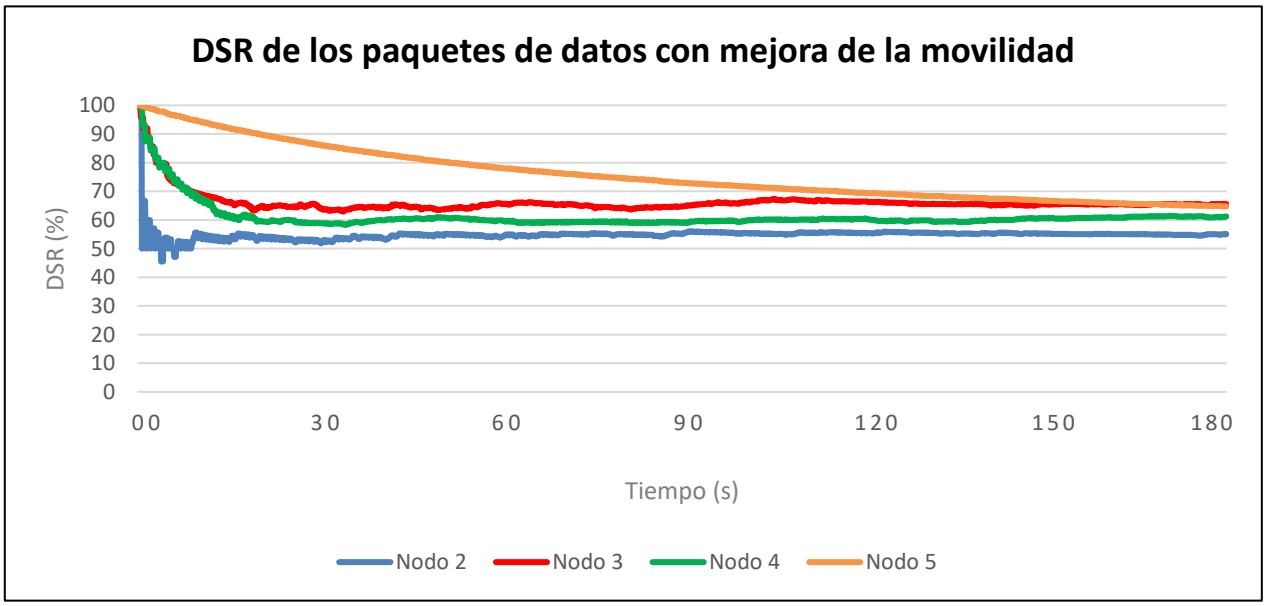


Figura 68. DSR de los paquetes de datos con la mejora en la movilidad.

Por lo tanto, se puede demostrar que la mejora realizada para aquellos nodos que se encuentren en movimiento ayuda a las comunicaciones y las favorece en todos los aspectos. Este es un punto muy importante a tener en cuenta en la industria, ya que actualmente existen muchos robots móviles que se encargan del suministro de piezas, por ejemplo, y este mecanismo favorecería las comunicaciones en caso de disponer de ellos

8.2.5 Montaje de la red SDN-WISE

Finalmente se ha realizado el montaje con ambas redes inalámbricas, una con los nodos fijos y otra con un nodo con movilidad, como se mostró en el apartado de objetivos. Para ello se ha modificado el controlador SDN-WISE para que permita el manejo de varias redes, pudiendo agregar flujos a las distintas redes y mostrar el *slotframe* por separado de cada una de ellas. Esta mejora permite una gestión de las redes de forma centralizada permitiendo una diferenciación de las redes.

A continuación, se pueden observar ambas redes juntas, quienes el controlador permite gestionarlas de forma separada, pero mostrándolas en el mismo panel, Figura 69.

Por un lado, se dispone de la red 1, dicha red cuenta con un dispositivo *sink*, 3 nodos fijos y un nodo móvil. También se dispone de otra red inalámbrica, la red 2, la cual cuenta con un dispositivo *sink* y 4 nodos fijos. Como se observa, los dispositivos se pueden diferenciar claramente ya que cuentan con un primer número que indica el número de red y los dos siguientes números que son el identificador del dispositivo. El controlador SDN-WISE por lo tanto es capaz de gestionar ambas redes por separado.

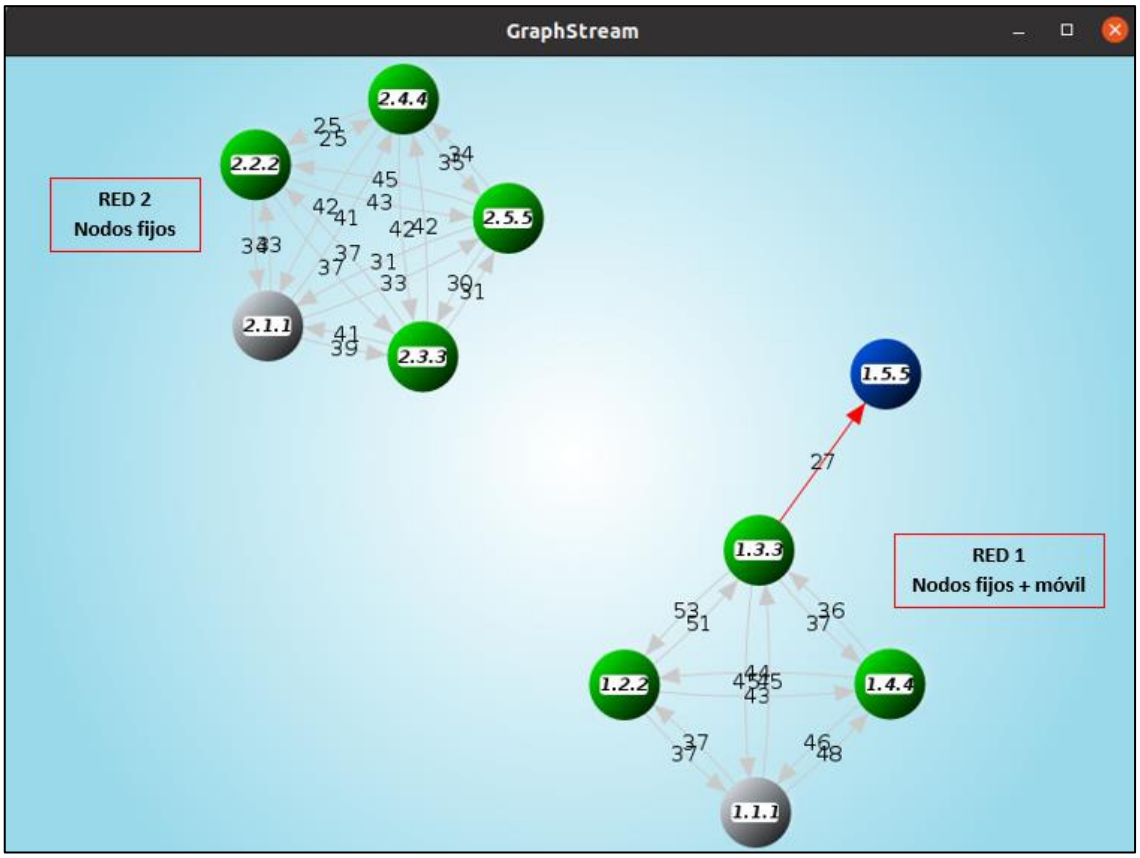


Figura 69. Escenario final del controlador SDN-WISE.

Por otro lado, en la herramienta Grafana se pueden observar las mediciones de cada sensor por separado para tener el entorno controlado.

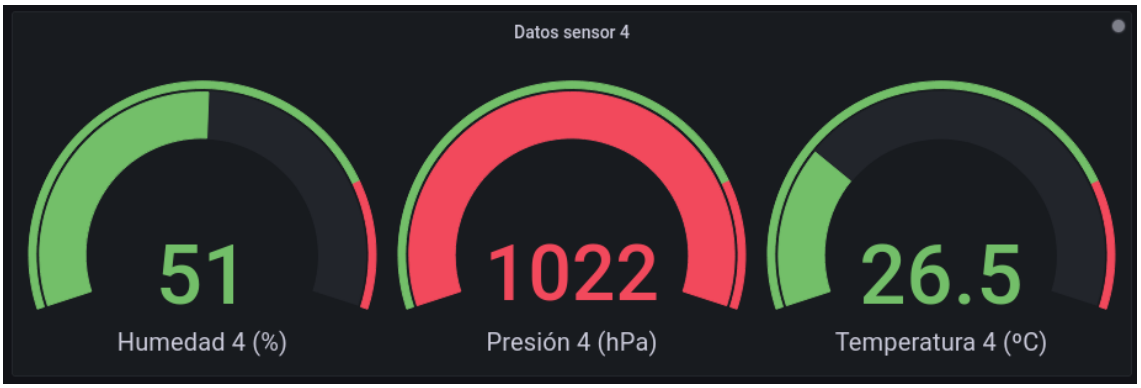


Figura 70. Datos de los sensores en la herramienta Grafana.

Capítulo 9. Resultados

9.1 Red cableada SDN

Dado el desarrollo de la parte de la red cableada SDN, los resultados en cuanto a paquetes recibidos son los que se muestran a continuación de forma gráfica.

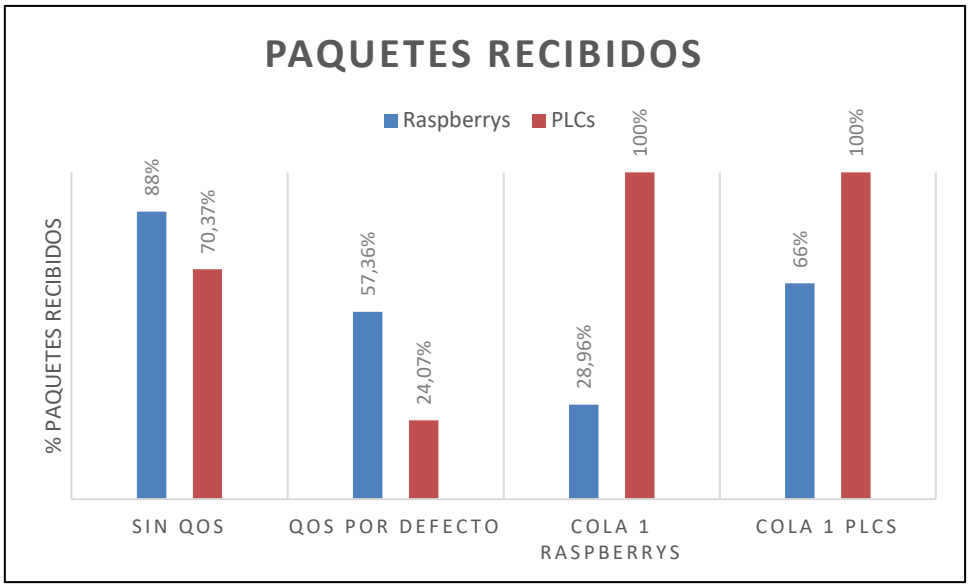


Figura 71. Resultado paquetes recibidos de las pruebas de la red cableada SDN.

Como se observa, Figura 71, el hecho de emplear o no QoS afecta bastante a las comunicaciones, ya que sin el empleo de colas se observa que las comunicaciones de cierto tráfico pueden predominar con respecto a otros tráficos.

Por ese motivo, es necesario el empleo de colas en aquellos tráficos que sean importantes y necesiten un ancho de banda determinado para poder realizar la comunicación sin que afecte la saturación del otro tráfico.

En la gráfica se observa cómo dependiendo de dónde se asignen las colas se podrá ofrecer un mayor rendimiento de la red, viendo las necesidades de cada tráfico y ajustando las colas. Por ese motivo, si en lugar de asignar la cola 1 al tráfico de las Raspberrys se asigna al de las PLCs que requieren un menor ancho de banda, se obtienen unos mejores resultados en cuanto a paquetes recibidos.

Otros términos a tener en cuenta en la QoS son el PDR y el DSR. En este caso se ha analizado del tráfico de las PLCs. La siguiente gráfica, Figura 72, muestra cómo el empleo de colas afecta muy favorablemente a dichos términos ya que alcanzan el 100%.

Esto quiere decir que no se han perdido paquetes y los que han llegado lo han hecho en el periodo de tiempo establecido.

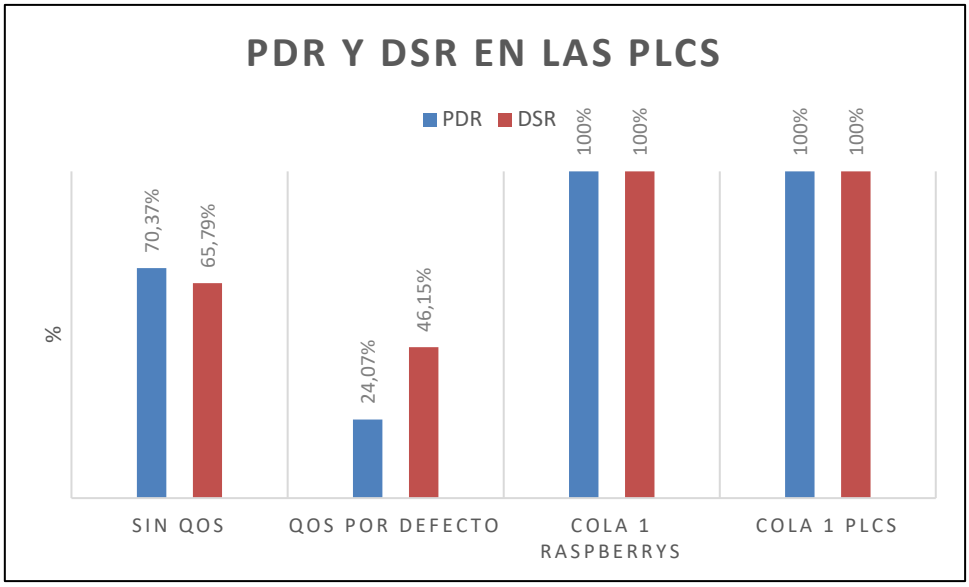


Figura 72. PDR y DSR de las PLCs de las pruebas de la red cablead SDN.

9.2 Red inalámbrica SDN-WISE

9.2.1 Comparación del entorno simulado y del entorno real

Como se comentó al principio, el simulador Cooja permite verificar el funcionamiento de los nodos, pero se debe de tener en cuenta en todo momento que se trata de un entorno limpio y sin interferencias, lo que permite obtener muy buenos resultados.

Si se comparan los valores de RSSI obtenidos, los valores en las simulaciones son mejores que los del entorno real. En la siguiente figura se muestra la diferencia en dBm del entorno simulado con el entorno real. Como se observa, la media se encuentra en los 42 dBm de diferencia, dato a tener en cuenta cuando se realicen simulaciones.

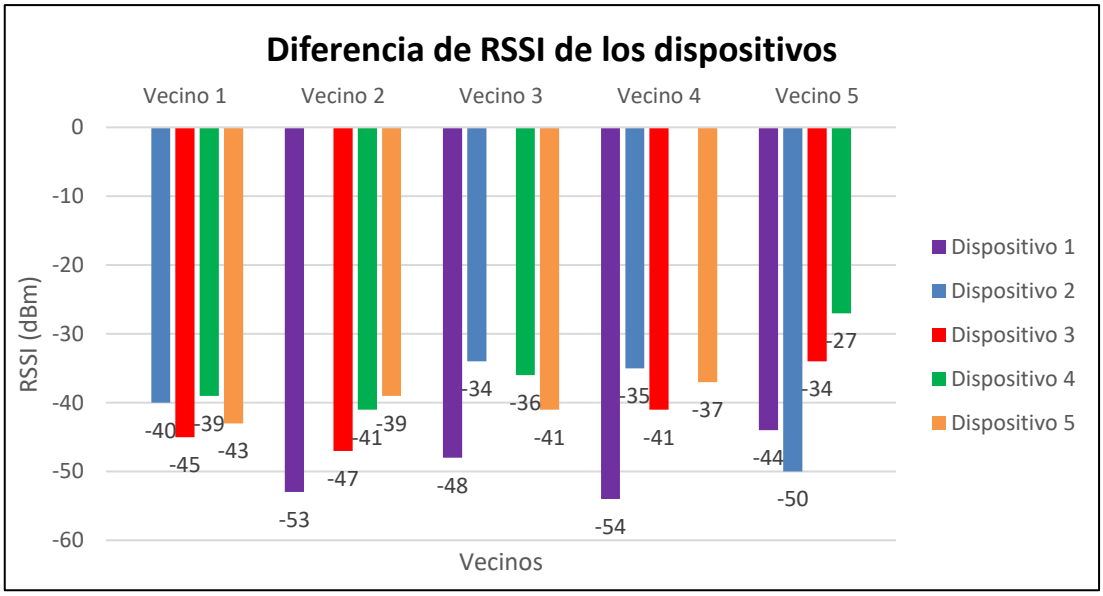


Figura 73. Diferencia del valor de RSSI de un entorno simulado a uno real.

En cuanto al DSR de los paquetes de *report*, en el entorno simulado el DSR tomaba un valor del 100% en todos los casos, mientras que en el entorno real estos valores llegaban a disminuir hasta un 40% menos que el caso anterior, hasta alcanzar el valor de 60% al estabilizarse. Se observa que la exactitud con la que llegan los paquetes no es la misma pero no sufren un retraso grave.

En cuanto a los paquetes de datos, se va a ver la diferencia tanto del DSR como del PDR con el *deadline* de 100 ms. En el caso del PDR en ambos casos es del 100%, lo que indica que los dispositivos funcionan correctamente sin producirse pérdidas en ninguno de los dos escenarios. En el caso del DSR, Figura 74, este sufre mayores variaciones, alcanza una diferencia del 20% entre los valores obtenidos en la simulación y los del entorno real. Como el valor del PDR es del 100%, esta diferencia del DSR indica que todos los paquetes se han transmitido sin sufrir pérdidas, pero aproximadamente un quinto de ellos llega más tarde de lo establecido.

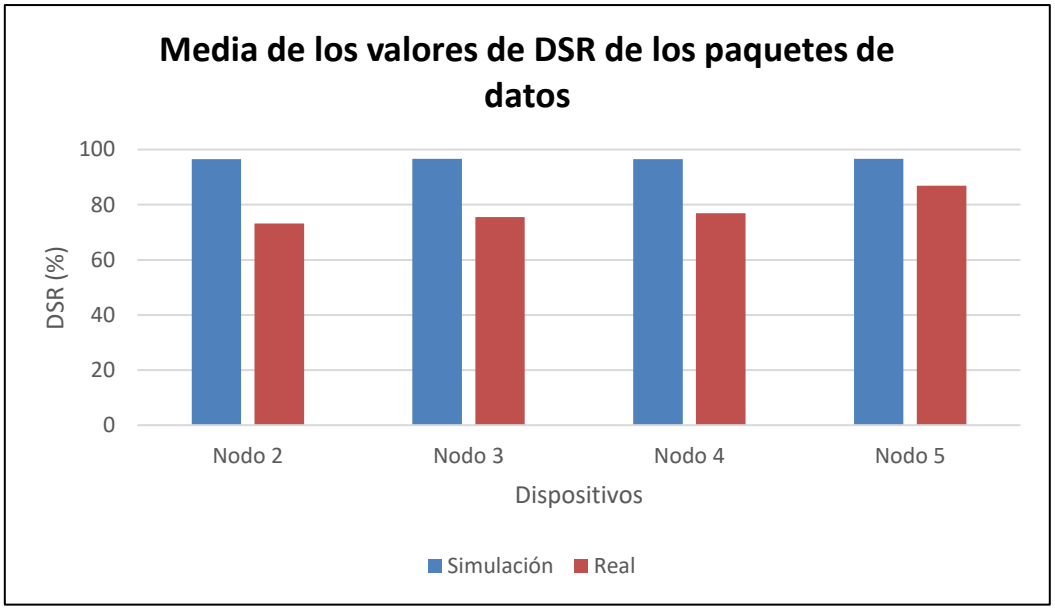


Figura 74. Comparación de la media del valor de DSR de los paquetes de datos.

9.2.2 Resultados de las mejoras añadidas

Se va a realizar la comparación de los mensajes de *report*, comparando los valores de DSR. Como se observa, Figura 75, el implementar o no la mejora en la movilidad ha mejorado considerablemente los valores obtenidos. En el caso de tener un nodo móvil sin dicha mejora, los valores medios de DSR no llegaban a alcanzar ni el 50%, quedándose la media por debajo del 20%. La excepción es el nodo 1, el *sink*, ya que está conectado directamente.

En el caso de aplicar la mejora en la movilidad, se puede observar cómo los valores medios se encuentran por encima del 60%. Estos valores se asemejan mucho más a los obtenidos cuando los nodos no se encontraban en movimiento, lo que implica una mejora considerable.

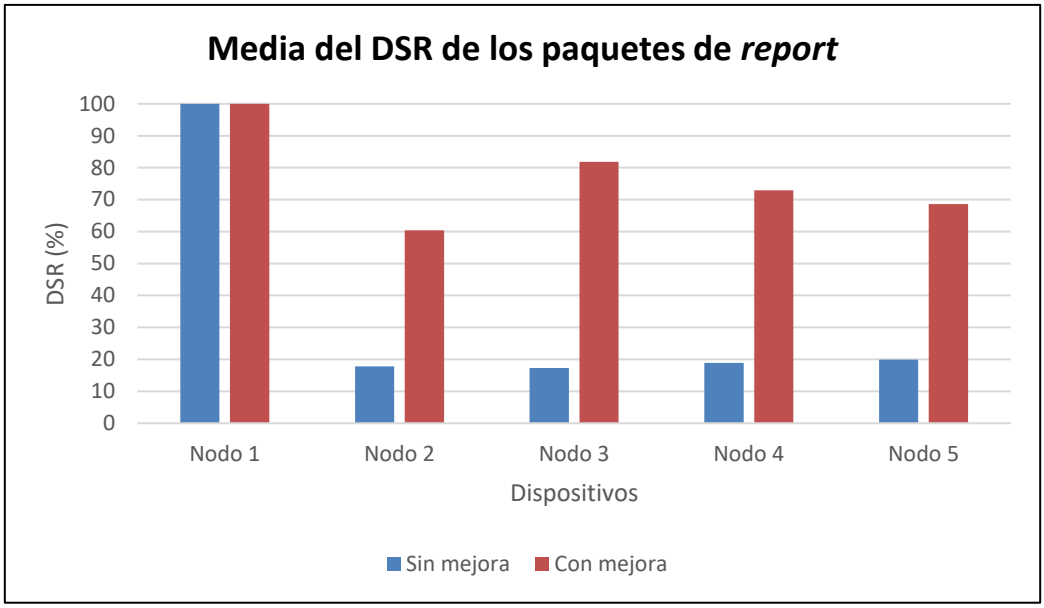


Figura 75. Comparación de la media del DSR de los paquetes de *report* con movilidad.

En cuanto a los paquetes de datos, se va a comparar tanto el PDR como el DSR. Empezando por el PDR, Figura 76, este valor es muy importante ya que refleja si se han sufrido pérdidas o no. En el caso de tener un nodo móvil sin ninguna mejora, se reflejan algunas pérdidas llegando a perderse hasta un 2% de los paquetes, lo que implica un empeoramiento en la calidad de servicio de las comunicaciones.

En el caso de añadir las mejoras en la movilidad, se observa cómo las pérdidas son nulas, obteniendo valores del 100% en todos los dispositivos, lo que implica que se reciben todos los paquetes.

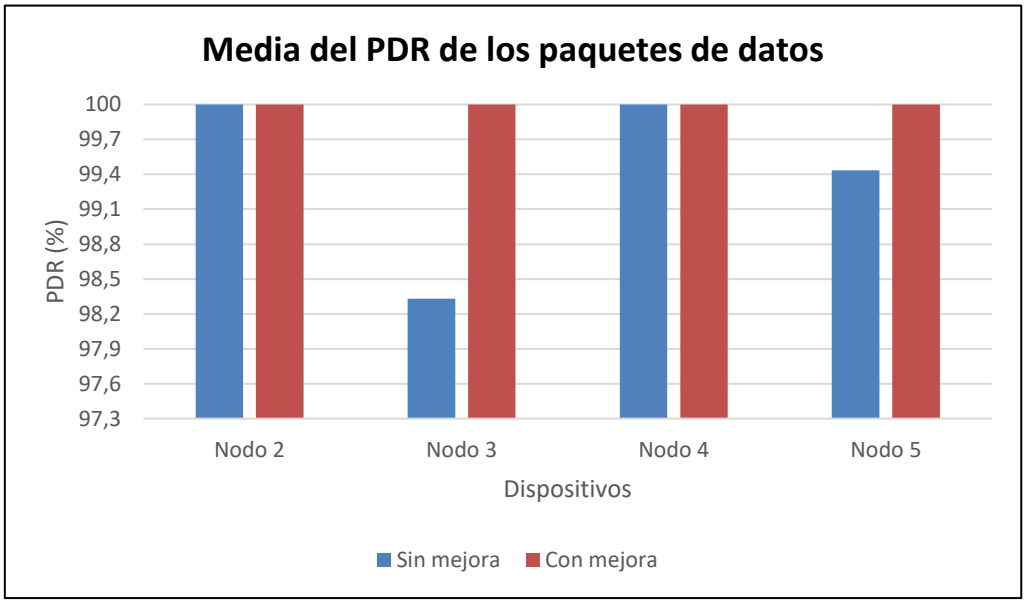


Figura 76. Comparación de la media del PDR de los paquetes de datos con movilidad.

En el caso del DSR, Figura 77, en la prueba realizada sin la mejora, los valores no llegan a estabilizarse hasta alcanzar un valor aproximado del 50%. Esto implica que la mitad de los paquetes llegan fuera del periodo establecido. Esto puede ser debido a la alta carga en los nodos y por lo tanto pueden saltarse algún slot y transmitir los paquetes tarde.

Al aplicar las mejoras en la movilidad, se puede observar cómo los valores medios aumentan y se encuentran entorno al 55% y el 75%. Por lo tanto, dichos valores han mejorado y ahora, de los paquetes que llegan, solo un tercio de ellos lo hace fuera del periodo establecido. Esto supone que los nodos tienen una menor carga y por lo tanto se saltan menor número de slots.

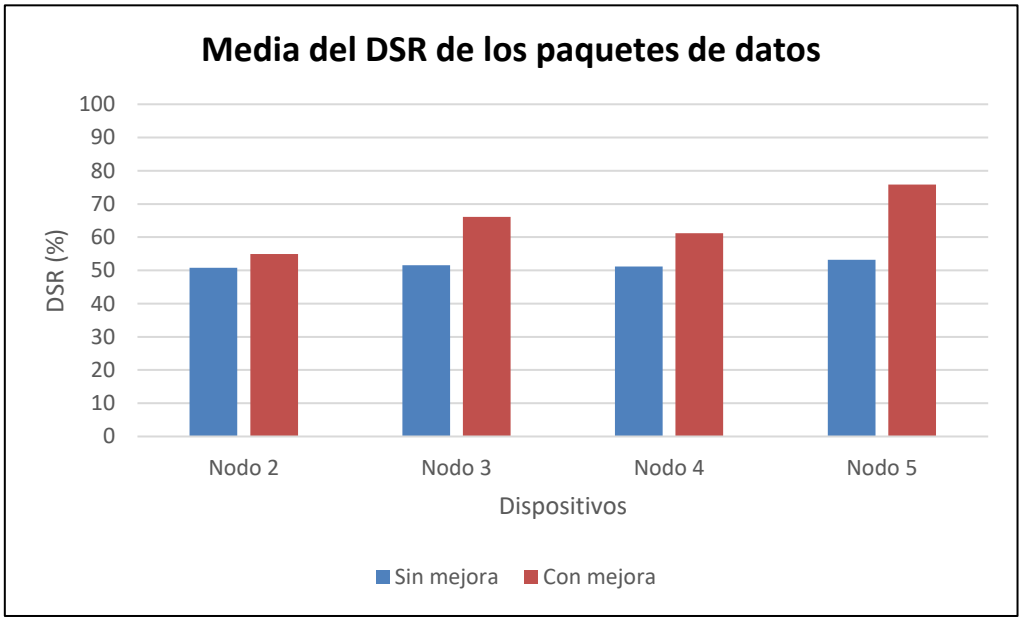


Figura 77. Comparación de la media del DSR de los paquetes de datos con movilidad.

Esta comparación demuestra que el aplicar la mejora en la movilidad permite reducir la carga de los nodos fijos, permitiendo recibir todos los paquetes sin que se lleguen a producir pérdidas. También se han mejorado los tiempos de recepción, ya que sin la mejora la mitad de los paquetes llegaban fuera del periodo establecido. Con la nueva mejora solo llegan entre un tercio y un cuarto de los paquetes fuera del periodo establecido.

Esta nueva mejora permite una comunicación más robusta entre los dispositivos y por lo tanto mejora las calidades de servicio.

Capítulo 10. Conclusiones y trabajos futuros

10.1 Conclusiones

Como se ha podido observar a lo largo del trabajo, las redes SDN presentan grandes ventajas en la Industria 4.0, ya que permiten la integración de múltiples tecnologías. En entornos industriales es muy importante tener en cuenta la multitud de tecnologías que se llegan a emplear, ya que se dispone de múltiples redes. SDN permite la integración de dichas redes, aportando una separación del plano de datos y el plano de control. De esta forma se obtiene un control más centralizado capaz de gestionar dichas redes, como se ha estudiado en el trabajo.

En este trabajo se ha realizado una red SDN, en primer lugar, configurando y gestionando una red cableada. A continuación, se ha configurado y gestionado una red inalámbrica de sensores IWSN mediante SDN-WISE. Se han añadido nuevas funcionalidades al controlador SDN-WISE, permitiendo una mayor gestión de la red en el punto de control.

Seguidamente, se ha estudiado la posibilidad de que uno de los nodos inalámbricos se encontrara en movimiento. Para ello, se ha añadido una nueva funcionalidad a la aplicación que permita al nodo móvil conectarse al nodo con mejor señal conforme se vaya moviendo.

Finalmente se ha realizado la integración de ambas redes inalámbricas en el controlador para obtener una gestión centralizada.

Con la realización del trabajo descrito se han podido obtener las siguientes conclusiones:

- El empleo de colas en las redes cableadas SDN y del protocolo TSCH en las redes inalámbricas SDN permite gestionar distintos tráficos, aportando mayor calidad de servicio. Esto es muy importante a tener en cuenta en entornos industriales porque se dispone de múltiples tecnologías en las redes y se pueden gestionar sus tráficos dependiendo de las necesidades y prioridades de cada una para evitar fallos.
- El empleo de los dos controladores permite tener una gestión más controlada y mayor fiabilidad, de esta forma trasladando el estudio a un entorno industrial, el controlador de la red cableada ONOS se puede emplear para la gestión del tráfico entre múltiples factorías y el controlador de la red inalámbrica SDN-WISE para el control de las múltiples redes inalámbricas dentro de una misma factoría.
- El empleo de la nueva funcionalidad para nodos móviles permite un ahorro en todos los nodos. También permite una gestión de los nodos más eficaz, mejorando las comunicaciones de los dispositivos. Finalmente permite una comunicación más robusta al tener en cuenta en mayor profundidad el valor de RSSI.

10.2 Trabajos futuros

Como posibles trabajos futuros para desarrollar serían los siguientes:

- Existe la posibilidad de integrar ambos controladores, ONOS y SDN-WISE, para simplificar el control. Esta unión se puede llevar a cabo desarrollando una API REST que permita pasar la topología de un controlador a otro y que todos los flujos se creen desde un solo controlador. En este trabajo se ha decidido dejar los controladores separados ya que aportan una mayor fiabilidad en caso de fallos en la red.
- Para reducir la carga del *sink*, se pueden añadir nuevas mejoras que permita formar un *sink* con varios dispositivos, *multisink*. De esta forma, se reduciría la carga y los flujos se establecerían con el dispositivo *sink* con menor carga.
- Finalmente, dado que se tiene una red totalmente reconfigurable se puede emplear ML (*Machine Learning*) o aprendizaje automático junto con IA (Inteligencia Artificial) para automatizar algunas de las tareas a realizar en la red, como la configuración de las rutas en condiciones complejas. De esta forma se pueden mejorar los procesos productivos en el entorno industrial obteniendo grandes beneficios.

Capítulo 11. Bibliografía

[1] José Enrique Álvarez, "Asegurando el IoT: nuevas directrices para conseguir una cadena de suministros de productos y soluciones IoT segura", smartLIGHTING, 2020, [Online]. Disponible: https://smart-lighting.es/seguridad-cadena-suministros-iot-enisa/

[2] Paloma Recuero de los Santos, "Breve historia de Internet de las cosas (IoT)", Telefónica Tech AI of Things, 2020, [Online]. Disponible: https://empresas.blogthinkbig.com/breve-historia-de-internet-de-las-cosas-iot/

[3] Mehmet Baygin, Hasan Yetis, Mehmet Karakose, Erhan Akin, "An effect analysis of industry 4.0 to higher education", IEEE, 2016.

[4] Rioja2, "Industria 4.0: Indagando en la historia moderna", rioja2, 2017, [Online]. Disponible: https://www.rioja2.com/n-115584-2-industria-40-indagando-en-la-historia-moderna/

[5] Ramón Jesús Millán Tejedor, "Qué es… SDN (Software-Defined Networking)", COIT & AEIT, 2016.

[6] ONF, "Software-Defined Networking (SDN) Definition", Open Networking Foundation, [Online]. Disponible: https://opennetworking.org/sdn-definition/

[7] Ramón Jesús Millán Tejedor, "SDN: el futuro de las redes inteligentes", Conectrónica, 2014.

[8] Network World, "Las SDN serán vitales para el Internet de las Cosas", COMPUTERWORLD, 2014, [Online]. Disponible: https://www.computerworld.es/telecomunicaciones/las-sdn-seran-vitales-para-el-internet-de-las-cosas

[9] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig "Software-Defined Networking: A Comprehensive Survey", IEEE, 2014.

[10] Juan Carlos Chico, David Mejía, Iván Bernal, "Implementación de un Prototipo de una Red Definida por Software (SDN) Empleando una Solución Basada en Hardware", JIEE, 2014.

[11] William Stallings, "Software-Defined Networks and OpenFlow", The Internet Protocol Journal, 2013.

[12] Team Tesca, "What is Wireless Sensor Network, and Types of WSN?", TESCA, 2021, [Online]. Disponible: https://www.tescaglobal.com/blog/what-is-wireless-sensor-network-and-types-of-wsn/

[13] Aprendiendo Arduino, "Conectividad IoT", Aprendiendo Arduino, [Online]. Disponible: https://aprendiendoarduino.wordpress.com/tag/6lowpan/

[14] Francisco Martín Archundia Papacetzi, "El estándar IEEE 802.15.4", BIBLIOTCAS UDLAP, [Online]. Disponible: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf

[15] CCNA, "Conozca mejor el protocolo de red IEEE 802.15.4", CCNA, [Online]. Disponible: https://ccnadesdecero.es/protocolo-de-red-ieee-802-15-4/

[16] Juan Camilo Pimienta Gómez, Juan Pablo Méndez Perdomo, Simón Dávila Saravia, Gabriel Gómez Corredor, "IEEE 802.15.4, Redes de Sensores Inalámbricas -WSN y Smart campus", IEEE, 2021.

[17] Federico Orozco-Santos, Víctor Sempere-Payá, Teresa Albero-Albero, Javier Silvestre-Blanes, "Enhancing SDN WISE with Slicing Over TSCH", Sensors, 2021.

[18] Manjeet Dhariwal, "OPENKILDA: LESSONS LEARNED FROM ONOS & ODL", CloudSmartz, 2019.

[19] Farzaneh Pakzad, "COMPARISON OF SOFTWARE DEFINED NETWORKING (SDN) CONTROLLERS. PART 2: OPEN NETWORK OPERATING SYSTEM (ONOS)", aptira, [Online]. Disponible: https://aptira.com/comparison-of-software-defined-networking-sdn-controllers-part-2-open-network-operating-system-onos/

[20] ONF, "ONOS", Open Networking Foundation, [Online]. Disponible: https://opennetworking.org/onos/

[21] ONF, "ONOS: Open Network Operating System", Open Networking Foundation, [Online]. Disponible: https://github.com/opennetworkinglab/onos

[22] SDN-WISE lab, "The stateful Software Defined Networking solution for the Internet of Things", SDN-WISE, [Online]. Disponible: https://sdnwiselab.github.io/

[23] Mininet, "Mininet", Mininet Project Contributors, [Online]. Disponible: http://mininet.org/

[24] Mininet, "Mininet", Mininet Project Contributors, [Online]. Disponible: https://github.com/mininet/mininet

[25] Programador clic, "Aprendizaje SDN / NFV --- Mininet --- 1. ¿Qué es Mininet?", programador clic, [Online]. Disponible: https://programmerclick.com/article/93222162186/

[26] Sergio Luis Garí Santesteban, "Selección del Sistema Operativo Contiki y el Simulador COOJA", [Online]. Disponible: https://1library.co/article/selecci%C3%B3n-sistema-operativo-contiki-simulador-cooja.zkwxd1ez

[27] Edinburgh Napier University, "Cooja Simulator Manual Version 1.0", IoT Networking Research Group, 2016.

[28] Contiki-os, "The Contiki Operating System", The Contiki Open Source OS for the Internet of Things, [Online]. Disponible: https://github.com/contiki-os/contiki

[29] Simon Duquennoy, "Contiki-ng", Contiki-ng, [Online]. Disponible: https://github.com/contiki-ng/contiki-ng/wiki

[30] Damián A., "Grafana, un software de código abierto para análisis y supervisión", Ubuntulog, [Online]. Disponible: https://ubunlog.com/grafana-software-analisis-supervision/#Caracteristicas_generales_de_Grafana

[31] Grafana Labs, "Install Grafana", Grafana, [Online]. Disponible: https://grafana.com/docs/grafana/latest/installation/