PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

PROPUESTA DE GESTIÓN DE REDES COMUNITARIAS DE ACCESO DE ÚLTIMA MILLA, BASADA EN SDN COMO COMPLEMENTO A LA RED REGIONAL DE FIBRA ÓPTICA (RRFO) Y LA RED DORSAL NACIONAL DE FIBRA ÓPTICA (RDNFO) EN EL PERÚ

Tesis para obtener el título profesional de Ingeniero Electrónico

AUTOR:

Cristian Martin Narvaez Diaz

ASESOR:

M.Sc. Ing. Gumercindo Bartra Gardini

Lima, Mayo, 2024

Informe de Similitud

Yo, GUMERCINDO BARTRA GARDINI, docente de la Facultad de CIENCIAS E INGENIERÍA de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada: PROPUESTA DE GESTIÓN DE REDES COMUNITARIAS DE ACCESO DE ÚLTIMA MILLA, BASADA EN SDN COMO COMPLEMENTO A LA RED REGIONAL DE FIBRA ÓPTICA (RRFO) Y LA RED DORSAL NACIONAL DE FIBRA ÓPTICA (RDNFO) EN EL PERÚ, del autor: Cristian Martin Narvaez Diaz,

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 11%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 18/04/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: 25/04/2024 ……

Apellidos y nombres del asesor / de la asesora: BARTRA GARDINI, GUMERCINDO	
DNI: 07231977	Firma
ORCID: https://orcid.org/0000-0002-4307-1285	

Resumen

Las redes de acceso de última son aquellas que comunican las compañías operadoras con los clientes finales en zonas aledañas. Sin embargo, por falta del desarrollo de estas redes en el país, no se suelen implementar. Esta falta de desarrollo se debe a que su implementación conllevaría una mayor demanda del tráfico de datos en la red, lo cual generara una gestión lenta de los dispositivos y también, debido a que no posee un sistema de control que permita cambiar su comportamiento a una manera dinámica flexible.

El estudio pretende presentar los diseños de plataforma de gestión programable basada en SDN para una red de acceso de última milla, la cual permitirá una gestión adecuada de los diferentes equipos que conforman una red. Software Defined Networking son redes emergentes que permite un control centralizado, pues el controlador se encuentra en un solo punto de la red y puede ejecutar acciones desde aquí, según las necesidades de la red. Esto provee a las redes una mayor eficiencia y, por tanto, permitirá agilizar la administración de estas, ya que el controlador al estar centralizado contará con un conocimiento más amplio de la totalidad de la red, lo que facilitará una utilización más eficiente de sus recursos, posibilitando así su flexibilidad y escalabilidad en el ámbito de los datos.

El diseño presentado tendrá como alcance a ser un test bed con entornos de simulación a través de emuladores de red como VNRT o Mininet, brindados por los Grupos de Investigaciones de Red Avanzada de PUCP. Sin embargo, se espera que, para una implementación en una red real, pueda realizarse sin ningún problema, gracias a las características de estos emuladores que permiten que, al pasar a un plano real, el código y la configuración de la red no cambien en lo absoluto.

Índice

Índice de figuras

Introducción

En la actualidad el Perú no cuenta la implementación de red de accesos de ltimas millas, las cuales conectan las zonas alejadas de capitales de cada distrito con la compañía operadora y que, además, podrían servir como un complemento a Redes Dorsales Nacionales de Fibras Ópticas (RDNFO) y Redes Regionales de Fibras Ópticas en el Perú. La razón por la cual no se contemplan estas redes, es debido a que en estas zonas al no poseer un gran número de habitantes no genera rentabilidad para las operadoras móviles o de internet, ya que estas necesitan acceder a la administración de los servicios que proveerán a los consumidores, lo cual resulta ser muy costos. Por eso, surge la iniciativa de producir una manera más fácil y menos costosa de administrar las redes de acceso de última milla. Por otro lado, un nuevo paradigma de networking que está actualmente siendo una opción en el área de las redes que busca centralizar el control es la red definida por softwares (SDN). Esta novedosa perspectiva posibilita trasladar la gestión de los equipos de red desde sus propios dispositivos hacia un controlador centralizado. Este controlador, al contar con un conocimiento integral de la red, facilita una utilización más eficaz de sus recursos, lo que a su vez permite lograr flexibilidad y escalabilidad en el ámbito de los datos. Por todo lo mencionado antes, la investigación consiste en propuestas de gestión de redes comunitarias de acceso de última milla, basada en SDN como complemento a la Red Regional de Fibras Ópticas (RRFO) y a la Red Dorsales Nacional de Fibras Ópticas (RDNFO) en Perú.

Capítulo 1, se define el contexto y exigencia de plataformas de gestiones para redes. También trabajos relacionados con esta tesis que permitirán ilustrar las herramientas que se requieren para el diseño de redes de acceso basado en SDN. Finalmente, se establecen objetivo y justificación del estudio.

Capítulo 2, definido por conceptos que requieren conocer para desarrollo de esta tesis. Se define los distintos tipos de tecnologías que se podrían utilizar para los diseños de las redes de accesos y distintas topologías. Así mismo, la red definida por softwares, sus arquitecturas, las interfaces, los protocolos que utiliza y los parámetros a medir para una buena gestión de red.

En el Capítulo 3, se plantea un modelo solución a modo de la simulación de una topología como prueba del correcto funcionamiento del controlador. Todo esto para poder explicar las operaciones de controladores, detallándose el mecanismo que implementara.

Finalmente, capítulo 4, presentan las distintas pruebas, simulaciones y resultados obtenidos con el fin de evaluar la operatividad de los componentes clave del controlador, así como el enfoque de gestión adoptado para medir la capacidad de expansión de la red.

1. Marco Problemático

1.1. Administración de la red de accesos de últimas millas en el Perú.

El MTC suscribió un acuerdo con la compañía Aztecas Comunicación Perú S.A.C. en la fecha del 17 de junio del 2014; para desarrollar el diseño de Redes Dorsales Nacionales de Fibras Ópticas (RDNFO)[1]. Esta infraestructura comprende la instalación y funcionamiento de extensas redes de fibras ópticas, abarcando cerca de 13 000km y conectando Lima con la 22 capital de regiones, así como 180 capital de las provincias (Fig. 1)[2].

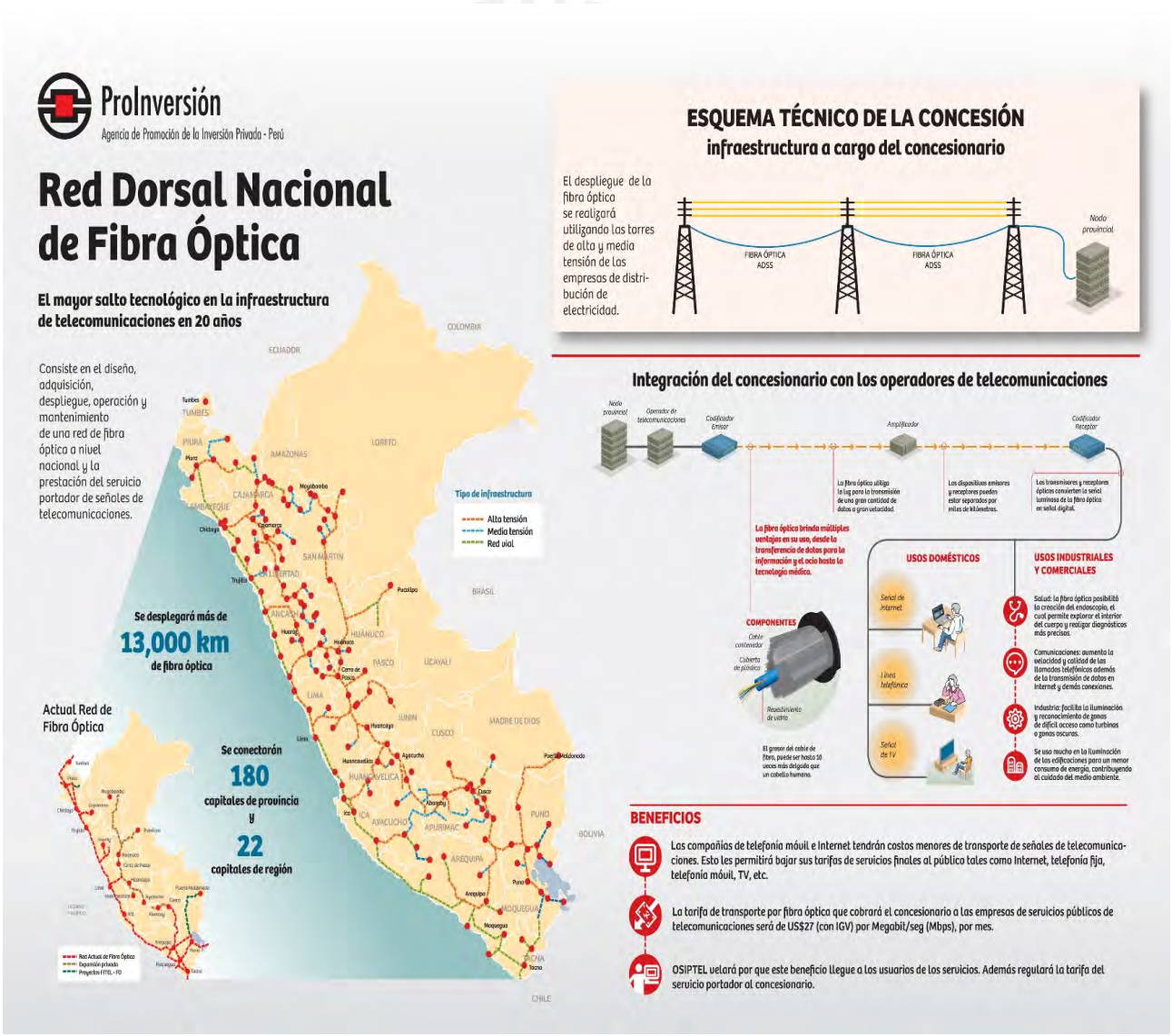

Figura 1: Alcance del proyecto de construcción de la Redes Dorsales Nacionales de Fibras Ópticas en Perú [2].

Los beneficios generados por el proyecto, fueron la disminución de costos de transportes de señal de telecomunicación para las compañías de telefonía móvil e Internet [2]. Como complemento a esta red, se diseñó la Red Regional de Fibra Óptica en el Perú, la cual consta en la conexión de la capital de la provincia con las capitales de los distritos que la conforman. Esta provee Internet con bandas anchas a 1500 distritos, permitiendo que miles de personas elevan sus productividades y competitividades [3]. No obstante, existe una dificultad de acceso de la red a las zonas rurales aledañas a las capitales de cada distrito, como por ejemplo Pueblo Nuevo Colán distrito Colán, San Pedro de Hualla ubicado el distrito de Hualla, etc. Esto se debe a la falta de desarrollo de redes de accesos de últimas millas, necesarias para conectar la compañía operadora a las zonas lejanas. La razón por la cual no se contemplan estas redes, es debido a que en estas zonas al no poseer un gran número de habitantes no genera rentabilidad para las operadoras móviles o de internet, puesto que estas necesitan acceder a la administración de los servicios que proveerán a los consumidores, lo cual resulta ser muy costoso [4].

Por ende, se puede deducir como problemática central la imposibilidad en la administración de las redes de accesos de últimas millas debido a su costo. Entre las causas principales que la generan, se encuentra el incremento de clientes. Los cuales, generan una mayor demanda de tráficos de dato en las redes. Otro factor, que se suele presentar es la gestión lenta de los dispositivos. Esto se debe a que, en una red conformada por diferentes dispositivos, algunos de ellos no logran interrelacionarse con otros, lo cual, ocasiona que estas máquinas no generen una buena eficiencia cuando trabajen en conjunto [5]. También se debe a la poca flexibilidad que posee la red por ausencia de sistemas de controles que permitan cambiar su comportamiento a una manera automatizada y dinámica [5]. Presentándose como consecuencias, falta de soporte local para los consumidores. Además, genera reclamos por parte de los clientes por el mal servicio, lo cual, puede ocasionar la pérdida de los clientes [6]. Por último, exige una inversión costosa para las compras y mantenimientos de equipo de alta tecnología para solucionar la mala administración [6]. Ante esta problemática, se necesita una plataforma de gestión óptima que de soporte y ayude en la administración de las redes de accesos de últimas millas para todas las zonas rurales sin conexión a la red.

1.2. Estado del arte

Un nuevo paradigma de networking que está actualmente siendo una opción en el área de las redes que busca centralizar el control son la red definida por softwares (SDN). Este enfoque innovador posibilita la eliminación del control directo en los dispositivos de red, trasladándolo a un componente central conocido como controlador, el cual posee un conocimiento integral de toda la red, lo que facilita una optimización más efectiva de sus recursos, brindando así flexibilidad y escalabilidad en el ámbito de los datos [7]. El surgimiento de este paradigma se debe, como se ve en la Figura 2, a la tendencia de implementarse más funcionalidad al Plano Datos en el hardware, poniendo el software al Plano de Control [8]. También se debe a la exigencia de reducirse los costos de equipo, facilitar su mantenimiento, brindándole una mayor flexibilidad, y simplificar su complejidad [8].

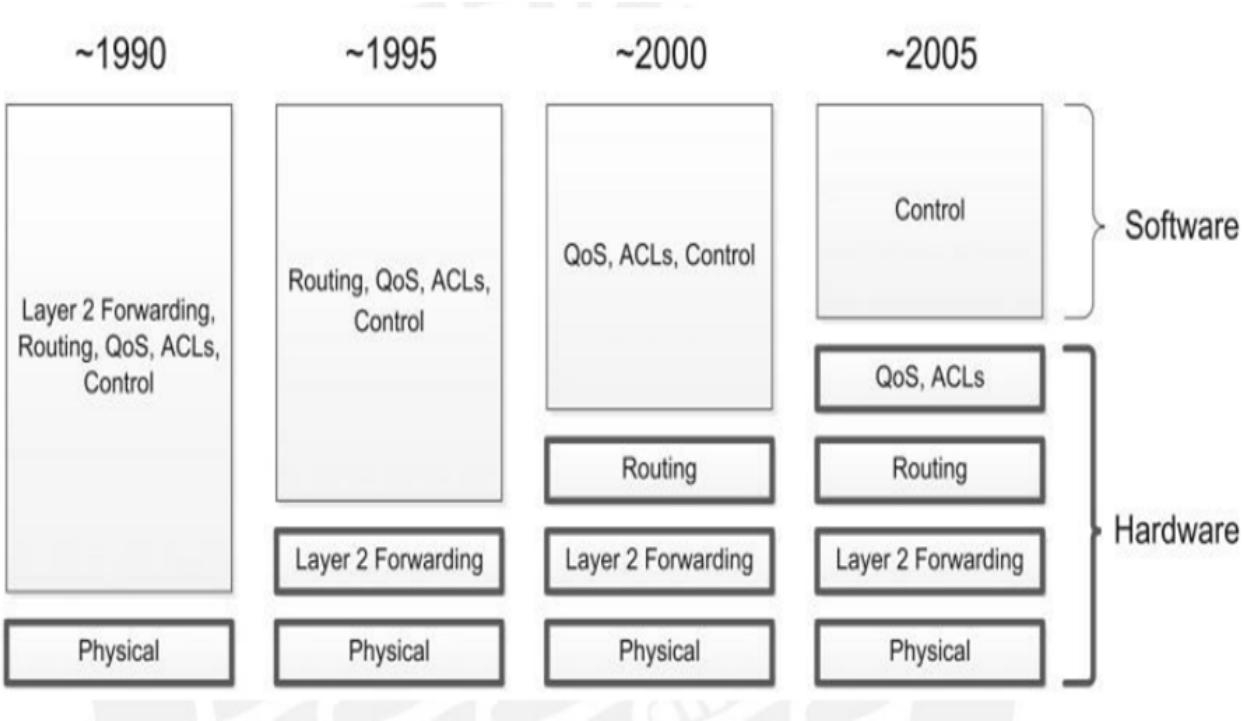

Figura 2: Evolución de la tecnología SDN [9].

Describiendo su arquitectura brevemente, como se observa en Fig. 3, constan de 3 capas: la de aplicaciones, control e infraestructura. En la primera capa, las aplicaciones SDN interactúan con el controlador para lograr una función de red específica con finalidad de complacer la necesidad de los operadores de red [10]. Estas establecen comunicación con la capa de control mediante interfaces de programación de aplicaciones (API) en dirección norte, informándole a dicha capa acerca de los recursos requeridos por las aplicaciones y su destino [11]. En la segunda capa se encuentra el controlador SDN, que son los software que proporcionan unas vistas y controles centralizado de todas las redes [11]. Finalmente, en la tercera capa se encuentran los enrutadores y conmutadores que por medio de la interfaz openflow indica qué rutas debe tomarse el dato según decididos por los controladores [11].

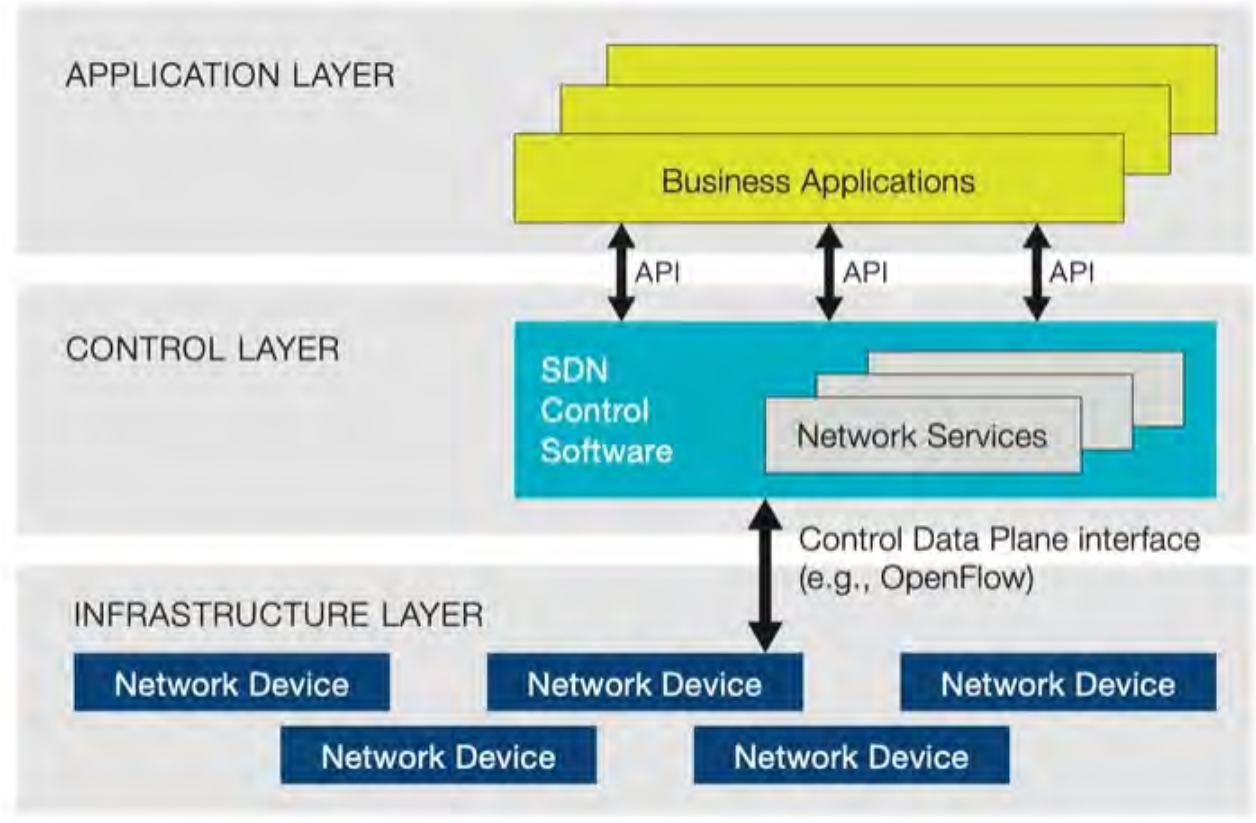

Figura 3: Arquitectura del SDN [11].

1.2.1. Gestión de redes de fibras óptica GPON, basado en SDN

La conexión GPON funciona con dispositivos que se encuentra en centralita del operador llamados OLT (Opticales Lines Terminales) y el dispositivo que se encuentra en los hogares llamados ONT (Opticals Nodes Terminales) o también ONU (Opticales Network Units) [12]. Es necesario intercalar divisores de fibra, denominados splitters, entre ambas, cuya función se limita a reunir o separar las distintas fibras. De este modo, se logra la subdivisión progresiva de las fibras en líneas adicionales hasta llegar al domicilio del usuario, donde se instala el ONT/ONU (ver Figura 4))[12].

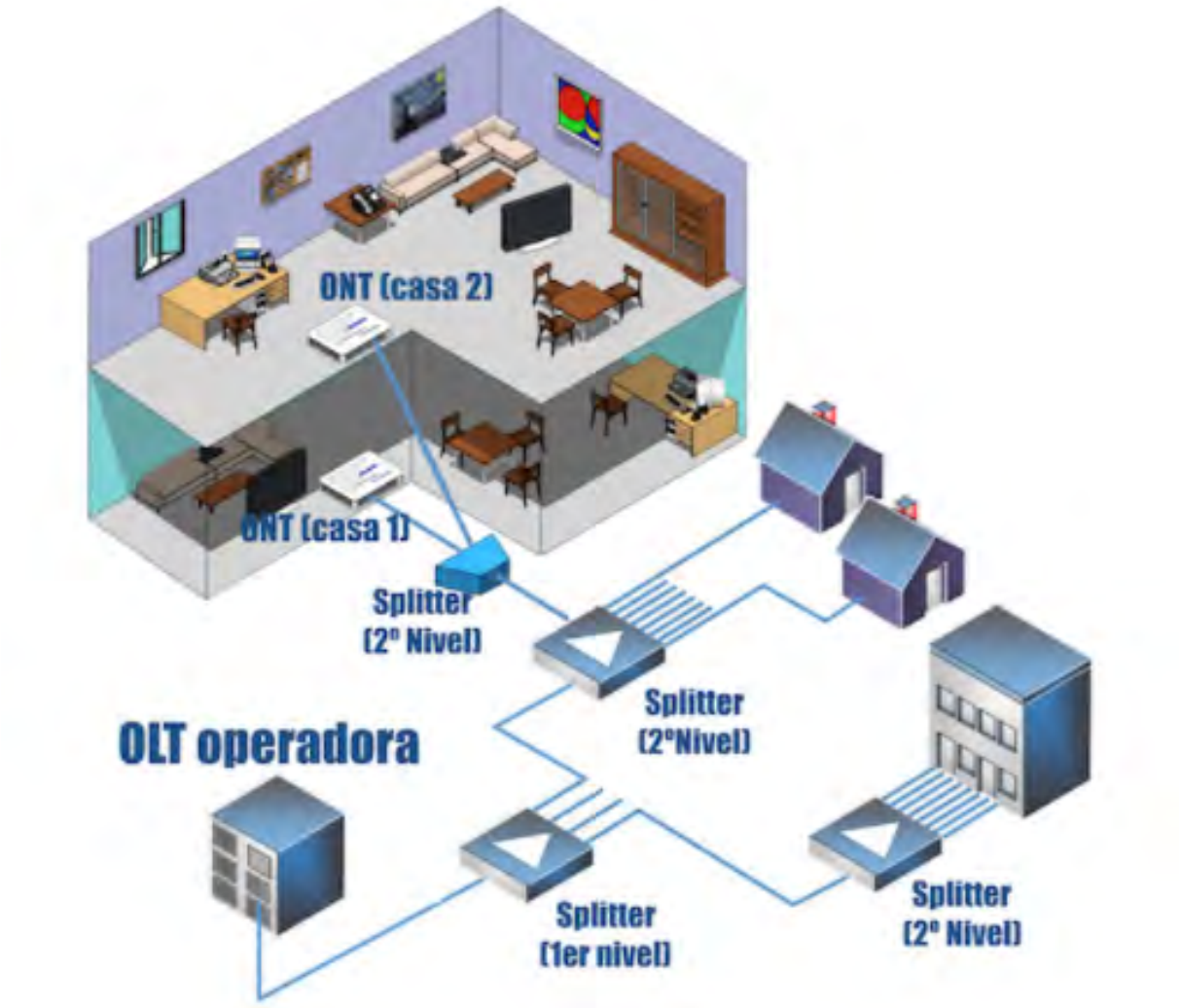

Figura 4: Arquitectura de una conexión GPON[12].

Para introducir el paradigma SDN en el área de GPON, se desarrolló un nuevo protocolo que permite que los dispositivos GPON se conviertan en unidades programables [13]. Por ello, se realizó una solución centrada en OpenFlow, el cual es el protocolo basado en SDN. Este nuevo se llama OpenFlow PLUS y permite que cada dispositivo (OLT, ONT) dentro del árbol GPON sea programable [13]. El Openflow Plus posibilita la comunicación tanto entre la ONT como OLT hacia un controlador, el cual accede a la ONT por medio de la OLT. Debido a esto, se pueden optimizar al máximo las tablas de flujo que poseen cada uno de estos componentes para que incrementen su funcionalidad y velocidad para cada función que vaya a realizar: reenvío de paquetes, control dinámico de ancho de banda y recopilación de datos estadísticos (ver Figura 5) [13].

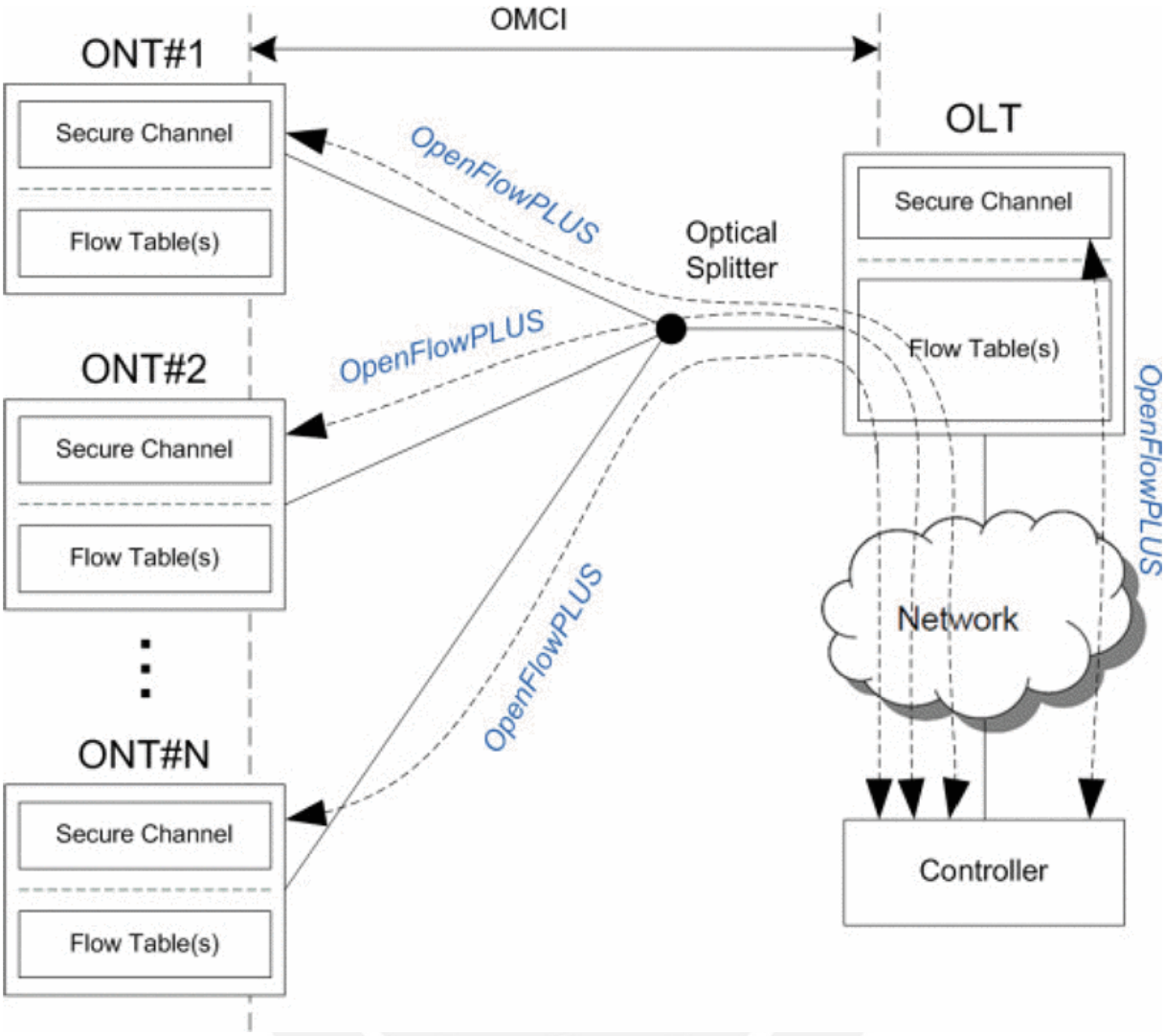

Figura 5: Descripción general de la solución GPON basada en OpenFlow Plus [13].

1.2.2. Gestión de una red Inalámbrica, basada en SDN

Para introducir SDN en red inalámbrica, se desarrolló los diseños de redes WLAN basadas en arquitectura de esta. La arquitectura de la red inalámbrica se segmenta en 03 capa: capas de infraestructura, capas de control y capas de aplicación de la misma manera que la arquitectura de la SDN. La capa de infraestructura está compuesta por AP y conmutadores OpenFlow, la capa de control está compuesta por el controlador SDN y capa de aplicación está compuesta por aplicaciones realizadas por programación independiente [14]. La arquitectura se centra en la capa de control. La capa de control interactúa con capas de infraestructura mediante la interfaz SouthBound y realiza la interacción de datos con capas de aplicaciones mediante interfaz NorthBound (ver Figura 6) [14]. Al generar que la arquitectura de esta red se centralicen en el controlador, permitiendo una perspectiva global, se puede resolver eficazmente el problema del

traspaso de terminales de la red sin interrupciones, para que, de esta manera, se pueda optimizar la QoS de las redes y mejorar la utilización de sus recursos [14]. Por último, genera un método de traspaso con equilibrio con recopilaciones, control y análisis de informaciones de la red [14].

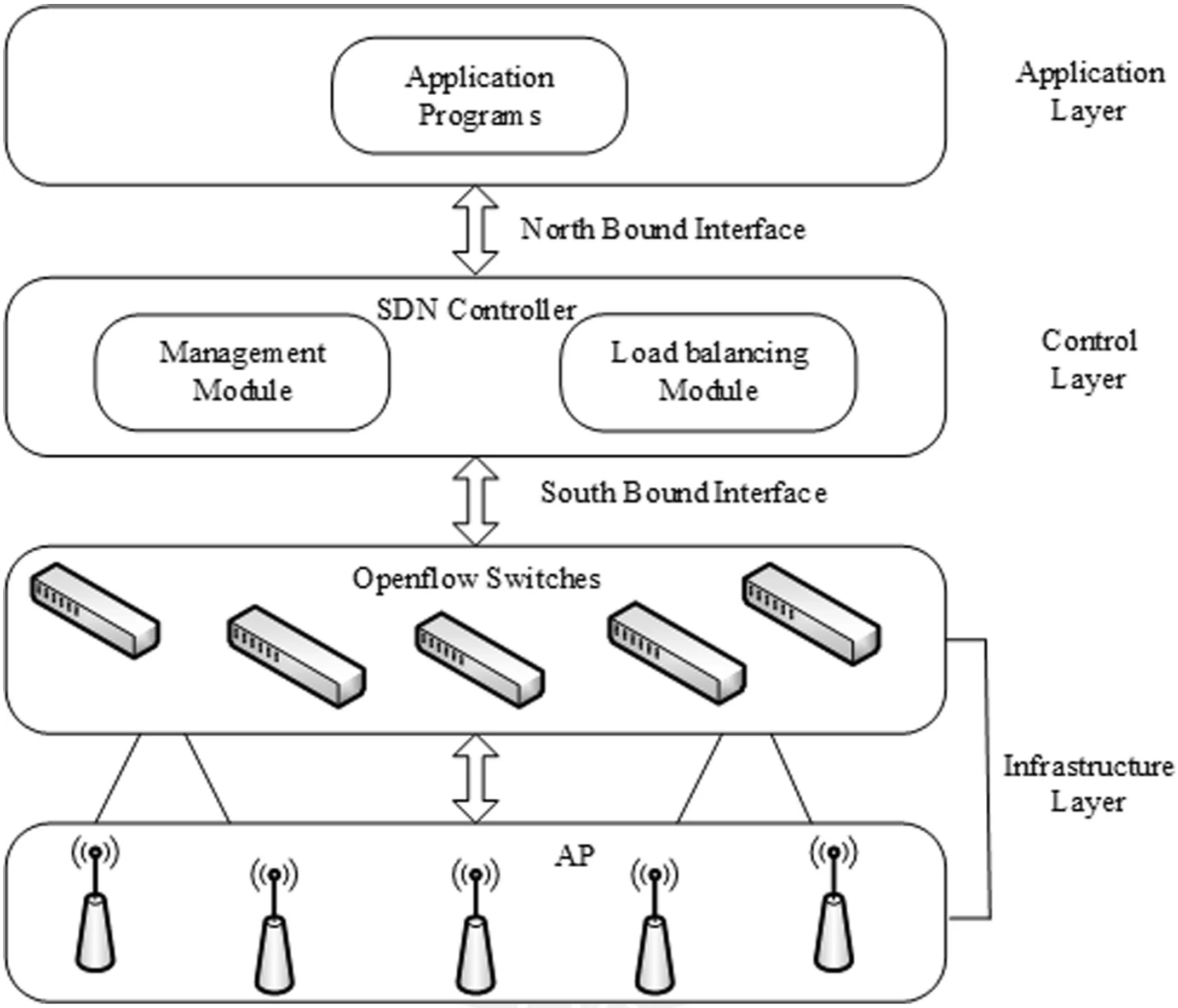

Figura 6: Arquitectura WLAN basada en SDN [14].

1.3. Justificación

Los estudios previos demuestran que la tecnología SDN, al permitir que su administración se realice desde un controlador centralizado, haciendo más fácil su monitorización, funciona como la mejor solución para la problemática central, la cual es la imposibilidad en la administración de las redes de accesos de últimas millas debido a los costos.

Entre las características más relevantes en utilizar esta tecnología:

- Aumenta la funcionalidad y velocidad de la red: Los recursos de la red como el ancho de banda y encaminamiento se pueden asignar de manera rápida para los nuevos servicios.
- Disminución de costos relacionados con la gestión de la red: Se pueden utilizar dispositivos de diferentes fabricantes. Debido a esto, se genera un ahorro en la compra de dispositivos que tengan que ser compatibles.
- Genera redes dinámicas y adaptables: Se pueden configurar en poco tiempo, puesto que evita la configuración de cada equipo en la red.

Sumando estas características más la administración centralizada, se justifica que SDN es la opción óptima para el diseño de plataformas de gestiones de redes de acceso comunitaria de última milla. Por ende, el desarrollo de la presente tesis aportará en el análisis de la tecnología SDN aplicada al área de gestión de redes, con la posibilidad de dar confiabilidad de empezar con generación de redes de acceso en el Perú.

1.4. Objetivos

1.4.1. Objetivo General

El objetivo general es diseñar la plataforma de gestión de las redes de acceso comunitarias de última milla basada en SDN para la conexión a la Red Regional de Fibra Óptica (RRFO) y Red Dorsal Nacional de Fibra Óptica (RDNFO), en el Perú.

1.4.2. Objetivos específicos

- Recopilar la demanda del tráfico de datos y acceso a Internet de los usuarios en las localidades aledañas a las capitales de distrito.
- Diseñar la Red de acceso comunitario de última milla con equipos que soportan SDN/OpenFlow.
- Simular y programar los elementos de la red basada en SDN.
- Simular la plataforma de Gestión de la Red de Acceso Comunitaria utilizando SDN.
- Realizar análisis y validación mediante los resultados obtenidos.

1.5. Alcances

- El presente proyecto de tesis plantea en diseñar una topología que funcione como un prototipo de una red de acceso de última milla de un distrito que posee algunas comunidades aledañas, en este caso se enfocara en la comunidad de pueblo de Colán ubicado en el distrito de Colán, y sobre las bases de topologías de las redes proponer un sistema de control basado en tecnología SDN que ayude en la administración de esta red por medio de la obtención de parámetros como la cantidad de hosts, rango de puertos, ancho de banda, tráfico de interfaz de red, etc.

2. Marco teórico

Este capítulo abarca la explicación del paradigma SDN, junto con los fundamentos necesarios para comprender las diversas tecnologías viables para implementar el diseño de las redes de accesos comunitarias de últimas millas.

2.1. Red de accesos inalámbrica

La red de acceso se refiere a aquellas que facilitan la conexión en la última etapa del trayecto de comunicación. En términos simples, son las infraestructuras que constituye las conexiones con las empresas proveedora y los clientes finales [4]. En contraste, la red inalámbrica es aquella que emplea señales de radios para establecer conexión entre dispositivos, prescindiendo de la necesidad de utilizar cables [15]. La transmisión y recepciones de datos de estas redes se ejecutan mediante antena, por todo esto, se puede concluir que una red de acceso inalámbrico es la red que conecta las compañías operadoras con clientes por medio de dispositivos que utilizan onda de radios para las transmisiones y recepciones de datos.

2.1.1. Tecnologías de la red inalámbrica

La red inalámbrica puede categorizarse en 04 grupos distintos conforme sus aplicaciones y alcances de las señales: red inalámbrica de áreas personales (WPAN - Wireless Personal Area Networks), redes inalámbricas de áreas local (WLAN - Wireless Local Area Networks), red inalámbrica de áreas metropolitanas (WMAN - Wireless Metropolitan Area Networks) y red inalámbrica de áreas amplias (WWAN - Wireless Wide Area Networks) [15]. Figura 7 se muestra la categorización de la principal tecnología empleada actualmente.

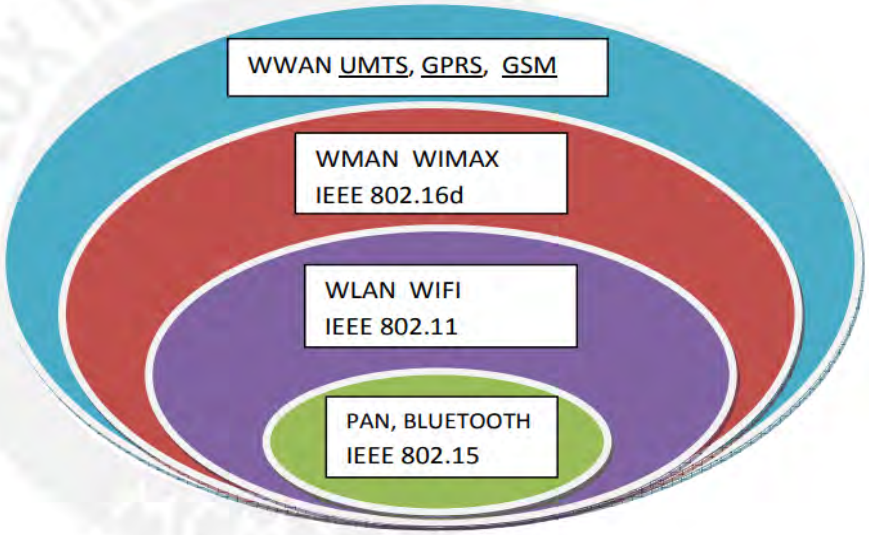

Figura 7: Clasificación de las tecnologías inalámbricas [16].

2.1.1.1 Red inalámbrica de áreas personales (WPAN)

Las redes personales se diseñan para las conexiones del dispositivo móvil sin emplear cables; como teléfonos móviles, cámaras de fotos o PDAs. Esta red, basada en los estándares IEEE 802.15, emplean principalmente tecnología Bluetooth [16]. Esta tecnología proporciona unas velocidades máximas del 1 Mbps con unos alcances máximos de aproximadamente 30 metros. Una de las ventajas destacadas de Bluetooth son sus bajos consumos de energías, esto lo hace especialmente adecuados para dispositivo periféricos de reducido tamaño [16].

2.1.1.2 Red inalámbrica de áreas locales (WLAN)

La WLAN están concebidas para ofrecer accesos inalámbricos en área con alcances típicos de 100 metros, siendo empleadas primordialmente con entornos como los hogares, las escuelas, sala del ordenador o ambiente de oficinas [15]. Emplea ondas electromagnéticas, como las de radio e infrarrojo, para establecer la conexión entre el dispositivo conectado con las redes a través del adaptador. Esto se realiza en vez de emplear el cable coaxial o de fibras ópticas, que son habituales en LAN convencional por cable, como Ethernet y Token Ring [16]. Esta red emplea el estándar IEEE 802.11 y se comercializan bajo la denominación Wi-Fi.

2.1.1.3 Red inalámbrica de áreas metropolitanas (WMAN)

Las redes metropolitanas están creadas para establecer conexiones de alta velocidad a larga distancia. Las WMAN se fundamentan en el estándar IEEE 802.16, comúnmente conocido como WiMAX (Worldwide Interoperability for Microwave Access). WiMAX son unas tecnologías de comunicación con arquitecturas puntos a multipuntos, diseñadas por ofrecer una transmisión del dato veloz mediante la red inalámbricas [15]. Se estima que esta red puede alcanzar distancias de hasta 50 km, Esto posibilita la interconexión de redes inalámbricas LAN más pequeñas a través de WiMAX, generando así una extensa red metropolitana inalámbrica (WMAN).

2.1.1.4 Red inalámbrica de áreas metropolitanas amplias (WWAM)

La red inalámbrica de áreas amplia abarca distancias superiores a 50km y generalmente operan con frecuencia con licencias. Estos tipos de red puede mantenerse con extensa área, como ciudad o naciones, mediante diversos sistemas satelitales o ubicación con antena gestionadas por unos proveedores del servicio del Internet [15]. Entre la tecnología esencial aplicada son GSM, GPRS o UMTS; estas son utilizadas ampliamente en las comunicaciones telefónicas móviles [16].

2.1.2. Topologías de redes inalámbricas

Las topologías de redes son ubicación lógica o física de componentes principales que la conforman. Entre las principales topologías de red se encuentran las topologías Punto-Punto y Punto-Multipunto [17].

2.1.2.1 Punto-Punto

Las redes Punto a Punto se caracterizan por arquitecturas en la que cada canal de datos se destina exclusivamente para las comunicaciones entre dos nodos, a diferencia de las redes multipunto, donde cada canal de datos puede emplearse para la comunicación de diversas maneras [18]. Las conexiones que vinculan el nodo de las redes puntos a puntos se puede categorizar como en 03 variedades según la dirección de la comunicación que transporta. Como primer lugar, tenemos el enlace Simplex, donde la transmisión ocurre en un solo sentido. A continuación, está el Half-dúplex, donde transmisión se realizan con ambos sentidos, pero de manera alternada, que pueden transmitir en momentos dados, sin la posibilidad de que ambos transmitan simultáneamente. Por último, el Full-dúplex permiten que las transmisiones se realicen en 02 dirección al mismo tiempo [18]. Los sistemas que utilizan topología punto a punto se construyen con conjuntos reducidos de elementos que conectan números limitados de unidades. Este tipo de redes puntos a puntos resulta notablemente sencilla tanto en términos de operación como de instalación [19].

La Figura 8 nos proporciona una representación más nítida de una conexión de tipo Punto a Punto.

Figura 8: Conexión Punto-Punto [19].

2.1.2.1 Punto-Multipunto

La red Punto-Multipunto es aquella en el cual los canales del dato tiene la capacidad de ser utilizado para la comunicación con varios nodos [18]. En las conexiones Puntos a Multipuntos, existe nodos centrales que se comunican con diversos puntos dispersos en las redes (Figura 9). Esta configuración, la comunicación fluye desde los nodos centrales hacia el punto remoto de las redes, así como desde este punto hacia los nodos centrales [19].

Figura 9: Conexión Punto-Multipunto [18].

2.2. Software Define Networking (SDN)

Como se mencionó anteriormente en el estado del arte del capítulo 1, Software Defined Networking (SDN) es un modelo en el cual la gestión centralizada de la red se lleva a cabo a través de un controlador central. Este enfoque posibilita la separación de la inteligencia de las redes convencionales del hardware, transfiriendo las capacidades de toma de decisiones al servidor. Esto implica que se puede lograr un control independiente de cada capa de la red sin depender de hardware y software costosos como routers y switches. En consecuencia, la red basada con SDN abstractizan el elemento y funciones con los controladores para ejecutar aplicación (como equilibradores de cargas), convirtiéndolas con redes programables según los requisitos de los usuarios [20].

2.2.1. Arquitectura

Centrándonos más detalladamente, las arquitecturas de la tecnología SDN conforme con RFC 7426 están conformados por 03 capas [20]. Primero, están las capas de infraestructuras o DAL (Device and Resource Abstraction Layer), compuesta con diversos dispositivos con las redes, son componentes básicos encargados de reenviar paquetes [20]. Operation plane, que se encuentra dentro de esta capa, contienen los estados de los dispositivos de redes y sus componentes [8].

En una segunda instancia, nos encontramos con las capas de controles, compuesta por CAL (Control Abstraction Layer) y la MAL (Management Abstraction Layer). Estas capas tienen la responsabilidad de acoger y reenviar el paquete proveniente de las capas de infraestructuras [20]. El Management Plane se encarga de supervisar, configurar y mantener el Plano de Operación, mientras que el Control Plane toma decisiones sobre cómo debe ser el reenvío de paquetes con los elementos de las redes e informa estas acciones a otros dispositivos para sus ejecuciones [8].

Finalmente, las capas de las aplicaciones, permitiendo los desarrollos de aplicación que facilitan las implementaciones del servicio de redes, como las seguridades, ingeniería de tráfico, enrutamiento, calidad de servicios, entre otros [20]. El Application Plane abarca la aplicación y servicio que determinan los comportamientos de las redes. La Figura 10 ilustra la arquitectura SDN tal como lo describe el RFC 7426. [20].

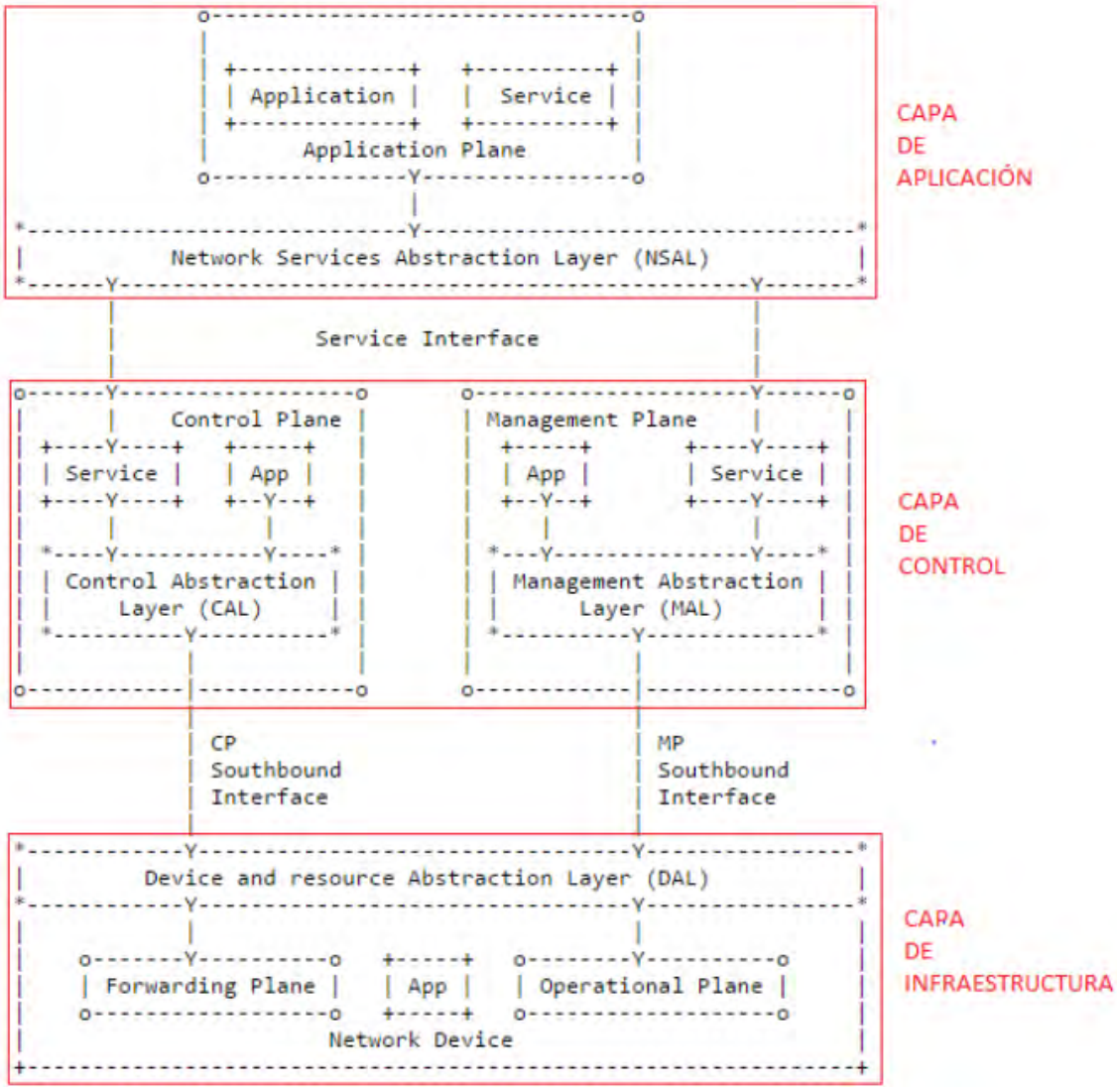

Figura 10: Arquitectura en capas de SDN [20].

2.2.2. Interfaces

Una interfaz es un punto donde se realiza la interacción de una entidad con otra. Los interfaces más importantes para la tecnología SDN son la SouthBound Interface y la NorthBound [20].

Figura 11 muestra un esquema de arquitectura donde muestra donde se utiliza cada una de las interfaces.

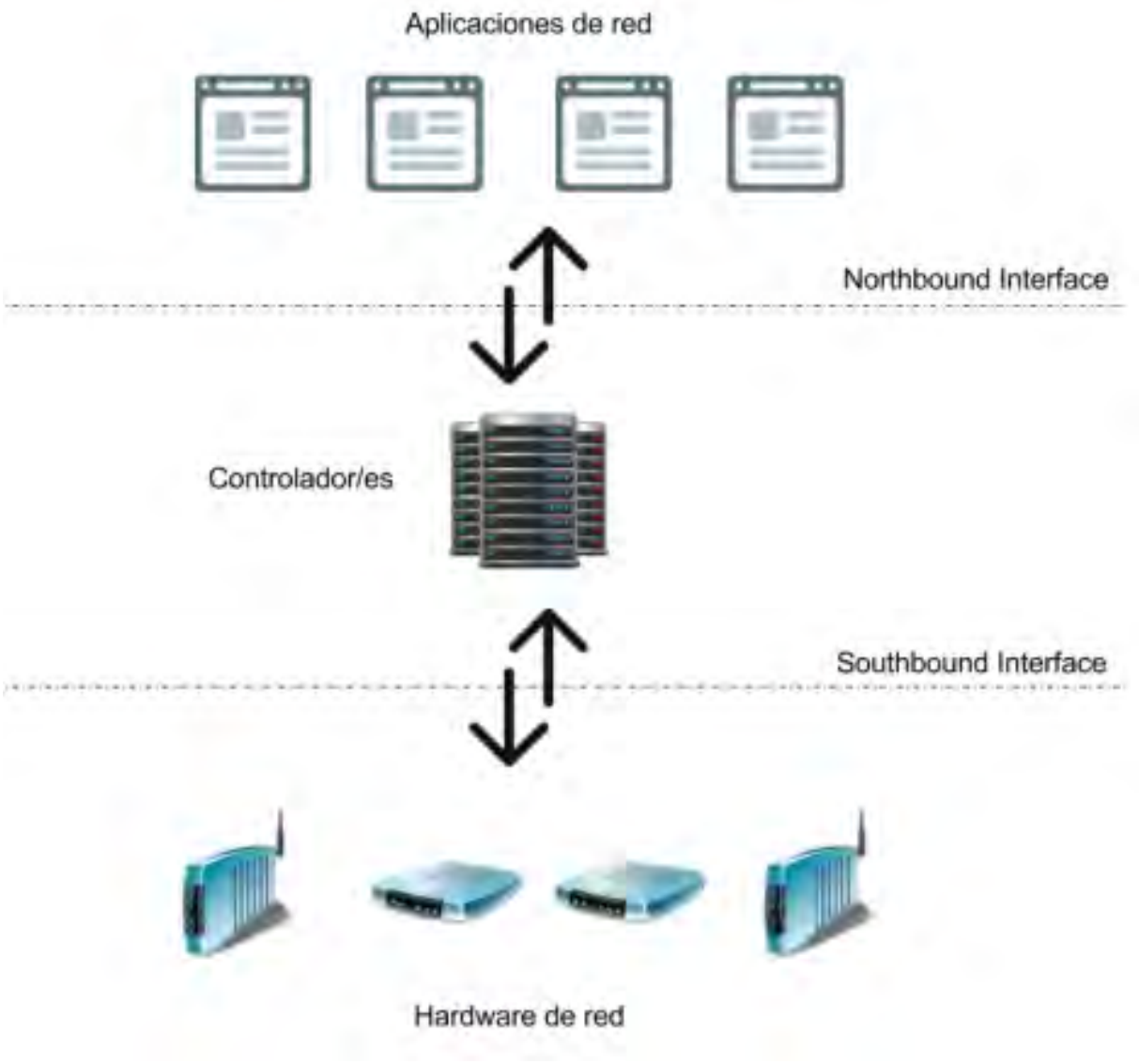

Figura 11: Interfaces NorthBound y SouthBound [7].

2.2.2.1 Interfaces NorthBound

Las interfaces NorthBound facilita las comunicaciónes con las capas de controles y capas de aplicaciones mediante la interfaz de programaciones API [20]. De igual manera, se utiliza para planificar los controles de las redes de la aplicación de redes que se ejecuta sobre los controladores. Esta aplicación, mediante API, presentan las capacidades de desarrollar flujo para dirigir paquete, distribuir los tráficos por diferentes rutas y responder ante cambios en la red como los fallos de enlace y, finalmente, redireccionar el tráfico para realizar distintas tareas de seguridad [7].

2.2.2.2 Interface SouthBound-OpenFlow

La SouthBound Interface se encarga de realizar la comunicación entre los elementos que se encuentran en la capa de infraestructura con el controlador, que está en la capa de control, con los protocolos OpenFlow [20]. OpenFlow representa el primer protocolo SDN que fue aceptado por las comunidades de vendedores e investigadores debido a su naturaleza de código abierto. Gracias a esta característica, el controlador tiene la capacidad de llevar a cabo funciones como manifestar las topologías de las redes, establecer flujo de redes y atender la solicitud de la aplicación de redes [7].

2.2.3. OpenFlow

OpenFlow, como se ha indicado previamente, son protocolos y estándares de códigos abiertos que permiten a controladores de las redes establecer la ruta del paquete mediante los switches de redes. Estos protocolos posibilita la implementación de SDN con switches, con propósito de ofrecer unas plataformas de red virtuales, abiertas y programables [7]. Adicionalmente, está asociado con las conexiones a distancia con los controladores SDN y switches, se presenta en la Figura 12. Estos dispositivos pueden ser Ethernet, router y punto de accesos inalámbricos (Access Point AP). Estos nos brindan accesos a la tabla de flujos y a la regla que indican la dirección de tráficos del paquete [5].

OpenFlow aprovecha la presencia de tablas de flujos en la mayoría de los switches, las cuales pueden ser consultadas para tomar acciones sin añadir demora al procesamiento. Esto se utiliza en las implementaciones del firewalls, NAT, QoS y la recopilación del estadística [7]. Por ello, permiten programarse la tabla de flujos de los dispositivos de redes para mejorar la eficiencia en el tráfico de datos.

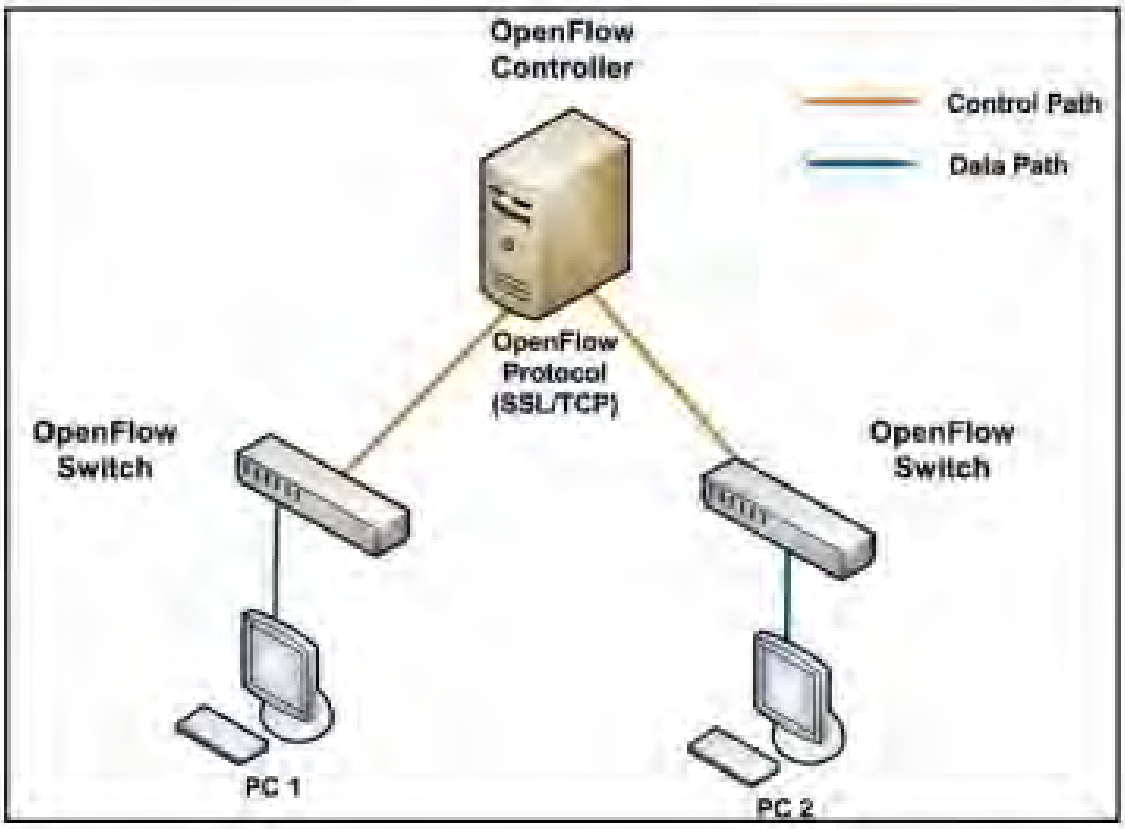

Figura 12: Arquitectura de red de OpenFlow [5].

Al realizar esta centralización del control, OpenFlow a través de su interfaz permite el manejo de dispositivo de las redes, permiten crear nuevas políticas para encontrar la mejor ruta, con anchos de bandas requeridos, para simplificar administración de las redes [6]. También permite la compatibilidad de diferentes dispositivos que se encuentren en la red sin importar la marca de fabricante que estos posean [6]. Por último, identifican y explotan los conjuntos de la función que corren con varios dispositivos como los Switches OpenFlow.

2.2.3.1 Switches OpenFlow

Los Switches OpenFlow deben de cumplir con unos requerimientos mínimos para que puedan tener un funcionamiento adecuado. Estos funcionamientos son:

- Tabla de flujos: Mediante este elemento se permite la acción de las entradas para determinar el proceso [21].
- SSL (Secure Sockets Layer): Corresponde a la capa de conexión segura, esta se utiliza entre el controlador y los dispositivos de conmutación [21].
- El protocolo OpenFlow, al proporcionar un medio de comunicación abierto y estandarizado, define una interfaz que permite la definición externa de cada entrada en la tabla [7].

2.2.4. Controladores SDN

Un controlador es el encargado de añadir y eliminar la acción para el flujo OpenFlow en las tablas del dispositivo. Proporciona unas interfaces de comunicaciones con el elemento de redes, como los switches OpenFlow, e interfaces para las interacciones con la aplicación del servicio de redes. Actualmente, existen varias implementaciones de controladores OpenFlow, por mencionar algunas:

- NOX: Fue el pionero entre los controladores OpenFlow, ofreciendo interfaces de programaciones de la aplicación (API) Northbound en C++ y Python, los cuales posteriormente quedaron desactualizada [7].

- POX: Similar a los controladores NOX, pero con compatibilidad para Python [7].

- Floodlight: Controladores basados con Java, desarrollados por comunidades de colaboradores de código abierto [7].

- Ryu: Controlador de código abierto basado en Python. Recibe respaldo de OpenStack, un software diseñado para virtualización y la creación de entornos en la nube [7].

- Trema: Es un marco de trabajo que permite a los programadores desarrollar un controlador SDN según sus preferencias, utilizando los lenguajes Ruby y C. Este entorno de programación produce un archivo de configuración que luego puede ser ejecutado por Trema para funcionar como un controlador SDN [22].

- Opendaylight (ODL): Consiste en la integración de diversos componentes de softwares, abarcan desde unos controladores completamente adaptable hasta la interfaz, protocolo del plug-ins y aplicación. Mediante las plataformas unificada, el usuario como el proveedor tienen la capacidad de innovar y colaborar para desarrollar y comercializar soluciones fundamentadas en NFV y SDN [22].

- Beacon: Controlador Java altamente adaptable. Se fundamenta en el marco de trabajo OSGI, que posibilita que las aplicaciones desarrolladas sobre él puedan ser iniciadas, detenidas, actualizadas e instaladas dinámicamente, sin perder la conectividad con los switches [7].

	Beacon	Floodlight	NOX	POX	Trema	Ryu	ODL
Soporte OpenFlow	OF v1.0	OF v1.0	OF v1.0	OF v1.0	OF v1.3	OF v1.0, v1.2, v1.3 y extensiones Nicira	OF v1.0
Virtualización	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Construcción de una herramienta virtual de simulación	Mininet y Open vSwitch	Mininet y Open vSwitch
Lenguaje de desarrollo	Java	Java	C++	Python	Rudy/C	Python	Java
Provee REST API	No	Si	No	No	Si (Básica)	Si (Básica)	Si
Interfaz Gráfica	Web	Web	Python+, QT4	Python+, QT4, Web	No	Web	Web
Soporte de plataformas	Linux, Mac OS, Windows y Android para moviles	Linux, Mac OS, Windows	Linux	Linux, Mac OS, Windows	Linux	Linux	Linux, Mac OS, Windows
Soporte de OpenStack	No	Si	No	No	Si	Si	Si
Multiprocesos	Si	Si	Si	No	Si	No	Si
Código Abierto	Si	Si	Si	Si	Si	Si	Si
Tiempo en el mercado	4 años	2 años	6 años	1 años	2 años	1 años	5 meses
Documentación	Buena	Buena	Media	Pobre	Media	Media	Media

Figura 13: Características de algunos controladores SDN [20].

2.3. Redes de acceso inalámbrica basadas en SDN

Las redes de acceso inalámbricas definidas por software buscan ofrecer un control centralizado y programático de la red, independientemente de los puntos de acceso inalámbricos (AP). Estos puntos de acceso implementan las instrucciones recibidas, como decisiones de política, y siguen siendo responsables de la transmisión y recepción del tráfico a través de los enlaces inalámbricos [23]. Por esta razón, y ya centrándonos en la parte de gestión mediante la tecnología SDN, la red inalámbrica presentará un cambio en su arquitectura muy similar a la arquitectura de la SDN y también contendrá un protocolo de gestión que se escogerá dependiendo del tipo de controlador que se utilizará en el diseños de las redes.

2.3.1. Arquitectura de las redes de acceso inalámbrica basada en SDN

En la Figura 14 se observa cómo sería una arquitectura de red de accesos basados con SDN, en donde se observa una similitud a la arquitectura de la tecnología SDN, ya que podemos observar los elementos que conforman la capa de infraestructura, los cuales serían los AP. También, se puede observar el controlador SDN centralizado. Por otro lado, por separado está la capa de aplicaciones, los cuales nos ayudaran a implementar los servicios que se desea para a la red. Para este tipo de arquitectura, sacado del libro [24], los AP solo retienen la función de reenvío de datos, por lo que el controlador no puede administrarlos. Por ende, es necesario establecer una vía de comunicación separada entre el controlador y cada AP; estos elementos suelen ser los switches OpenFlow como se pueden observar en Fig 6.

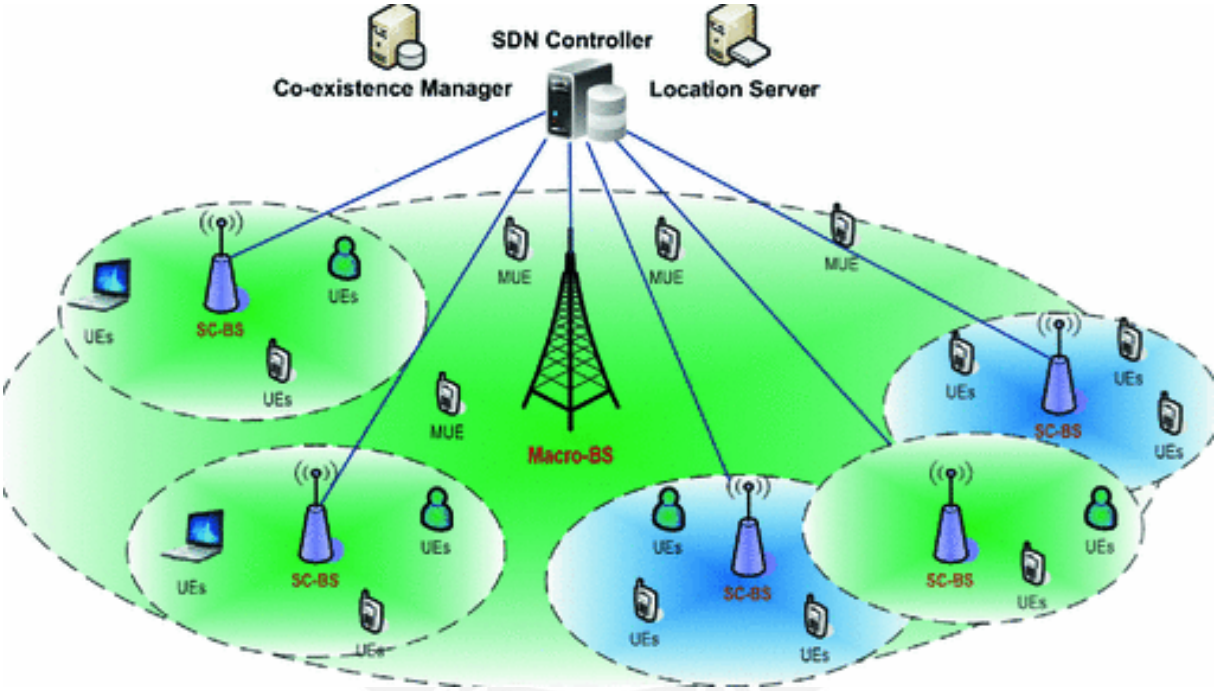

Figura 14: Arquitectura de una red de acceso basada en SDN [25].

2.3.2. Protocolos de gestión para SDN

En la capa de control de la arquitectura SDN se encuentra el plano de gestión, el cual es el MAL (Management Abstraction Layer), como se observa en la Figura 8. Este plano de gestión tiene la responsabilidad de realizar funciones de supervisión, configuración y mantenimiento del dispositivo de red. En SDN, la administración de las redes se ejecuta mediante una estructura jerárquica que incluye capas de protocolos, modelos y datos. Los protocolos de gestión emplean un lenguaje de modelo de datos como YANG (Yet Another Next Generation) para la configuración y obtención de datos del estado de los dispositivos. Algunos de los protocolos de gestión de red SDN incluyen NETCONF, RESTCONF, gRPC y OF-CONFIG (OpenFlow Management and Configuration Protocol) [26].

- NETCONF: Son estándar de configuraciones de redes establecido por el IETF en el RFC 6241, el cual ofrece métodos sencillos para establecer, modificar y excluir las configuraciones del dispositivo de redes. Adopta un modelo de comunicación de solicitud y respuesta mediante RPC (llamada a procedimiento remoto), utilizando codificación XML y un transporte seguro a través de SSH (Shell Seguro). NETCONF define repositorios de datos que almacenan la configuración de los dispositivos de red, junto con operaciones CRUD (Crear, Leer, Actualizar, Eliminar) que posibilitan la recuperación, configuración, duplicación y eliminación de estos repositorios de datos [26].

- RESTCONF: Es un estándar de configuración de red que utiliza HTTP, establecido por el IETF en el RFC 8040. A través de este protocolo, las aplicaciones web tienen la capacidad de acceder y ajustar la configuración de un dispositivo de red mediante una arquitectura cliente/servidor basada en RPC, donde los clientes y servidores operan según el modelo RESTCONF [26].

- gRPC: Es una plataforma moderna de llamadas a procedimientos remotos (RPC) de códigos abiertos y alto rendimiento que es compatible con diversos entornos. Ofrece una eficiente conexión entre servicios, tanto dentro como entre centros de datos, con capacidad plug-and-play para equilibrio de carga, seguimiento, verificación de estados y autenticaciones. Además, puede aplicarse en fases finales de las informáticas distribuidas para establecer conexiones entre dispositivo, aplicación móvil y navegador con servicio de backend [27].

- OF-CONFIG: Se trata de un protocolo de gestión creado por la ONF (Open Networking Foundation) con el propósito de manejar switches OpenFlow, ya sean físicos o virtuales. Su gestión se lleva a cabo mediante NETCONF, utiliza XML para la codificación de datos y se apoya en SSH como método de transporte [26].

Figura 15 muestra un cuadro con características de los protocolos de gestión para SDN.

	NETCONF	RESTCONF	gRPC
Standard	IETF	IETF	-
Resources	Paths	URLs	protobuf
Data Models	YANG Core Models	-	-
Data Modeling Language	YANG	Undefined	C++, Java, Python, Go, Ruby, ...
Management Operations	NETCONF	HTTP operations	Custom
Encoding	XML	XML, JSON, ...	Binary
Transport Stack	SSH TCP	SSL HTTP TCP	-

Figura 15: Características de los protocolos de gestión [28].

2.3.3. REST APIs

Parte importante de una red SDN son las REST API(Representational State Transfer Application Programming Interfaces), las cuales utilizan el modelo cliente/servidor y se comunican mediante el protocolo HTTP [20]. Estas se usan para lograr el intercambio de información entre el controlador y las aplicaciones. Además, por medio del protocolo HTTP, pueden acceder a la web y obtener información de los servidores web. También, son capaces de editar, publicar o eliminar información a través de ciertos comandos, los cuales son los siguientes:

- POST: Comando usado para la creación o la publicación de datos en la aplicación.
- GET: Comando usado para obtener datos de la aplicación.
- PUT: Comando utilizado para configurar parámetros dentro de una aplicación.
- DELETE: Comando utilizado para borrar datos de una aplicación.

Por otro lado, cabe resaltar que los formatos de datos para su interacción son del tipo XML, JSON o YAML, los cuales se pueden manejar mediante lenguajes de programación orientados a objetos como Python y Java. Entre uno de los formatos que brinda más ventajas al momento del intercambio de información es el JSON, el cual es un formato de texto más ligero y es mucho más entendible que el XML y YAML. Entre las ventajas que proporciona JSON es que permite a las máquinas tener una mayor facilidad en el análisis y posee un alto rendimiento. Por esta razón se hace una mayor preferencia para utilizar este formato de texto.

2.4. Monitoreo de Redes

Los monitoreos de redes se refieren a la utilización de un sistema que realiza una vigilancia continua sobre una red de computadoras con el objetivo de detectar componentes que puedan presentar fallos o funcionar lentamente. Posteriormente, informa al administrador de red a través de correos electrónicos, mensajes de paginación o diversas alarmas. Estos sistemas de monitoreo de red están diseñados para identificar problemas derivados de la sobrecarga, fallas en los servidores y posibles inconvenientes en la infraestructura de red, entre otros dispositivos. Entre principales sistemas de monitorización se tienen los siguientes:

- Cacti
- Nagios
- PRTG Network Monitor

2.4.1. Cacti

Cacti son utilidad de códigos abiertos que posibilita recopilación de dato a lo largo del tiempo procedente de diversos tipos de dispositivos, con el propósito de generar gráficos que representen su rendimiento a lo largo de un período determinado [29]. Esta herramienta ofrece a usuarios de todo el mundo un sólido y ampliable marco para la gestión de fallos y supervisión operativa. Además, incorpora unos marcos de recopilaciones del dato completamente distribuidos y resistente a fallos, así como función avanzada de automatizaciones basada con plantilla para dispositivo, gráfico y diversos métodos de adquisiciones del dato [30]. La adquisición de datos suelen ser los parámetros que posee una red cualquiera, como la supervisión de los servidores o el tráfico de las interfaces de la red. Entre uno de los parámetros más resaltantes que puede monitorear cacti es las estadísticas de las interfaces, las cuales nos detallan el tráfico enviado y recibido a cada puerto en específico. Esto nos ayuda en la recopilación de la información sobre las cantidades de tráficos que se envían y reciben sobre las interfaces del dispositivo.

2.4.2. Nagios

Nagios es una herramienta de supervisión de redes de código abierto ampliamente utilizado que observa los equipos y servicios designados, emitiendo alertas cuando su rendimiento no cumple con los criterios establecidos [31]. Sus características clave incluyen la supervisión del servicio de redes (SMTP, POP3, HTTP, SNMP, entre otros), los monitores del recurso del hardware del sistema (carga de procesadores, usos del disco, memorias, estados del puerto, etc.), independencias del sistema operativo, la capacidad de supervisión remotas con el túnel cifrados SSL o SSH, y opción de planear plugins personalizados para nuevo sistemas [31]. También posee la capacidad de monitorizar otros parámetros de la red como monitoreo de conectividad y de traceroute, los cuales se tomarán mayor énfasis para el desarrollo de esta tesis.

- Monitoreo de conectividad: Es un proceso que se utilizan para medirse cuantitativa o cualitativamente los rendimientos de una infraestructura de una red.
- Monitoreo de Traceroute: Asigna y agrega paquetes a medida que viajan desde nuestros nodos de supervisión a sus hosts y muestra la pérdida de paquetes y la latencia.

2.4.3. PRTG Network Monitor

PRTG Network Monitor es una herramienta esencial de monitorización de redes que desempeña un papel importante en garantizar el rendimiento del sistema informático y prevenir posibles fallas en las redes. Su utilidad radica en la optimización de la red al proporcionar informaciones detalladas sobre la utilizacion de las bandas anchas y diverso recurso de las redes [32]. Entre los principales recursos o parámetros que puede supervisar se tienen los siguientes:

- Monitorización del ancho de banda de la red: Se utiliza para evaluar el ancho de banda efectivo disponible en un sistema local. Este proceso proporciona información en tiempo real, incluyendo velocidades de carga y descarga, y desempeña un papel preventivo en la detección de posibles congestiones en la red.
- Monitorización de Rango de puertos: Monitorea un servicio de red al conectarse a varios puertos TCP / IP. Intenta conectarse a los números de puerto TCP / IP especificados de un dispositivo en sucesión y espera a que el dispositivo acepte cada solicitud. También, supervisa una serie de puertos e indica si están abiertos o cerrados.
- Tráfico en la interfaz de red: Se refiere a las informaciones que se transmiten mediante unas redes en un instante específico.
- Monitoreo de cantidad de hosts: Nos permite obtener las cantidades de máquinas host conectada a las redes.

2.5. Entornos de simulación para controladores SDN

Dado que SDN posibilita la creación de redes más autónomas mediante la gestión inteligente de cada nodo a través de un software de control, ha experimentado un continuo desarrollo por parte de destacados desarrolladores como CISCO. Esto ha llevado al surgimiento de entornos de emulación para SDN, los cuales se presentan como soluciones para nuevos desarrolladores e investigadores que buscan implementar laboratorios centrados en estas redes sin incurrir en grandes costos. En la actualidad, diversas herramientas están disponibles para emular redes SDN, cada una con sus propias características y enfoques, entre estas herramientas encontramos Mininet y Virtual Network Research Testbed (VNRT).

2.5.1. Mininet

Mininet es un simulador de redes diseñado para sistemas Linux, que ofrece la capacidad de generar entornos de redes virtuales mediante la creación de componentes como switches, routers, hosts, entre otros; y son gestionados por un controlador que se fundamenta en la tecnología de Red Definida por Software (SDN) [33]. Este simulador presenta diversas ventajas, como su rapidez en la simulación, la posibilidad de configurar redes personalizadas para la implementación de programas prácticos como Wireshark adaptado para Linux, la programación de la transmisión de paquetes, entre otras funcionalidades. Además, es compatible con una amplia gama de dispositivos, como ordenadores y servidores, proporcionando una alta capacidad de difusión que facilita el intercambio y la replicación sencilla de resultados. Además, permite la emulación mediante scripts en Python [33].

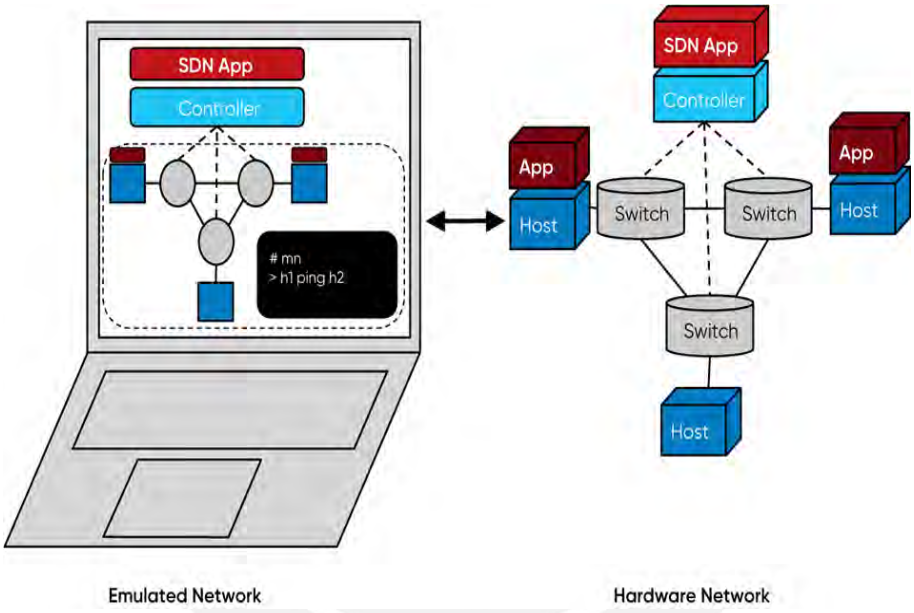

Figura 16: Herramienta de emulación Mininet [34].

2.5.1.1 Miniedit

Se trata de una herramienta que amplía las funcionalidades de Mininet, posibilitando la creación fácil de redes a través de una interfaz gráfica en el terminal. Esta interfaz simplifica la creación de redes al manejar la programación en un plano oculto al usuario. Sin embargo, es importante señalar que esta plataforma tiene algunas limitaciones en comparación con todas las capacidades que ofrece Mininet por sí mismo. Figura 17 muestra los entornos de simulación Miniedit

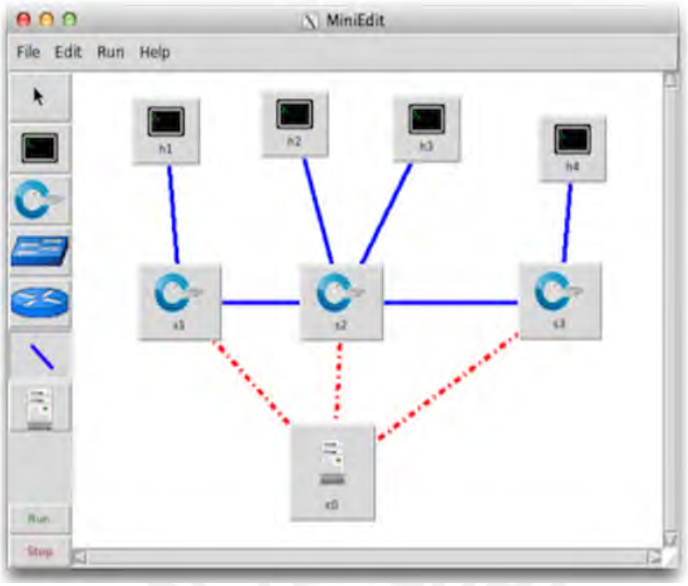

Figura 17: Entorno de trabajo Miniedit [35].

En el lateral de la Figura 17, podemos observar las herramientas de trabajo, concretamente son las siguientes:

 FLECHA DE SELECCIÓN: Se utiliza para desplazar los nodos dentro del área de trabajo.

 HOST: Facilita la inclusión de elementos como hosts o terminales en el lienzo; cuando está activada, posibilita la adición de varios elementos que luego pueden configurarse a través de las propiedades disponibles en el menú desplegado mediante el botón derecho.

 CONMUTADORES:Genera un Open vSwitch con capacidad para llevar a cabo una gestión estándar y activar de manera programable funciones de reenvío o transmisión de información, siendo configurables de manera similar al elemento mencionado anteriormente.

 CONMUTADORES TRADICIONALES: Establece un switch Ethernet con una configuración predefinida que opera de manera autónoma, sin depender del controlador. Este dispositivo no admite configuraciones y comienza con el protocolo Spanning Tree desactivado y se recomienda evitar conexiones en bucle.

 ROUTERES TRADICIONALES: Genera routers básicos que operan de manera independiente al controlador. En esencia, se trata de un host con la capacidad de IP Forwarding habilitada. No es posible configurar este nodo a través de MiniEdit.

 ENLACES: Establece la conexion entre los elementos de las redes.

 CONTROLADORES: Facilita la implementación de varios controladores de red; de manera predeterminada, se genera un controlador de OpenFlow.

EJECUTAR / PARAR: A través de este componente, se ejerce control sobre la simulación de la red creada. Durante la ejecución de una simulación, al hacer clic derecho sobre un elemento, se despliegan funciones operativas exclusivas del entorno de emulación, tales como abrir un terminal, revisar su configuración o ajustar el estado de un enlace (activándolo o desactivándolo).

2.5.2. Virtual Network Research Testbed (VNRT)

Es un entorno de pruebas de alta fidelidad y alta capacidad que sirve para probar el comportamiento de nuevas tecnologías. Este emulador provee una interfaz web amigable para el usuario, proporcionando las herramientas necesarias para crear, editar, exportar, importar y eliminar los escenarios de prueba de forma rápida [36]. VNRT se calibra de acuerdo a los recursos disponibles que el hardware posea, el cual puede estar conformado por uno o varios servidores. Esto implica que si se requiere simular escenarios más complejos, solo se deberá agregar más recursos (hardware) y volver a calibrar el emulador, lo cual convierte al VNRT en una solución escalable [36]. Figura 18 muestran las interfaces gráficas de VNRT.

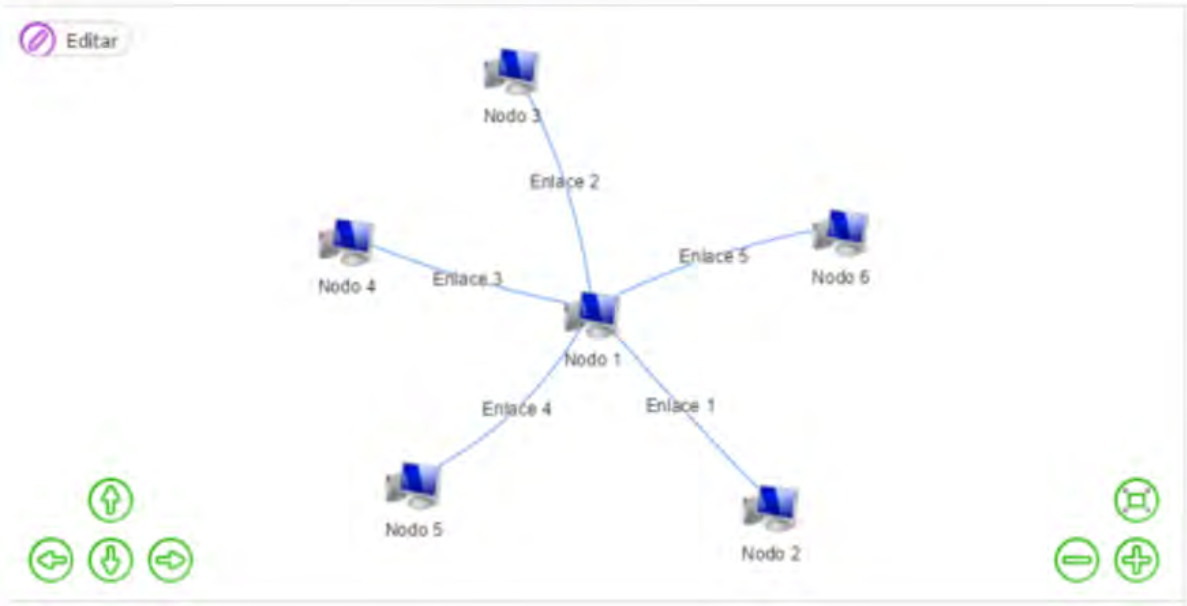

Figura 18: Interfaz gráfica VNRT [36].

3. Diseño del sistema

El capítulo, mencionará los diseños de la topología de un modelo solución, los elementos de software empleados para el diseño y la simulación de una red SDN.

3.1. Consideraciones de diseño para la topología SDN

El diseño de la topología SDN, que considera funcionalidades de gestión, se centra en la obtención de los parámetros fundamentales para un fácil y eficiente monitoreo, como también una mejor administración de la red. Entre los parámetros más importantes a considerar son el ancho de banda de la red, el rango de puertos, el tráfico en la interfaz de la red, estadísticas de interfaces, etc. Para ello se desea que el controlador SDN realice la recopilación de cada uno de estos parámetros, por lo cual, como mejor opción para utilizar, se optó por el controlador Floodlight. Esta decisión se debe a que el controlador Floodlight cuenta con una buena documentación, provee REST API, su lenguaje de programación es el lenguaje JAVA y su curva de aprendizaje es moderada.

Para simular e implementar una red definida por software o SDN se deben considerar elementos en sus tres capas:

- Capa de Infraestructura: Se debe considerar switches que soporten el protocolo Open-Flow. Un switch para la zona en donde se encontrará una máquina virtual (VM), otro para el otro extremo en donde también se encontrará una VM y un switch adicional, que gestione el flujo de datos entre los switches anteriores.
- Capa de Control: Se debe considerar un controlador SDN Floodlight que permita la gestión centralizada del flujo de datos entre los switches SDN.
- Capa de Aplicación: Se debe considerar un monitoreo de los parámetros de la red y de acuerdo a este análisis implementar las políticas basadas en REST API dentro del controlador.

Además de lo antes mencionado, debe considerarse una conexión de la red hacia Internet para poder monitorear el tráfico de datos que proviene de la red externa. Finalmente, según los requerimientos antes mencionados, en la Figura 19, se muestra la topología diseñada.

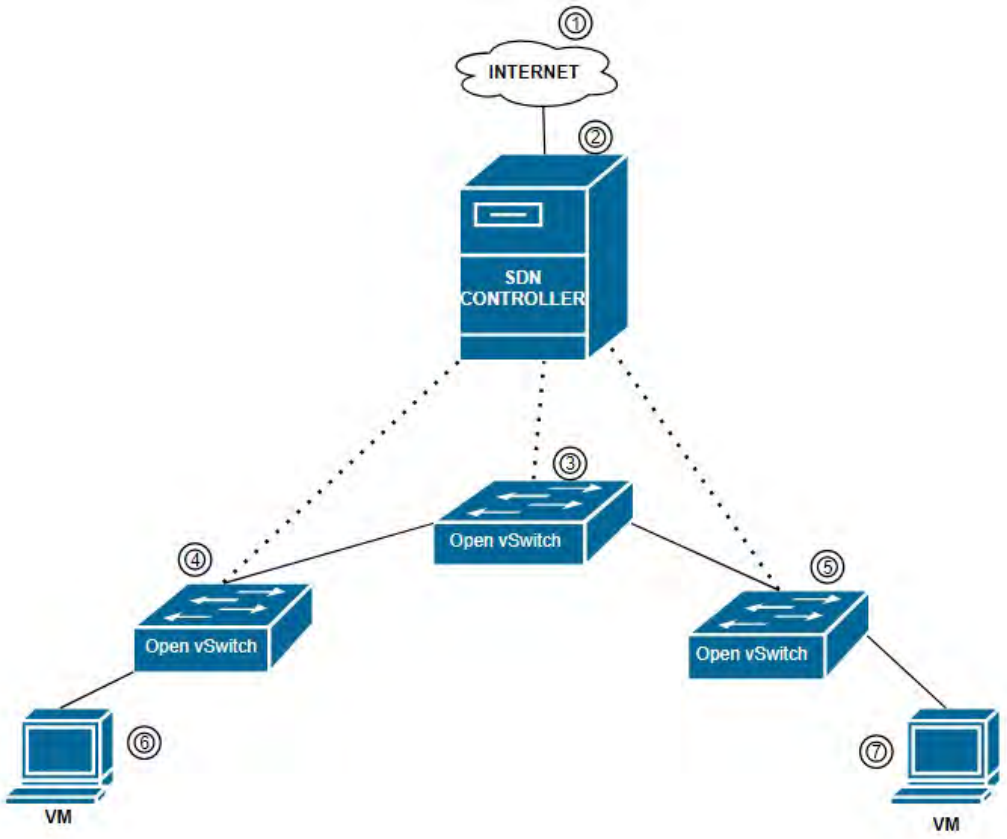

Figura 19: Topología de la red SDN [Elaboración propia].

3.1.1. Elementos de la Topología SDN

- 1) Internet: Permitirá el tráfico de la red externa hacia una red interna.
- 2) Controlador SDN: Controlador que gestionará el tráfico de forma centralizada en toda la red.
- 3) Open vSwitch central: Switch SDN que permitirá la gestión entre el elemento 4 y el elemento 5
- 4) Open vSwitch izquierdo: Switch SDN que permitirá la gestión de los clientes de la zona rural de la izquierda
- 5) Open vSwitch derecho: Switch SDN que permitirá la gestión de los clientes de la zona rural de la derecha.

3.2. Consideraciones de diseño para las conexiones de Simulación

El diseño de las conexiones para la simulación, se basa en el principio de la virtualización de redes. La simulación de la topología SDN para esta tesis se debe tener en cuenta las siguientes herramientas:

- Emulador de Red: Se debe considerar un emulador de red, donde se incluirá la topología. Mininet es una herramienta versátil que soporta el protocolo OpenFlow y permite la escalabilidad de una red simulada a una red real.
- Controlador SDN: Se debe considerar el controlador Floodlight, por las características mencionadas anteriormente
- Máquina Anfitrión: Se debe considerar una máquina anfitriona donde se instalará los dos elementos antes mencionados para la simulación.

  Adicionalmente, se considerará lo siguiente
- Instalación en una primera máquina virtual de Linux, donde se simula la topología sin el controlador, ya que este será conectado de forma remota.
- Instalación en una segunda máquina virtual de Linux, que se conectará hacia la topología de forma externa o remota. Además, solo el controlador debe tener conexión a Internet.
- Máquina Anfitrión: Debe soportar la instalación de las dos máquinas virtuales anteriormente mencionadas. Así como, una conexión a Internet y un navegador, en el cual se verá reflejado el resultado de la simulación total.

De acuerdo a lo antes mencionado, Fig 20, se visualiza el diseño de las conexiones para la simulación.

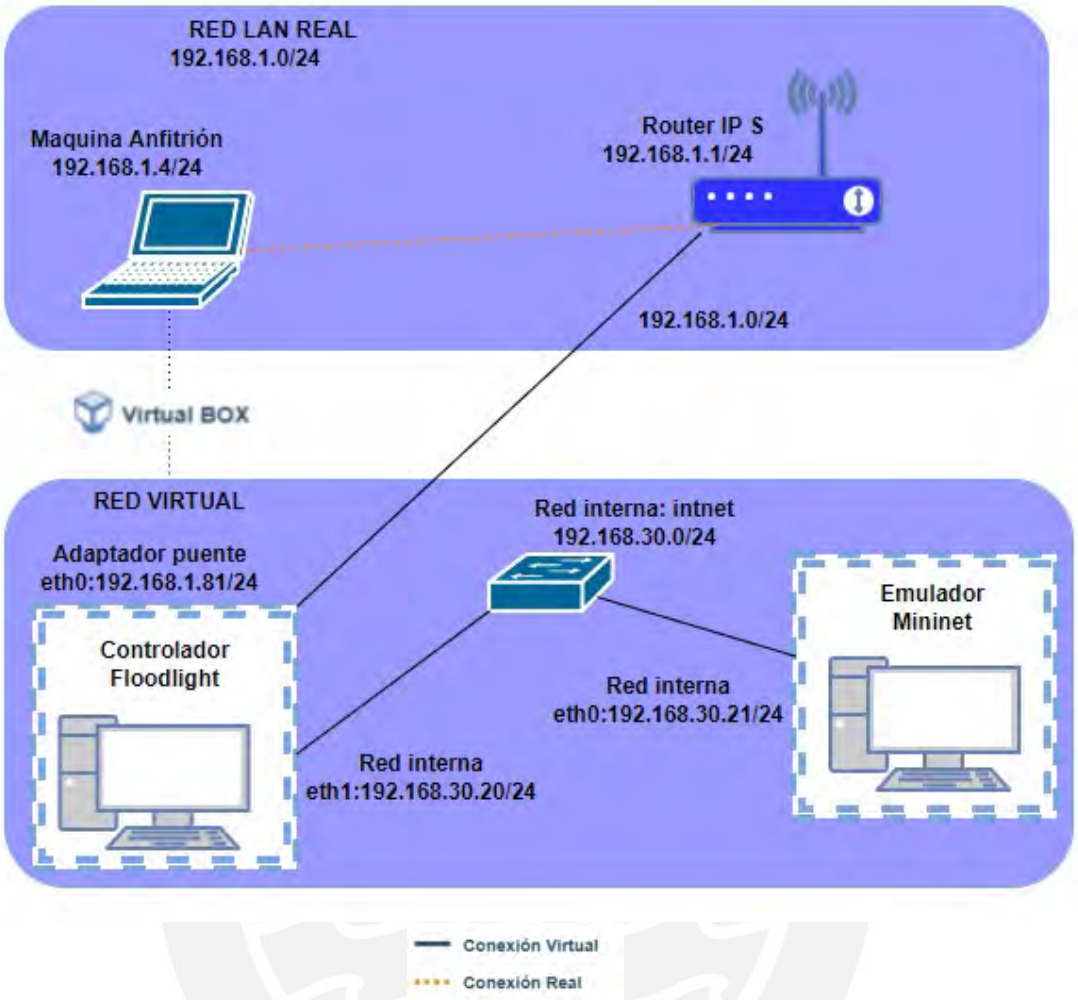

Figura 20: Diseño de Conexiones para la Simulación [Elaboración propia].

3.2.1. Elementos del Diagrama de Conexiones para la Simulación

- RED LAN REAL: En esta red se encuentra la máquina anfitriona que se utilizara para la instalación de los elementos previamente mencionados, más el router ISP o proveedor de servicios, que brinda el servicio de Internet. Esta red es una red interna con dirección 192.168.1.0/24.
- RED VIRTUAL: Se utiliza el programa Virtual Box, donde se despliegan las siguientes máquinas virtuales:
  - Controlador Floodlight: Accede a internet a través de una conexión Adaptador Puente, que simula una conexión cableada hacia el Router ISP de la red real. Es por ello que, la dirección para esta conexión que se encuentra en red con la RED LAN REAL (Dirección IPv4 -192.168.1.0/24). Específicamente esta máquina virtual toma la dirección IPv4 192.168.1.81/24. Por otro lado, se debe conectar hacia una red interna simulada con el emulador de red. Para esta segunda conexión se elige una dirección IPv4 192.168.30.0/24, y la máquina del controlador, toma la dirección: 192.168.30.20/24.

- Emulador Mininet: tendrá conexión al controlador, mediante una red interna, es por ello que se debe simular una conexión virtual interna. Se elige para la red interna una dirección 192.168.30.0/24 y la máquina del emulador, toma la dirección: 192.168.30.21/24.

3.3. Configuración del escenario de pruebas

3.3.1. Diagrama de bloques del Controlador

En la Figura 21 se visualiza los diagramas de modulo de nuestros controladores y la interacción con las aplicaciones, utilitarios y servicios que se pueden manejar con este. Para la simulación de la topología planteada se hace enfoque al módulo static flow entry pusher para obtener los parámetros principales de la red previamente mencionado.

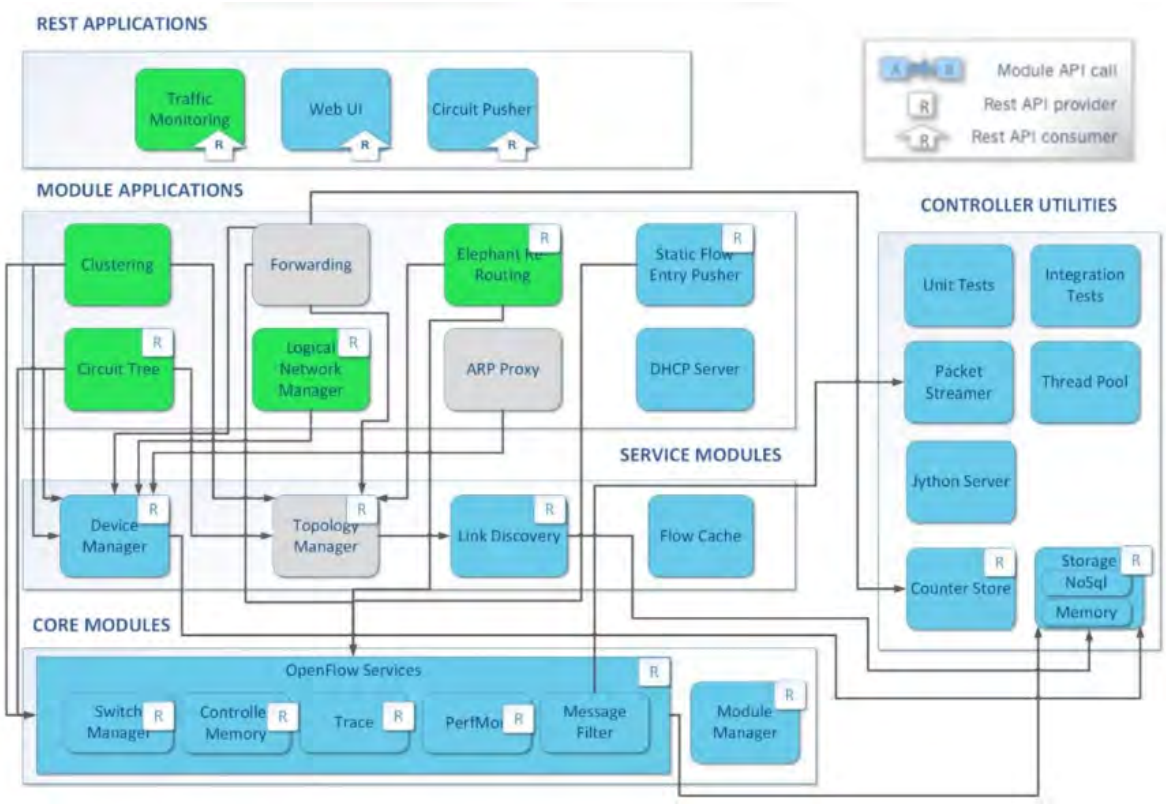

Figura 21: Diseño de Conexiones para la Simulación [Elaboración propia].

3.3.2. Instalación del Controlador dentro de una Máquina Virtual

3.3.2.1 Consideración para la Instalación

La instalación de la máquina virtual del controlador, se instalará a partir de la documentación encontrada de este proyecto. Para ello se requiere las siguientes especificaciones de hardware en una máquina anfitrión:

- Memoria RAM: 4GB (mínimo)
- Disco Duro: 8GB (mínimo)

3.3.2.2 Configuración de Adaptadores de Red de la VM

Se crea una nueva máquina virtual en Virtual Box, donde se copiará la imagen del controlador descargado. Dentro de esta máquina virtual se configuran los adaptadores de red:

- Adaptador 1: Adaptador Puente
- Adaptador 2: Red Interna. Para la red interna se requiere un nombre y se elige ïntnet". En la Figura 22 se puede observar la configuración de la máquina virtual.

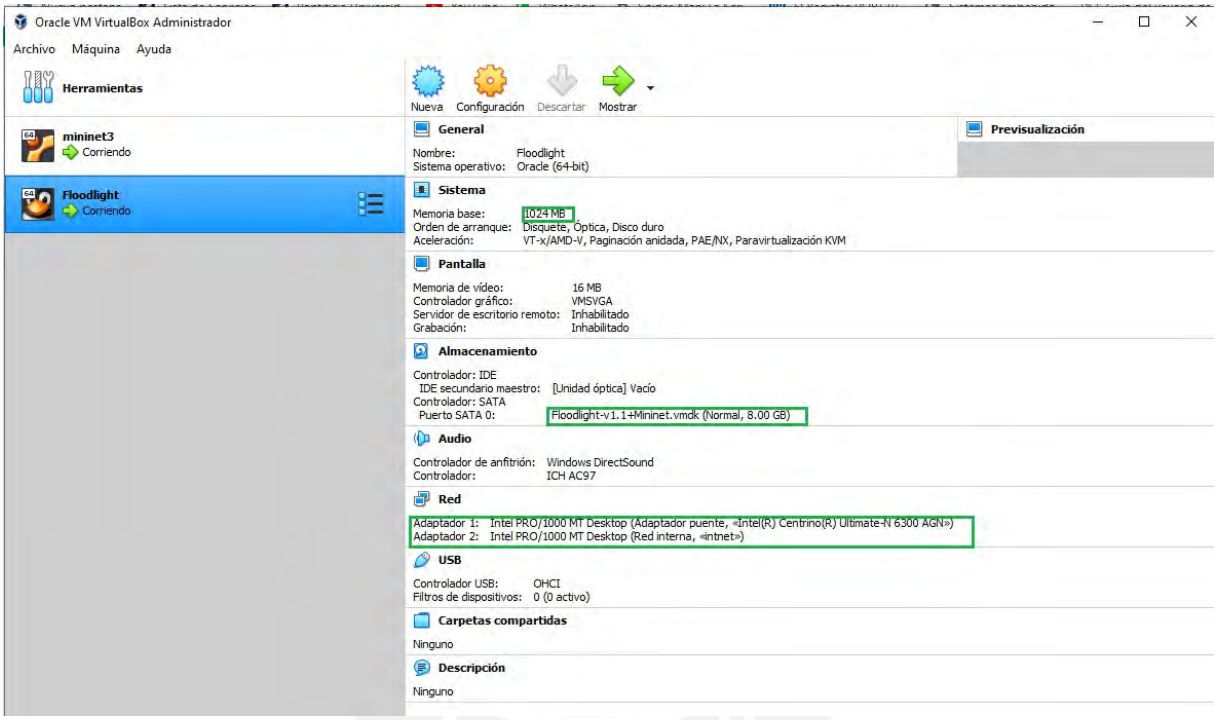

Figura 22: Configuración de la máquina virtual del controlador Floodlight [Elaboración propia].

3.3.2.3 Inicialización del Controlador

Se inicializa la máquina virtual y se debe configurar las direcciones IP de sus adaptadores de red. En la Figura 23, se observa, que esta máquina virtual tiene dos adaptadores de red activos, ambos se configuran según la conexión requerida.

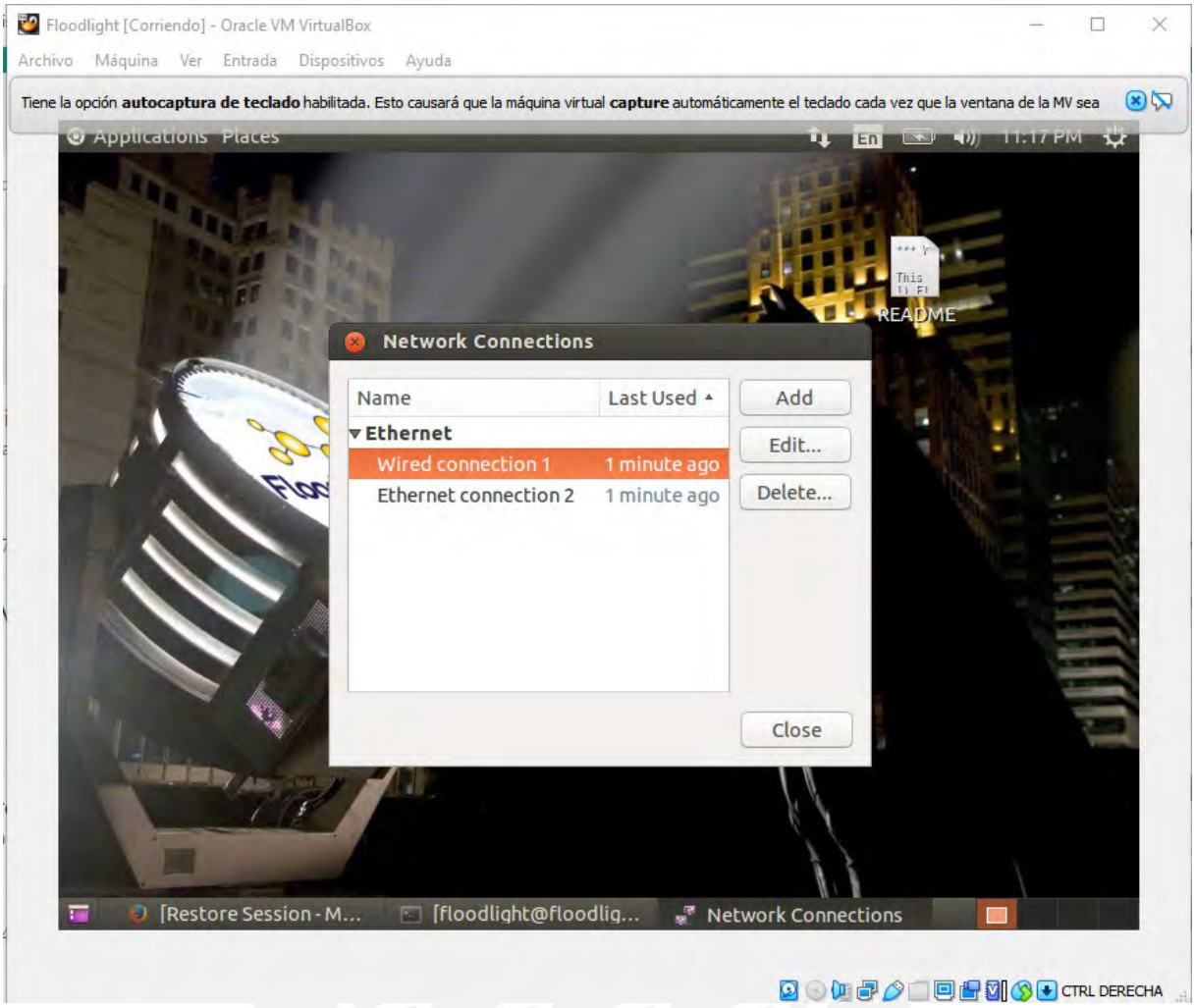

Figura 23: Adaptadores de red de la máquina virtual del controlador [Elaboración propia].

La configuración de la dirección IPv4 de la conexión: "Wired Connection 1", se configura con el método DHCP automático, esto es porque, se desea conseguir que mediante una conexión virtual puente hacia la máquina anfitrión el protocolo DHCP del router ISP brinde una dirección automática a esta máquina virtual, con la finalidad de acceder a Internet. La configuración de la dirección IPv4 de la conexión:"Wired Connection 2", se configura con el método manual, esto es porque se desea conseguir añadir una IP de forma manual que configure a la máquina virtual y le permita conectarse en una red interna de dirección: 192.168.30.20/24. Cuando la configuración, se haya realizado, se procede a verificar las conexiones de la máquina virtual donde se encuentra el controlador, esta verificación se grafica en la Figura 24.

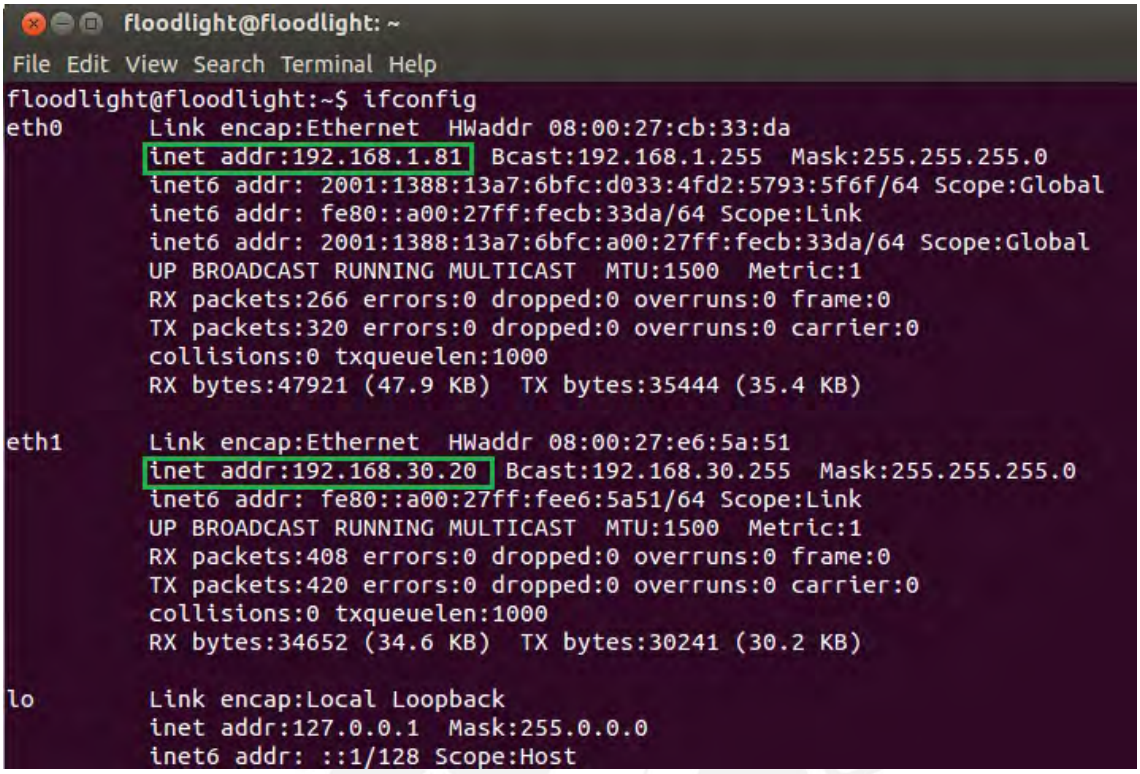

Figura 24: Configuración de IP de la máquina virtual [Elaboración propia].

El controlador, aún no ha sido inicializado, para ello se deben usar los siguientes comandos de Linux:

- cd floodlight
- ant
- java -jar target/floodlight.jar

Después de ejecutar dichos comandos se puede observar que el controlador se encuentra esperando, de forma remota, la topología por uno de los adaptadores de red de la máquina virtual previamente configurados. En la Figura 25, se observa que mediante el protocolo Link Layer Discovery Protocol (LLDP) los paquetes del controlador están siendo enviados para la espera de una información remota. El protocolo LLDP es un protocolo de detección de pares o vecinos de capa 2, permite que los dispositivos, en este caso el controlador, anuncie su información o la espera de información para ser ejecutada.

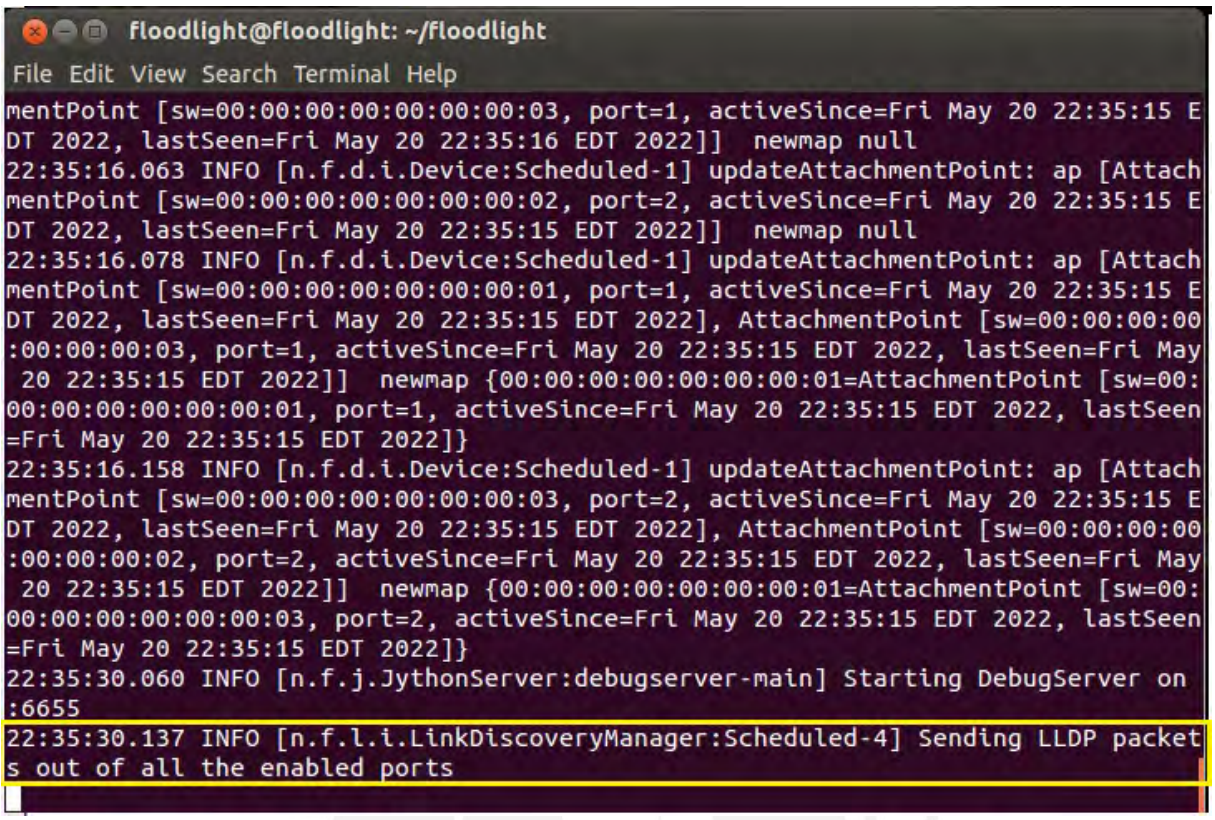

Figura 25: Controlador Floodlight Activado [Elaboración Propia].

El controlador floodlight se encuentra esperando que se ejecute la topología en mininet a través de la red interna para comenzar a gestionarla y mostrarla en la interfaz WEB.

3.3.3. Simulación de la Topología en Mininet

3.3.3.1 Configuración de Adaptadores de Red de la máquina virtual

Se crea una nueva Máquina Virtual (VM) en Virtual Box y se configura solo un adaptador de red como red interna con el mismo nombre de la red interna configurada en la Máquina Virtual del controlador floodlight. En la Figura 26, se observa, que el nombre es el mismo intnet.

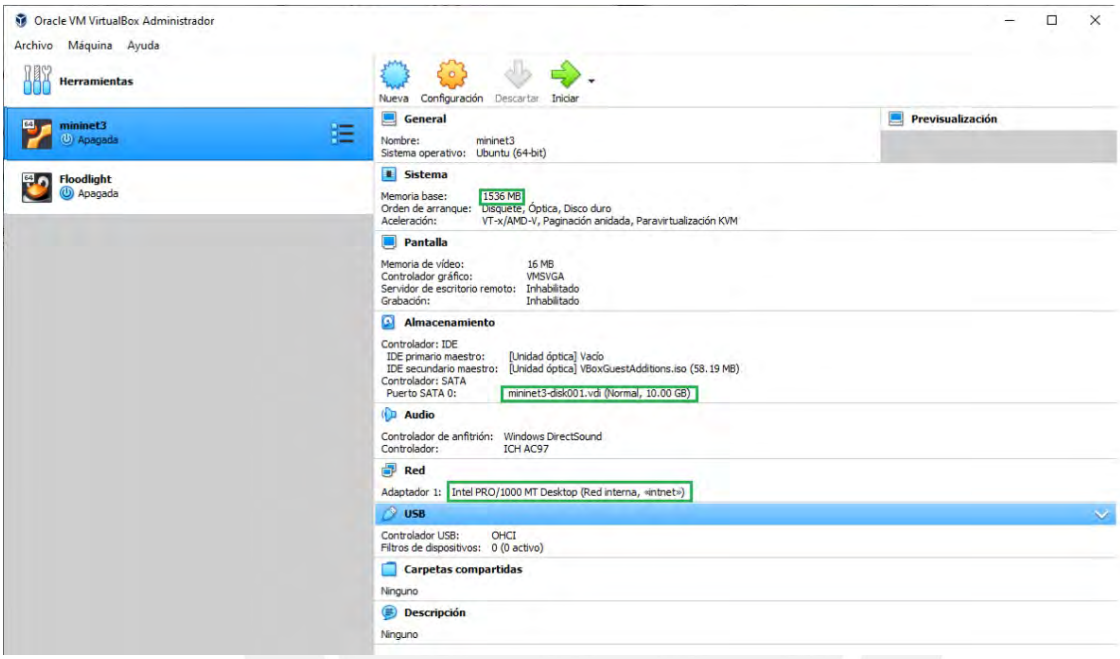

Figura 26: Configuración de la máquina virtual del emulador Mininet [Elaboración Propia].

3.3.3.2 Inicialización y Simulación en Mininet

Se inicializa la máquina virtual y se configura la dirección IPv4 con una dirección IP correspondiente a la red 192.168.30.0/24. En este caso se configura con la IP 192.168.30.21. En la Figura 27 se puede verificar esta configuración de dirección IP.

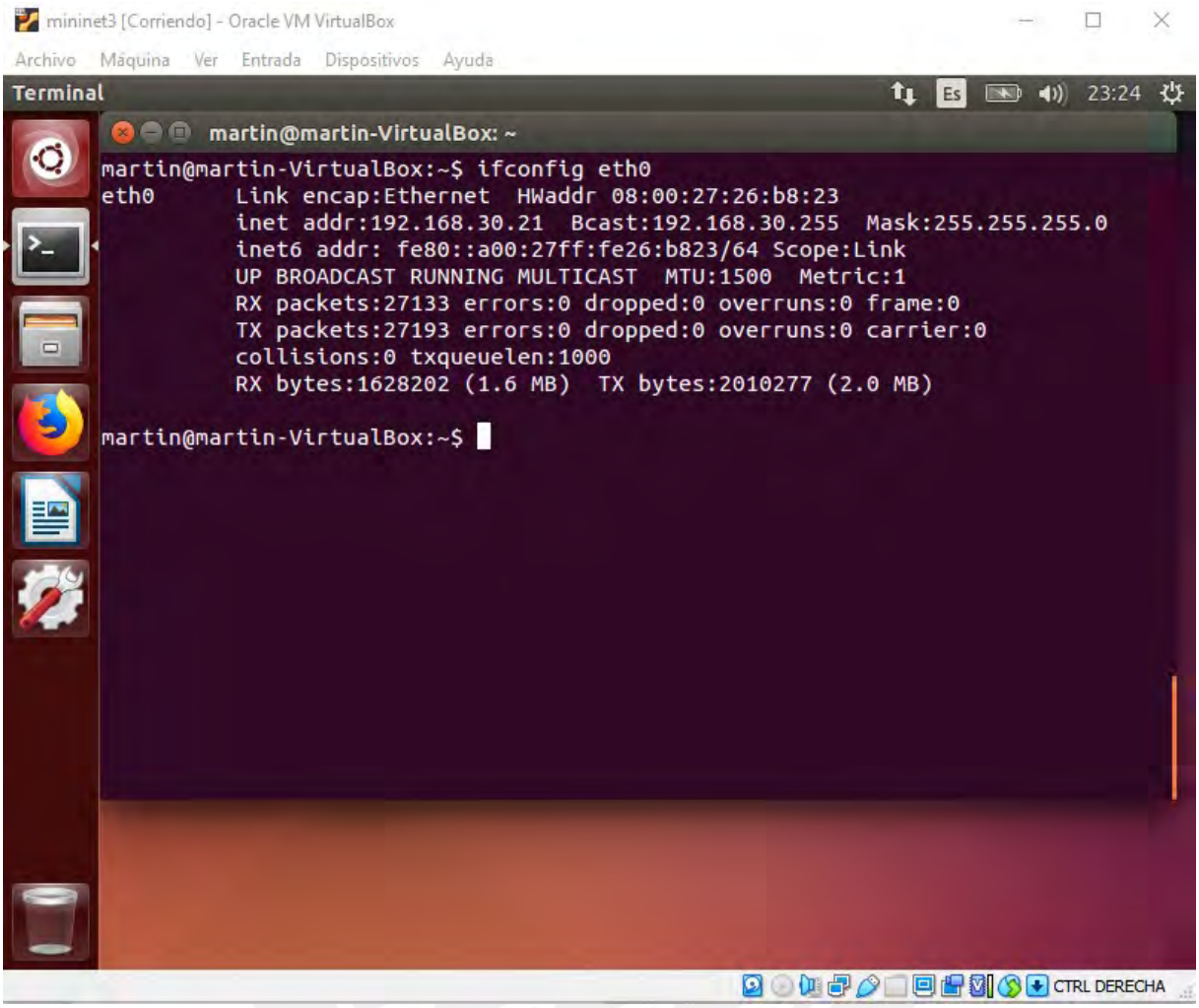

Figura 27: Configuración IP de la VM del Mininet [Elaboración Propia].

La simulación de la topología diseñada, se podía trabajar en Miniedit o podía programarse en un archivo ejecutable de Python. Se optó por la segunda opción para manejar redes más grandes mediante un script. En la Figura 28 se muestra el programa realizado para la simulación de la topología.

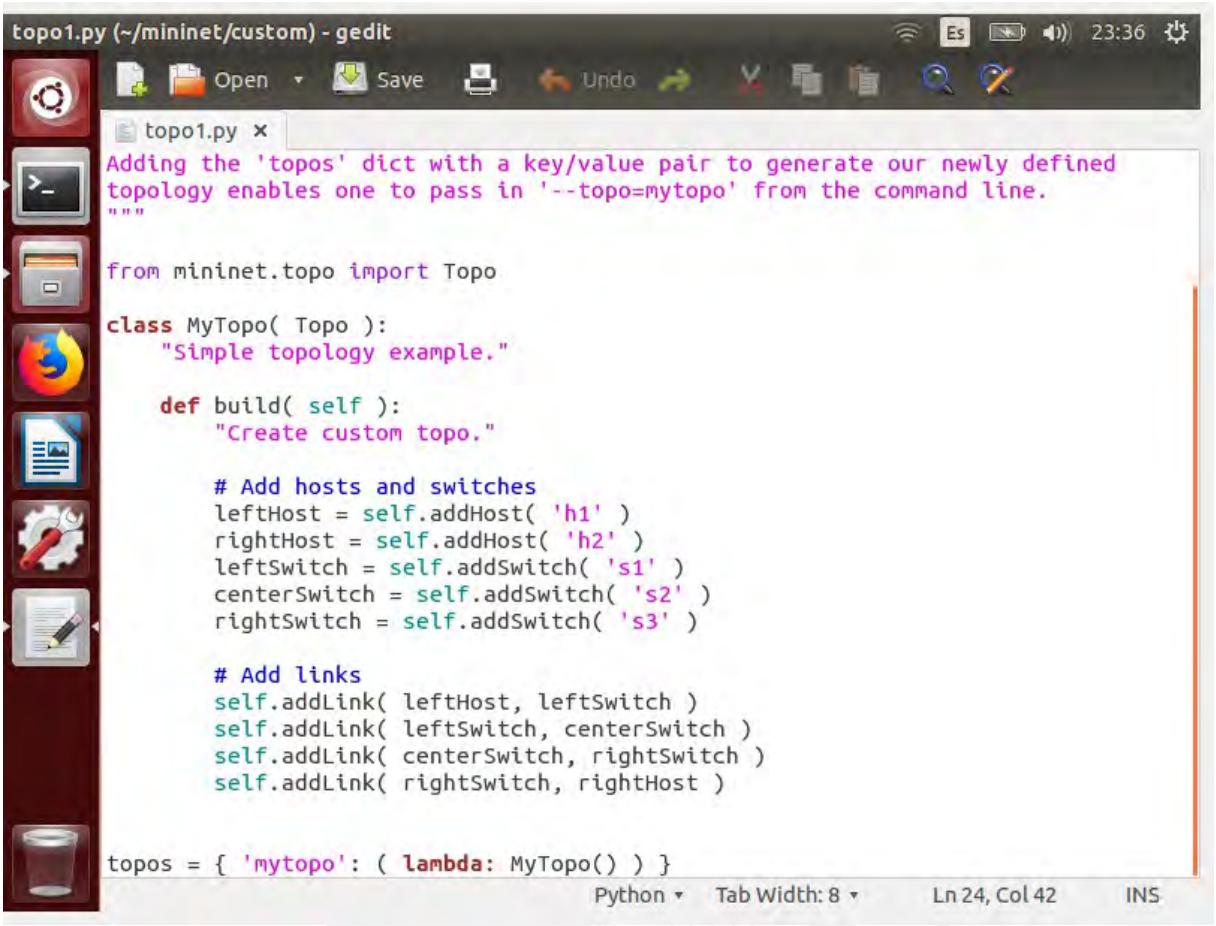

Figura 28: Código de la topología de la red [Elaboración Propia].

Finalmente, para controlar la topología desde un controlador externo, se ejecuta el comando que se observa en la Figura 29. Además, en dicha imagen se puede observar, que la topología ha sido simulada correctamente, al probar el comando pingall en mininet.

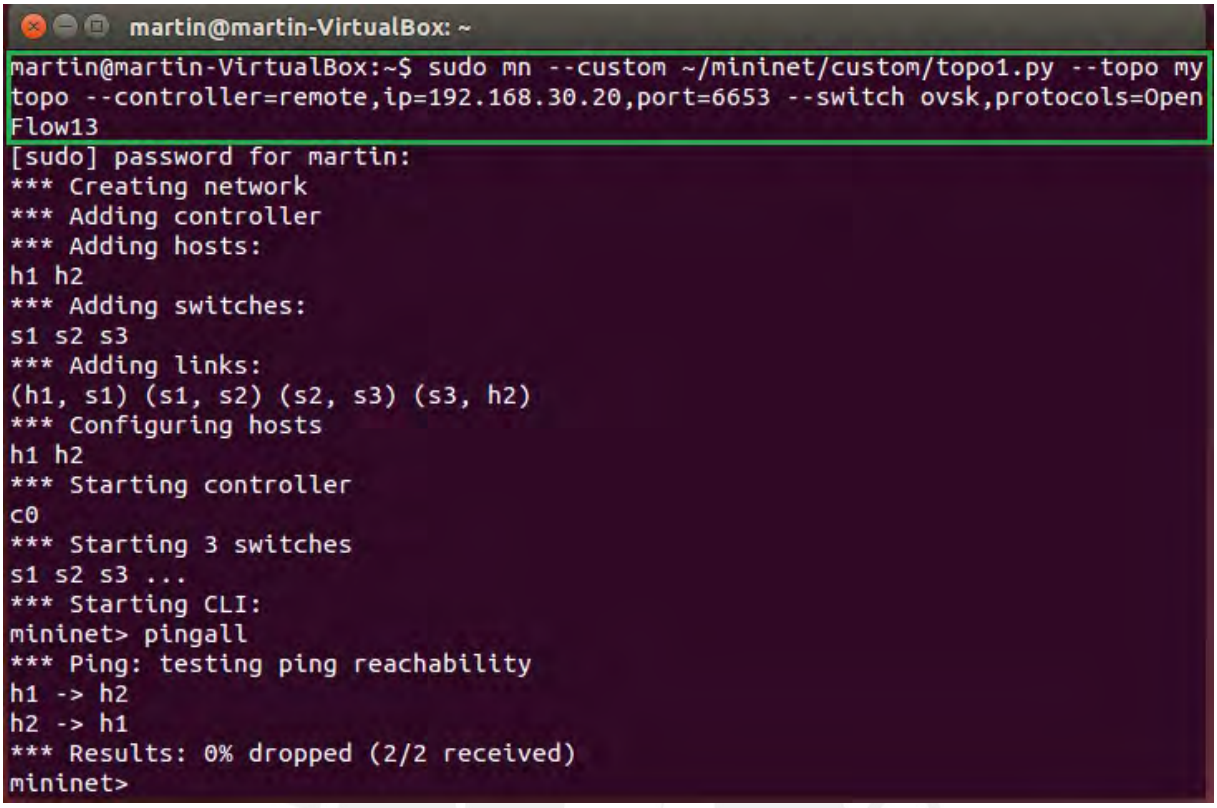

Figura 29: Simulación de la Topología en Mininet [Elaboración Propia].

3.3.4. Interfaz Web del Controlador

Como se mencionó en la arquitectura del controlador Floodlight, este tiene un módulo WEB-UI, es decir, mediante una interfaz de usuario web, se puede observar la topología que el controlador está controlando, y el estado del mismo. En la Figura 30, se observa que el controlador con IP 192.168.1.81 se encuentra controlando la topología simulada en mininet, lo que permite comprobar, que el escenario de pruebas se encuentra listo para realizar una simulación entre ambas máquinas virtuales VM.

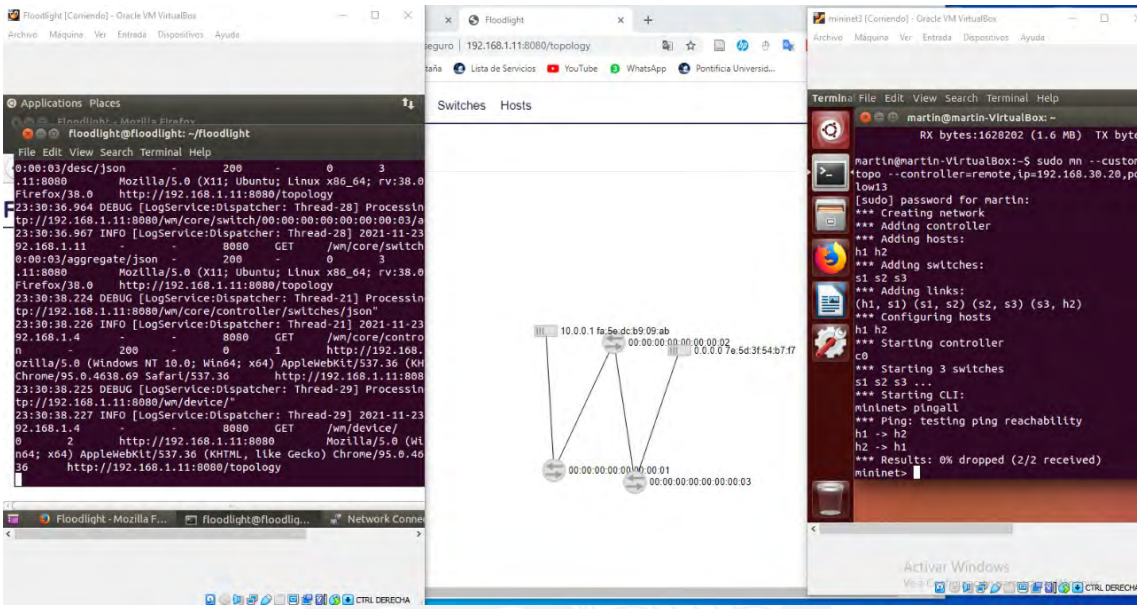

Figura 30: Funcionamiento de las Máquinas Virtuales [Elaboración Propia].

3.4. Topología planteada para redes de accesos de ultima milla

Para los diseños de las redes de accesos comunitarias de última milla se planteó una topología Punto-Multipunto, la cual usa los nodos centrales que se encargan de comunicarse con los demás puntos lejanos de la red. Es en este nodo central donde se planea agregar el controlador SDN para que de esta manera el controlador pueda comunicarse con los distintos componentes de la red que utilizan el protocolo OpenFlow.

La arquitectura de la red acceso comunitaria de última milla utiliza el mismo modelo de arquitectura que el de la tecnología SDN. En primer lugar, en la capa de infraestructura encontramos los 5 conmutadores OpenFlow, que en este caso serían los Access Point, los cuales nos permiten una comunicación inalámbrica. Cada conmutador OpenFlow hace referencia a una zona rural aledaña a la capital de cada distrito, la cual contiene los clientes (hosts) que en este caso serían las máquinas VM conectadas a su propio conmutador OpenFlow. En segundo lugar, en la capa de control se encuentra el controlador SDN, que en este caso sería el controlador Floodlight. Finalmente, en la capa de aplicación se encuentran las aplicaciones realizadas por programación independiente, las cuales se utilizarán para el monitoreo de la red.

En la Figura 31 se observa la topología que se ha planteado.

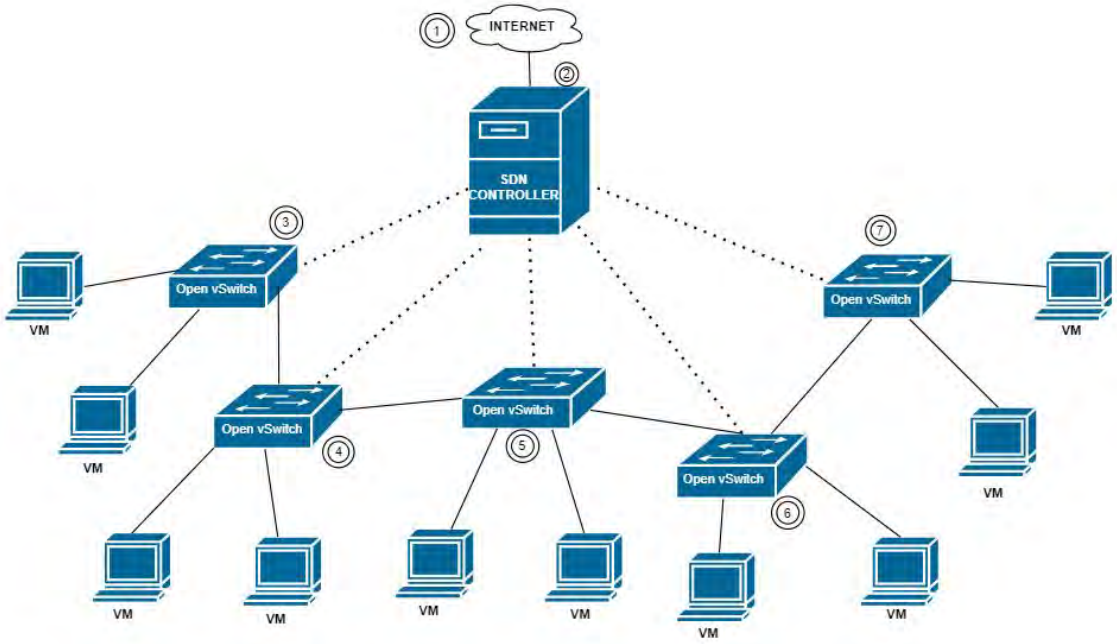

Figura 31: Topología para la red de acceso [Elaboración Propia].

3.4.1. Simulación de la red comunitaria de acceso de última milla con Floodlight y Mininet

Arquitectura de la red en un archivo ejecutable Python

En la Figura 32 se observa el programa de Python que permitirá simular la arquitectura de la red de acceso comunitaria de última milla planteada en la Figura 31 en un entorno SDN

```
from mininet.topo import Topo
class MyTopo( Topo ):
    def build( self ):
        "Create custom topo."
        # Add hosts and switches
        Host1 = self.addHost( 'h1' )
        Host11 = self.addHost( 'h11' )
        Host2 = self.addHost( 'h2' )
        Host21 = self.addHost( 'h21' )
        Host3 = self.addHost( 'h3' )
        Host31 = self.addHost( 'h31' )
        Host4 = self.addHost( 'h4' )
        Host41 = self.addHost( 'h41' )
        Host5 = self.addHost( 'h5' )
        Host51 = self.addHost( 'h51' )
        Switch1 = self.addSwitch( 's1' )
        Switch2 = self.addSwitch( 's2' )
        Switch3 = self.addSwitch( 's3' )
        Switch4 = self.addSwitch( 's4' )
        Switch5 = self.addSwitch( 's5' )
        # Add links
        self.addLink( Host1, Switch1 )
        self.addLink( Host11, Switch1 )
        self.addLink( Switch1, Switch2 )
        self.addLink( Switch2, Host2 )
        self.addLink( Switch2, Host21 )
        self.addLink( Switch2, Switch3 )
        self.addLink( Switch3, Host3 )
        self.addLink( Switch3, Host31 )
        self.addLink( Switch3, Switch4 )
        self.addLink( Switch4, Host4 )
        self.addLink( Switch4, Host41 )
        self.addLink( Switch4, Switch5 )
        self.addLink( Switch5, Host5 )
        self.addLink( Switch5, Host51 )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

Figura 32: Código de Python de la Arquitectura de Red de acceso [Elaboración Propia].

Simulación en Mininet

En la Figura 33 se observa la simulación en Mininet de la Arquitectura de Red de acceso, al hacer pingall se observa la conectividad entre todos los hosts.

```
martin@martin-VirtualBox:~$ sudo mn --custom ~/mininet/custom/topo2.py --topo my
topo --controller=remote,ip=192.168.30.20,port=6653 --switch ovsk,protocols=Open
Flow13
[sudo] password for martin:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h11 h21 h31 h41 h51
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s1) (h11, s1) (s1, s2) (s2, h2) (s2, h21) (s2, s3) (s3, h3) (s3, h31) (s3,
s4) (s4, h4) (s4, h41) (s4, s5) (s5, h5) (s5, h51)
*** Configuring hosts
h1 h2 h3 h4 h5 h11 h21 h31 h41 h51
*** Starting controller
c0
*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h11 h21 h31 h41 h51
h2 -> h1 h3 h4 h5 h11 h21 h31 h41 h51
h3 -> h1 h2 h4 h5 h11 h21 h31 h41 h51
h4 -> h1 h2 h3 h5 h11 h21 h31 h41 h51
h5 -> h1 h2 h3 h4 h11 h21 h31 h41 h51
h11 -> h1 h2 h3 h4 h5 h21 h31 h41 h51
h21 -> h1 h2 h3 h4 h5 h11 h31 h41 h51
h31 -> h1 h2 h3 h4 h5 h11 h21 h41 h51
h41 -> h1 h2 h3 h4 h5 h11 h21 h31 h51
h51 -> h1 h2 h3 h4 h5 h11 h21 h31 h41
*** Results: 0% dropped (90/90 received)
mininet>
```

Figura 33: Simulación en Mininet de la Arquitectura de Red de acceso [Elaboración Propia].

Visualización de la Arquitectura desde el Controlador

En la Figura 34 se observa la Arquitectura desde el controlador en la Interfaz de Usuario Web que este provee. Lo cual, indica que las gestiones de seguridad de la red ya se pueden realizar desde el controlador.

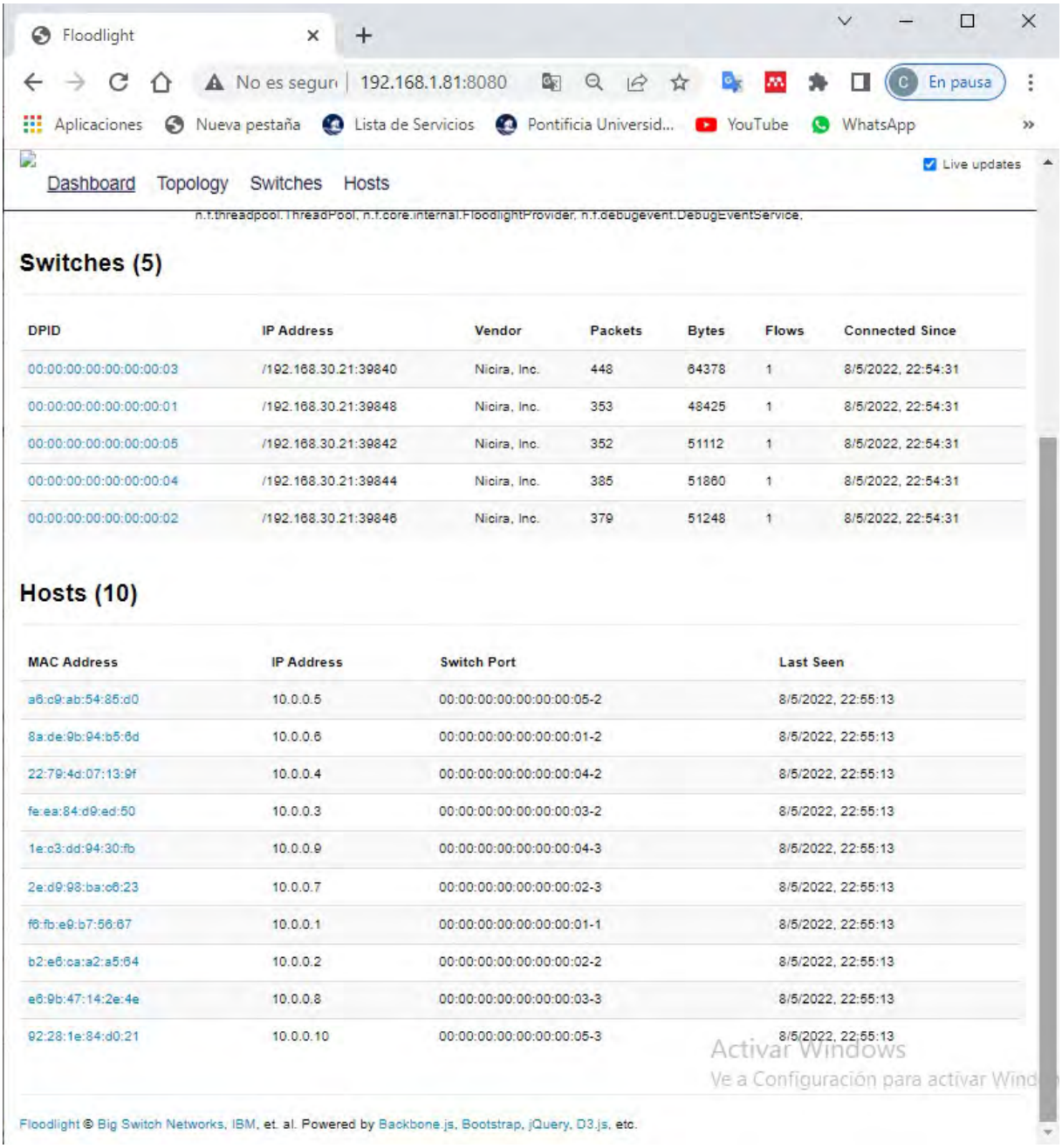

Figura 34: Visualización de la Red desde el Controlador [Elaboración Propia].

3.5. Implementación de las instrucciones de gestión en la plataforma de emulación

La implementación se realizará según lo especificado, para ello en el controlador se debe utilizar una lista de instrucciones, las cuales nos permitirán obtener una monitorización de toda la red que se implementó en la plataforma de emulación.

En la Figura 35 se observa las instrucciones que se han implementado:

Monitoreo de Rango de puertos y Trafico en la interfaz de la red

curl http://localhost:8080 /wm/core/switch/all/port/json | python -m json.tool

Monitoreo de Ancho de Banda

curl http://localhost:8080 /wm/core/switch/all/port-desc/json | python -m json.tool

Monitoreo de Cantidad de Host y Switches

curl http://localhost:8080 /wm/device/ | python -m json.tool

curl http://localhost:8080 /wm/topology/switchclusters/json | python -m json.tool

Resumen del controlador (n.° de conmutadores, n.° de enlaces, etc.)

curl http://localhost:8080 /wm/core/controller/summary/json | python -m json.tool

Enumerar flujos estáticos para un conmutador o todos los conmutadores

curl http://localhost:8080 /wm/staticflowpusher/list/00:00:00:00:00:00:00:03/json

Agregar un flujo Estatico a cada Switch

curl -X POST -d '{"switch":"00:00:00:00:00:00:00:03", "name":"flow-mod-1", "cookie":"0", "priority":"32768", "in_port":"1","active":"true", "actions":"output=2"}' http://localhost:8080/wm/staticflowpusher/json

Figura 35: Instrucciones a Implementar [Elaboración Propia].

4. Simulación, Pruebas y Resultados

Una vez establecido el diseño del sistema, en este capítulo se va a realizar la simulación y las pruebas para verificar el correcto funcionamiento del monitoreo de la red. Se presentarán pruebas donde se verificará el accionamiento del módulo de static flow entry pusher de forma centralizada a toda la red.

4.1. Simulación de la red SDN

Según lo señalado en el diseño del sistema, se simula en el entorno de emulación Mininet, la topología SDN propuesta que se muestra en la Figura 19.

Las Figuras 25 y 29 nos muestran la topología simulada en mininet, la cual ha sido simulada con un controlador configurado de forma remota. Además, de que también se realizó la prueba de conexión de toda la red.

Después de realizar toda la simulación, en la Figura 36 se podrá observar la interfaz web del controlador donde se encuentra la topología que ha sido simulada en Mininet y que está siendo controlada por el controlador Floodlight también simulado.

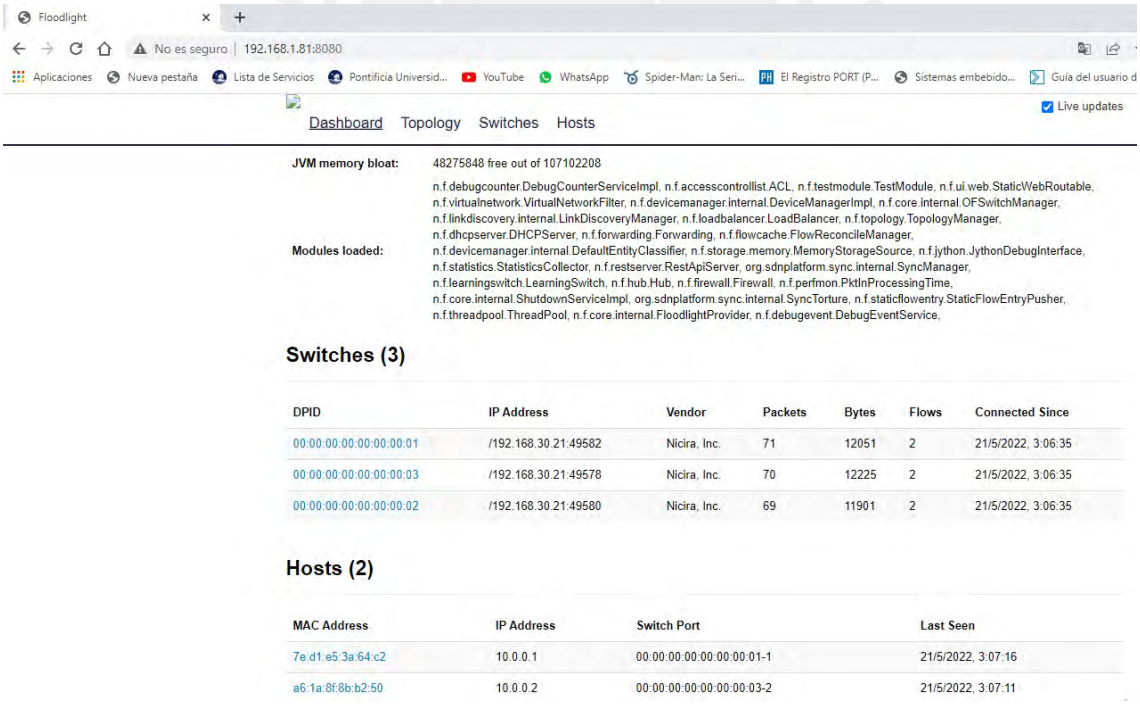

Figura 36: Visualización de la Red desde el Controlador [Elaboración Propia].

La topología simulada nos permitirá realizar pruebas de monitorización. A continuación, se presentan las pruebas realizadas en donde se podrá observar la obtención de los parámetros de monitorización previamente mencionados. Esto evidenciará que el controlador de forma remota y centralizada aplica las instrucciones necesarias para una correcta monitorización de la red y de esta manera optimizar su gestión.

4.2. Pruebas Realizadas

4.2.1. Monitoreo de Cantidad de Hosts y Conectividad

En la Figura 37 se puede observar en el terminal de Linux de la computadora donde se encuentra el controlador floodlight donde se utilizó la REST API core/controller/summary cuyo propósito es obtener un resumen del controlador: número de enlaces, número de conmutadores, número de hosts, etc.

Para la comprobación de la conectividad floodlight lo que realiza es una validación de cada servidor que está conectado a nuestra red a nivel de host, lo que significa que cada servidor diferente se visualizara como un host que está conectado a la red, de esta manera es como se realiza la prueba de conectividad para la red.

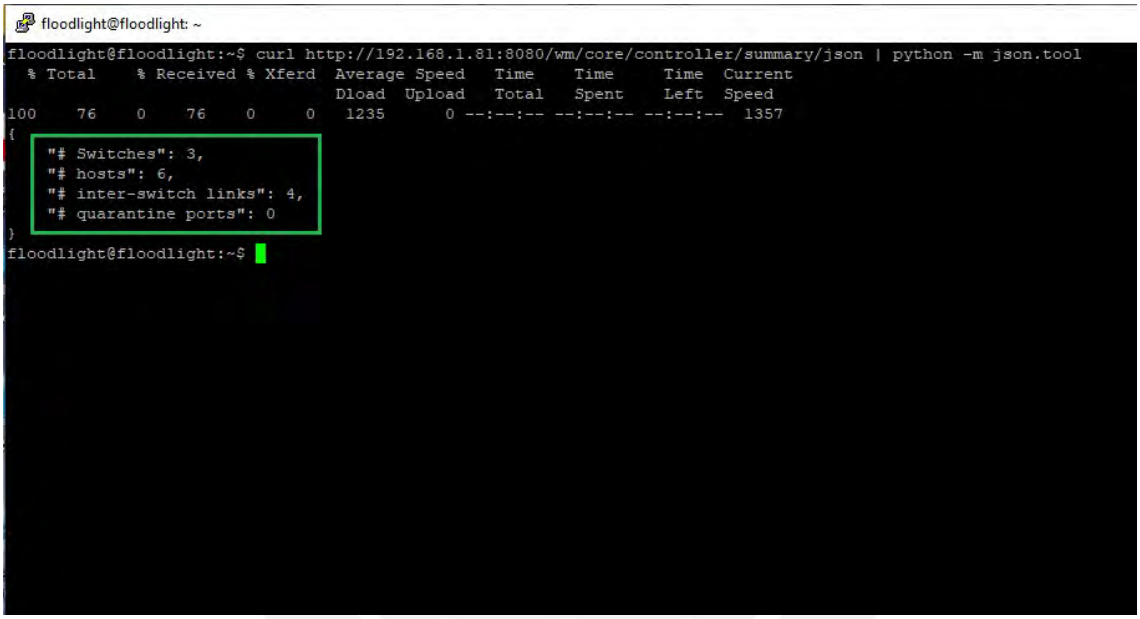

Figura 37: Monitoreo de Cantidad de Hosts y Conectividad en el Controlador [Elaboración Propia].

Como ultima observación que se realiza de la prueba de conectividad y el monitoreo de cantidad de hosts es que floodlight por defecto nos incluye 4 direcciones IPv4 que se visualizan como 4 hosts. Sin embargo, esto es un error del controlador, ya que cuando se desea visualizar la topología de la red en su interfaz web se podrá observar la topología que se diseñó para las pruebas, la cual solo consistía en 2 hosts y en 3 switches. Por esta razón, en el número de hosts que se puede ver en la imagen aparece 6, los cuales son los 2 hosts que definimos en la topología de prueba, más los 4 hosts que por defecto crea el controlador.

4.2.2. Monitoreo de Rango de Puertos

Para el monitoreo de rango de puertos se utilizó la REST API topology/links, la cual nos muestra los enlaces directos y túneles descubiertos por paquetes LLDP. Estos paquetes LLDP son el protocolo de descubrimiento de capa de vínculos, el cual es el método estándar de la industria que permite a los dispositivos anunciar sus capacidades u otras informaciones de la red en la que se encuentren.

Para este caso, en la Figura 38 se puede observar los puertos de cada switch y la manera en la que están conectados por medio de los links. En el primer caso, podemos observar que el switch 3 cuyo código DPID es el 00:00:00:00:00:00:00:03 está conectado al switch 2 (00:00:00:00:00:00:00:02) por su puerto 1 y el switch 2 al switch 3 por su puerto 2. En el otro caso, tenemos que el switch 2 está conectado al switch 1(00:00:00:00:00:00:00:01) por su puerto 1 y el switch 1 al switch 2 por su puerto 2 obteniendo de esta forma el rango de puertos de todos los switches de la red.

También, cabe recalcar que el puerto 1 del switch 1 y el puerto 2 del switch 3 están conectados a los hosts que se crearon cuando se simuló la topología.

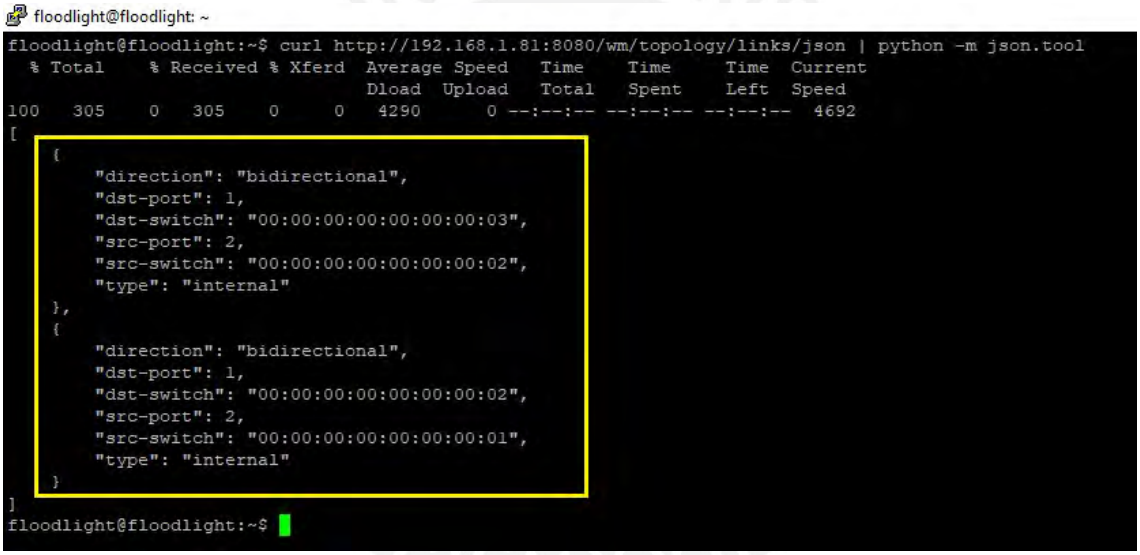

Figura 38: Monitoreo de rango de puertos en el Controlador [Elaboración Propia].

4.2.3. Monitoreo de Ancho de Banda de la Red

En el caso del monitoreo de ancho de banda, se utilizó la REST API switch/switchId/port-desc/ en donde switchld significa el código DPID del switch que se desea obtener la información. Esta REST API nos permite recuperar estadística más descriptiva de los puertos del switch en el que se decidió realizar la toma de información. Para este caso se decidió realizar la prueba en cada uno de los 3 switches, ya que se desea conocer toda la información posible de ellos, en este caso el ancho de banda de los puertos de cada switch.

En las Figuras 39, 40 y 41 se observa toda la información descriptiva de los 3 switches por separado; esta información detalla los puertos de cada switch, la interfaz que está conectada cada switch por su puerto con su respectivo nombre y por último el ancho de banda el cual es de 10000000 bits por segundo o también 10 Mbps, el cual en este

caso es igual para todos los puertos de los switches, ya que los 3 switches Openflow son iguales para este caso.

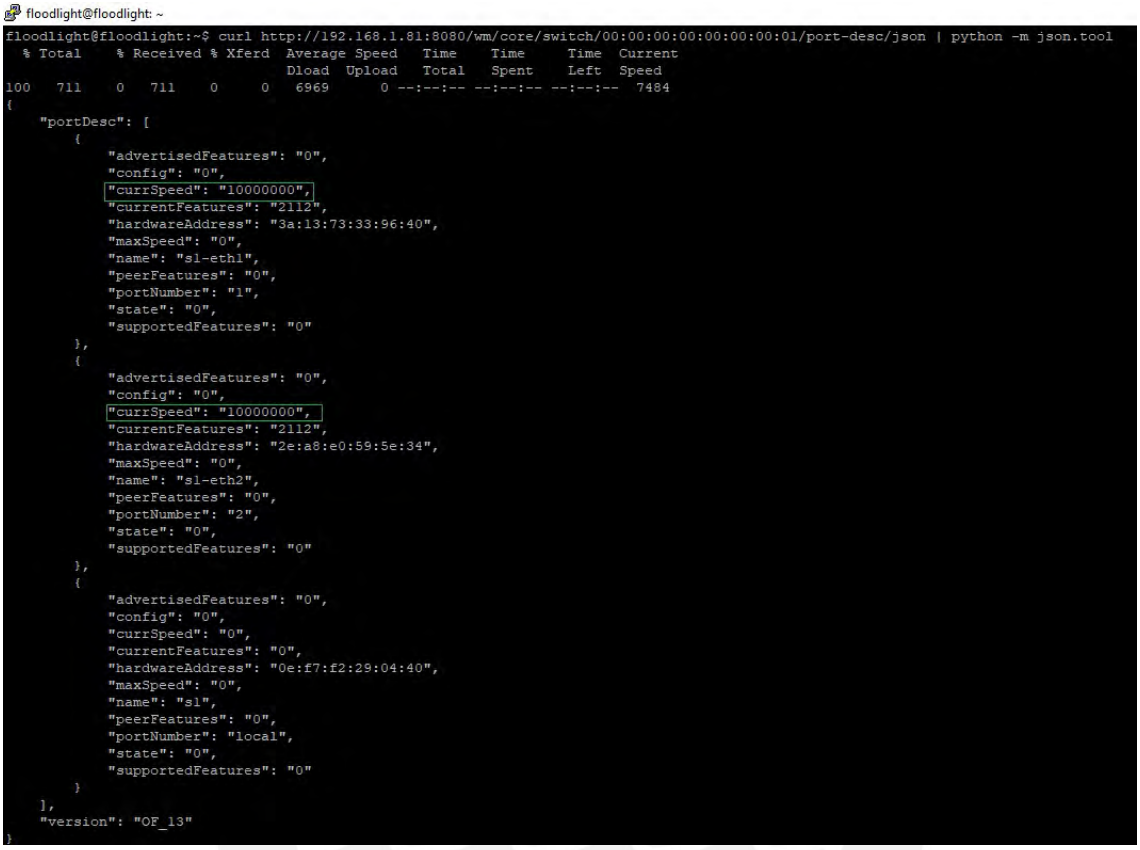

Figura 39: Monitoreo del ancho de banda del Switch 1 [Elaboración Propia].

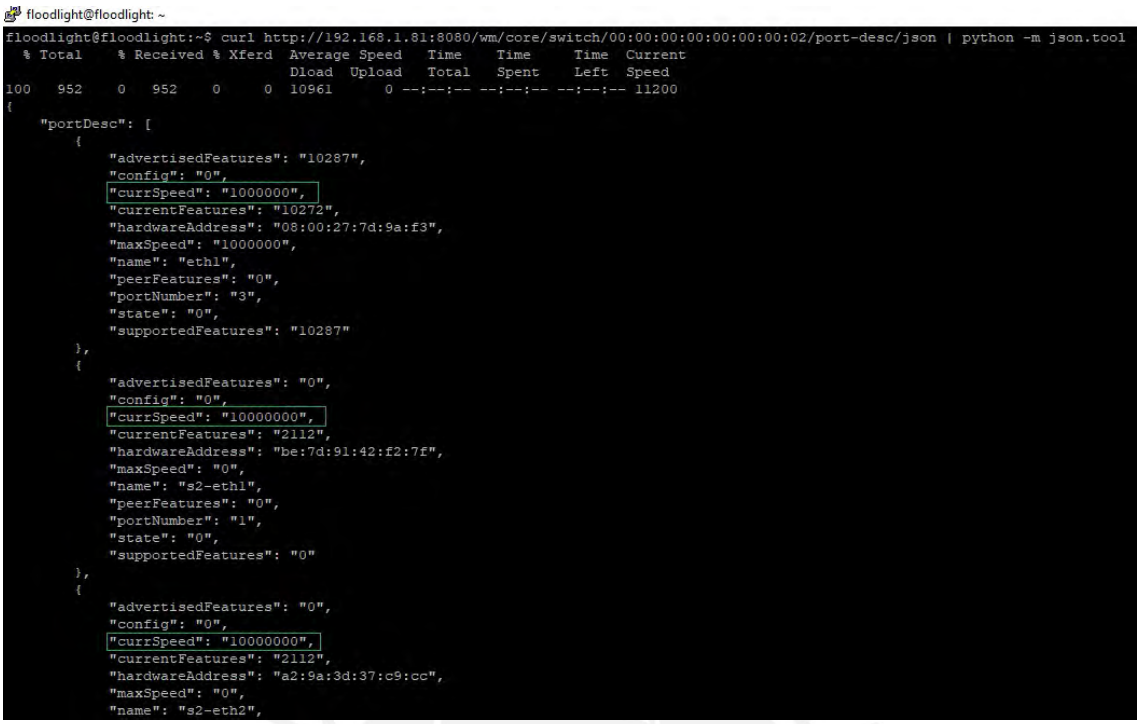

Figura 40: Monitoreo del ancho de banda del Switch 2 [Elaboración Propia].

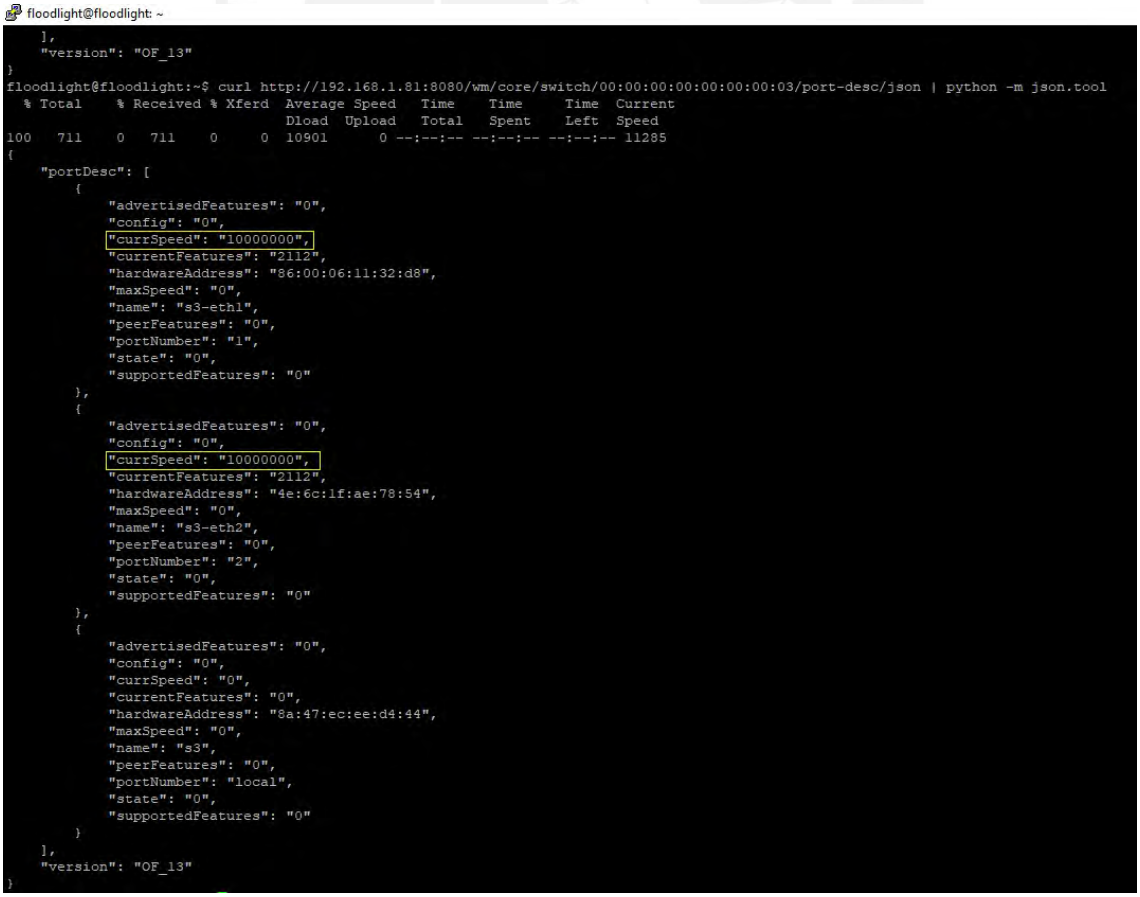

Figura 41: Monitoreo del ancho de banda del Switch 3 [Elaboración Propia].

4.2.4. Tráfico de interfaces de Red

Para la toma del monitoreo de tráfico de las interfaces de la red se realizó una prueba de conectividad Ping como se puede observar en la Figura 42, la cual es una utilidad de diagnóstico en redes de computadoras que comprueba el estado de la comunicación del anfitrión local con uno o varios equipos remotos de una red que ejecuten IP. Este utiliza el envío de paquetes ICMP de solicitud (ICMP Echo Request) y de respuesta (ICMP Echo Reply). Mediante esta utilidad puede diagnosticárse la transmisión y recepción de bytes y de paquetes en todos los interfaces de la red.

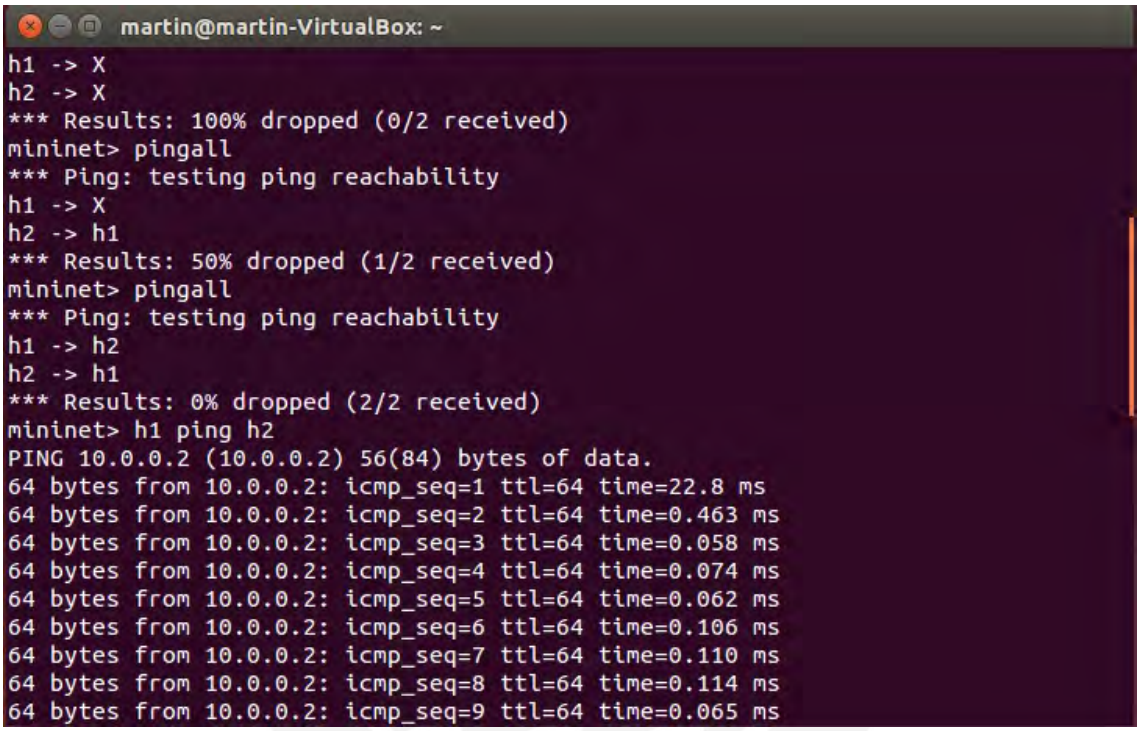

Figura 42: Ejecución de ping del Host 1 al Host 2 [Elaboración Propia].

Para este caso se utilizó la REST API switch/switchId/port/, la cual nos entrega una lista con la información de los paquetes y bytes que transcurren por las interfaces de la red cuando se realiza un envío de datos de cualquier punto de esta. Al igual que en el monitoreo de ancho de banda, para esta prueba se realizó una toma de datos de manera individual para cada switch, esto con el objetivo de poder apreciar mejor la recopilación de información que atraviesa por cada uno de ellos. En las Figuras 43,44 y 45 se pueden observar imágenes de comparación de cada uno de los switches. La imagen que posee los cuadros amarillos muestra una toma información de los switches cuando solamente se ha empezado a simular la red en mininet, cada cuadro amarillo muestra la transmisión y recepción de paquetes y bytes que se están realizando en los interfaces de la red antes de que se realiza la prueba de conectividad por medio del Ping, una vez realizada esta acción en la imagen con los cuadros de color verde se puede observar que tanto la transmisión de los paquetes y bytes empieza a incrementar, debido a los paquetes que el Ping está enviando de un host a otro. Es de esta manera en que se puede visualizar el tráfico de los interfaces de la red por medio de las REST API de este controlador

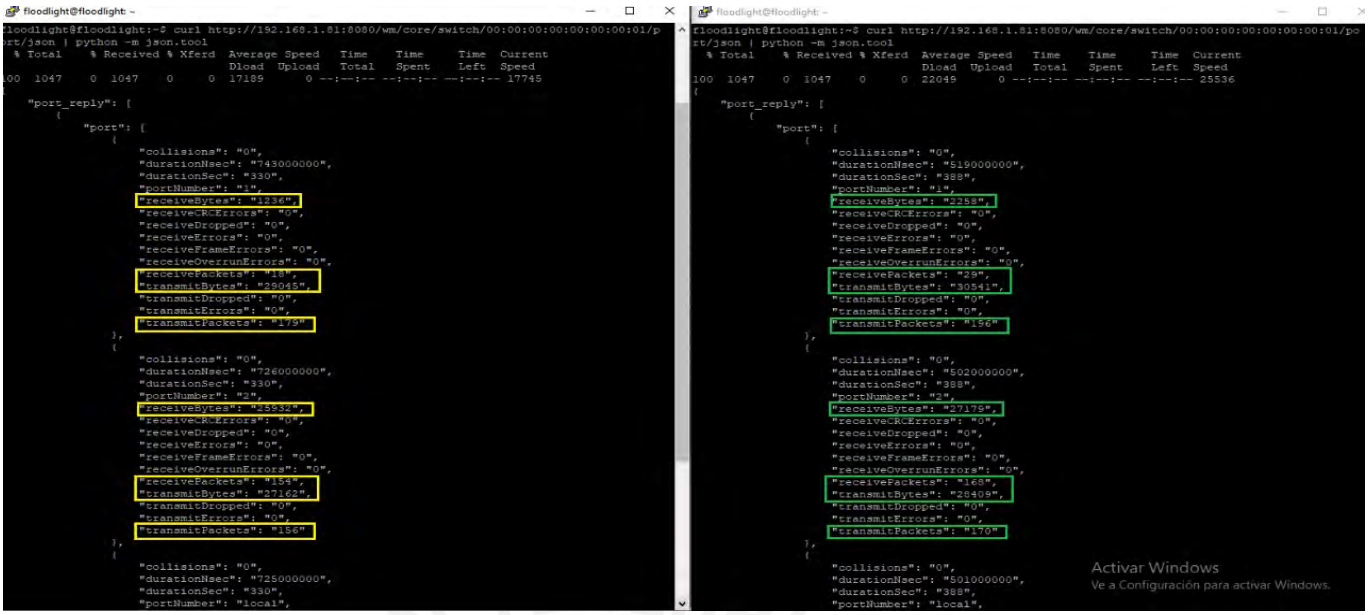

Figura 43: Trafico en las interfaces del Switch 1 [Elaboración Propia].

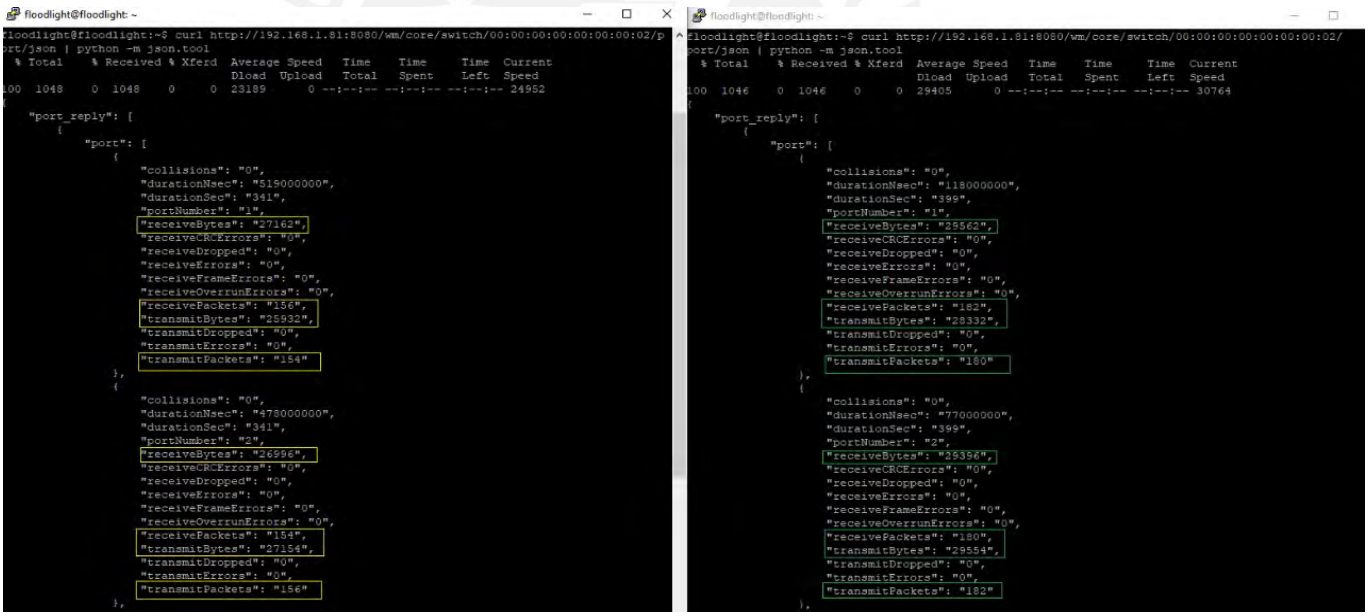

Figura 44: Trafico en las interfaces del Switch 2 [Elaboración Propia].

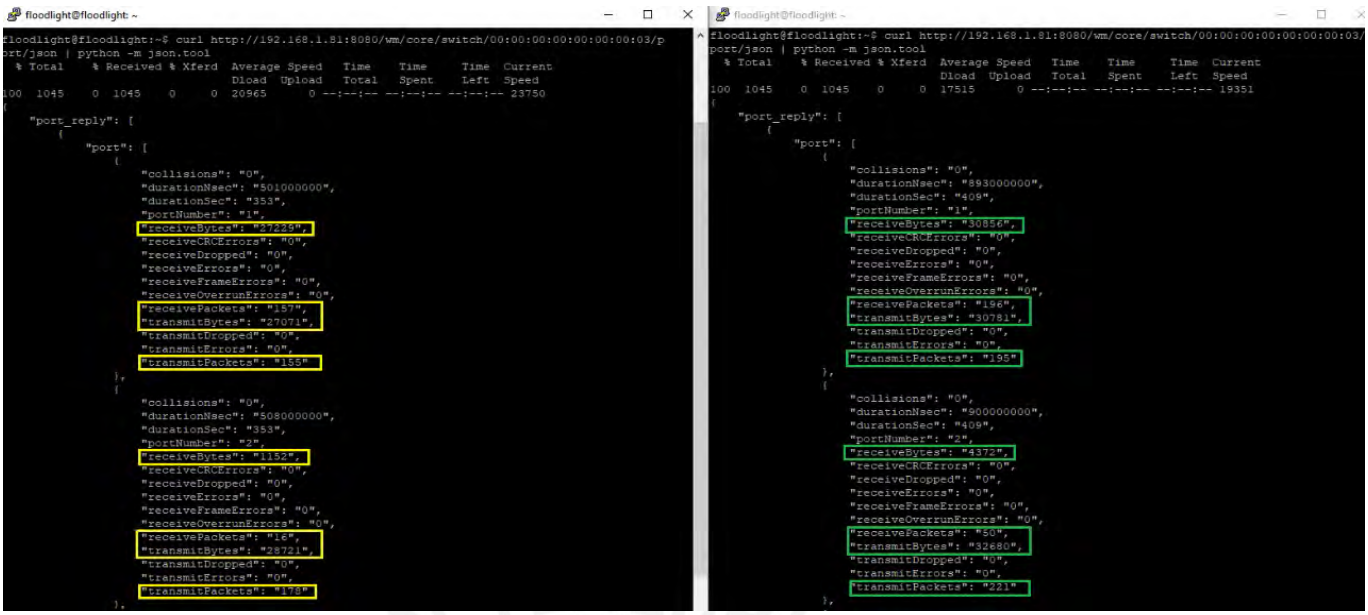

Figura 45: Trafico en las interfaces del Switch 3 [Elaboración Propia].

4.2.5. Estadísticas de interfaz de Red

Para la obtención de las estadísticas de las interfaces de la red, a diferencia de lo que se realizó en el monitoreo del tráfico de las interfaces de la red en donde se observaba el envío y recepción de bytes y paquetes, se optó por primero agregar un flujo estático, que posea otras características aparte del envío de paquetes y bytes, en un switch en específico y luego realizar la enumeración de flujos por medio de una REST API, la cual nos mostraría parámetros distintos a lo que se observó en el monitoreo de tráfico de las interfaces como la dirección del envío de paquetes por los puertos de cada switch, la acción que realiza en cada puerto, el grupo que pertenece el flujo, su prioridad, etc. Para esto se optó por escoger el switch 2 como el switch en donde se realizará la prueba, ya que este es el switch central de la topología de la red planteada, por lo que es en este switch que se podrá observar mejor el transcurso de los distintos flujos que se agregue en la red.

En la Figura 46 se puede observar justo la toma de los datos del flujo agregado en el switch 2, el cual se realizó previamente con una instrucción POST, en la cual detalla algunas características del transcurso del flujo: entrada del flujo por el puerto 1 y salida por el puerto 2, la prioridad del flujo ingresado y el nombre del flujo. Este resultado muestra que en este caso el envío de paquetes y de bytes es unidireccional, a diferencia de lo que se realizó por el tráfico de las interfaces.

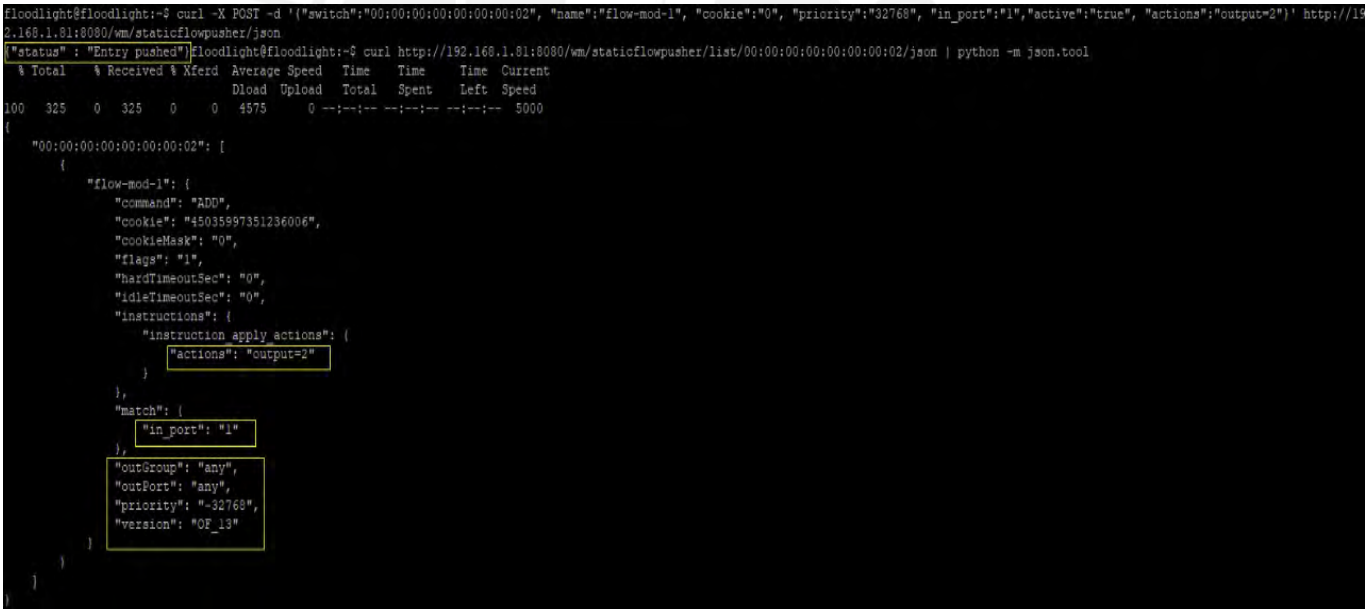

Figura 46: Lista de los flujos estáticos ingresados en el Switch 2 [Elaboración Propia].

Finalmente, en la Figura 47 se puede observar con otra REST API la cantidad de bytes que se envían en los 2 puertos del switch 2, tanto para la salida como en la entrada, obteniendo de esta manera una estadística muy amplia de los interfaces de la red.

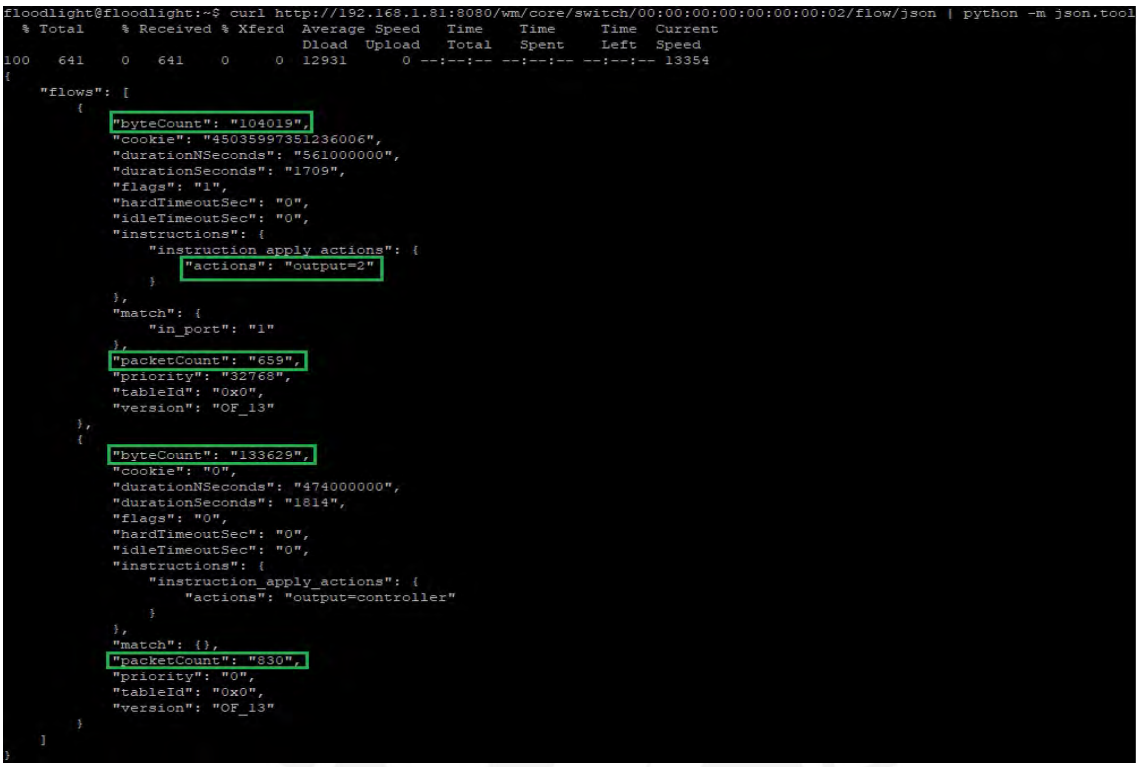

Figura 47: Estadísticas en las interfaces del Switch 2 [Elaboración Propia].

4.2.6. Monitoreo de traceroute

Para el monitoreo de traceroute se tuvo que realizar una configuración externa para que la red pudiera tener acceso a internet, ya que de esta manera se podría realizar un seguimiento de traceroute al servidor DNS de Google cuya dirección es 8.8.8.8. Para poder realizar el seguimiento primero se tuvo que agregar un adaptador de red NAT en la computadora virtual que tenía el mininet, es de esta manera que ahora la máquina mininet tendría 2 interfaces de red, una de red interna y otra de red NAT. Después de simular la topología de red virtual en mininet se tuvo que conectar la interfaz que poseía la red NAT con uno de los switches de nuestra topología, esto se pudo realizar por medio del comando ovs-vsctl add-port s2 eth1, el cual genera una conexión de la interfaz eth1 que es de la red NAT al switch 2 de nuestra red. Por este medio, nuestra red ahora tiene acceso a internet, por lo que su seguimiento de traceroute es ahora posible. Para el seguimiento de traceroute primero se tuvo que ingresar al terminar independiente de uno de nuestros hosts, esto es posible por medio del comando xterm de mininet, la cual nos abre un terminal de la máquina que elegimos, en la Figura 48 se puede observar que se optó por el host 1 y en su terminal se colocó la instrucción de traceroute al servidor de Google 8.8.8.8. También se puede observar las diferentes rutas que están siguiendo los paquetes enviados por el traceroute

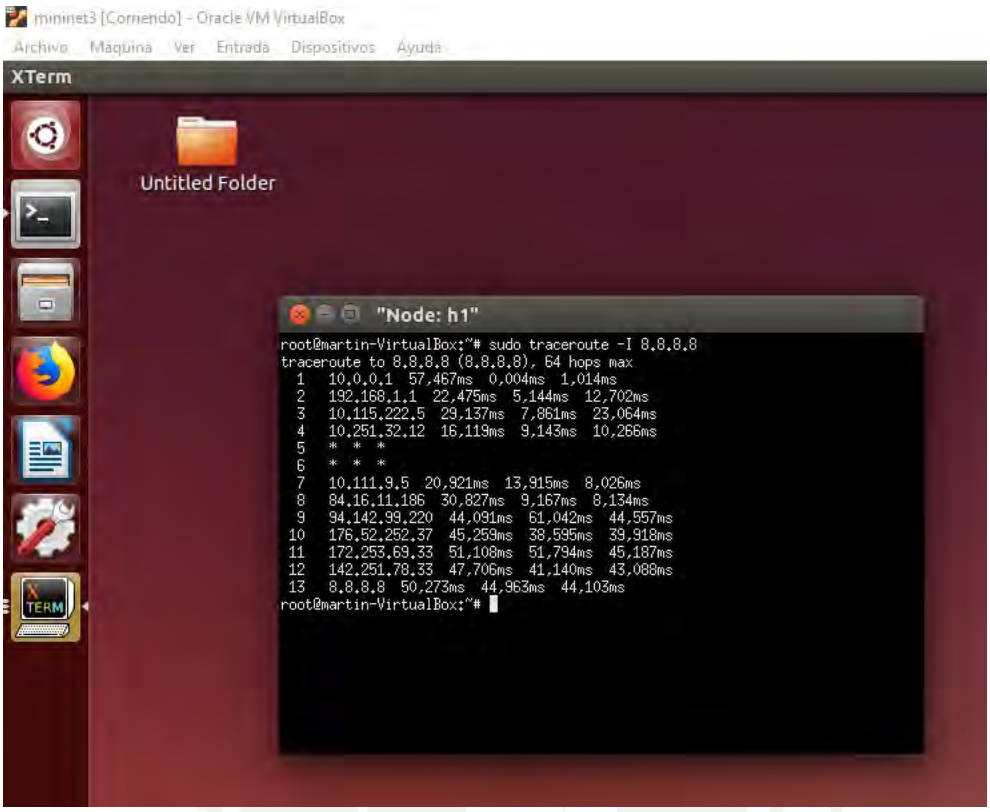

Figura 48: Ejecución del traceoute del Host1 hacia DNS de Google [Elaboración Propia].

Finalmente, para la finalización del monitoreo del traceroute se optó al igual que monitoreo de tráfico de interfaces por la REST API switch/00:00:00:00:00:00:00:02/port/, solo que en este caso se centró principalmente en el switch 2, ya que este posee la conexión hacia internet. En la Figura 49 se observa también dos imágenes de comparación, solo que en este caso se puede observar que la comparación ahora es de un nuevo puerto que se agregó (puerto 3), el cual es el que conecta el swtich 2 hacia internet. Al igual que en el caso del tráfico de interfaces, se observa que el puerto 3 antes de realizar la instrucción de traceroute está enviando y recibiendo paquetes y bytes como se puede ver en los cuadros amarillos, pero después de realizar la instrucción se observa un incremento de estos, ya que la instrucción está enviando y recibiendo los paquetes y bytes hasta el servidor de Google como se puede observar en los cuadros verdes. De esta manera finalizamos y cumplimos con el objetivo de realizar el monitoreo del traceroute de la red hacia un servidor externo a esta.

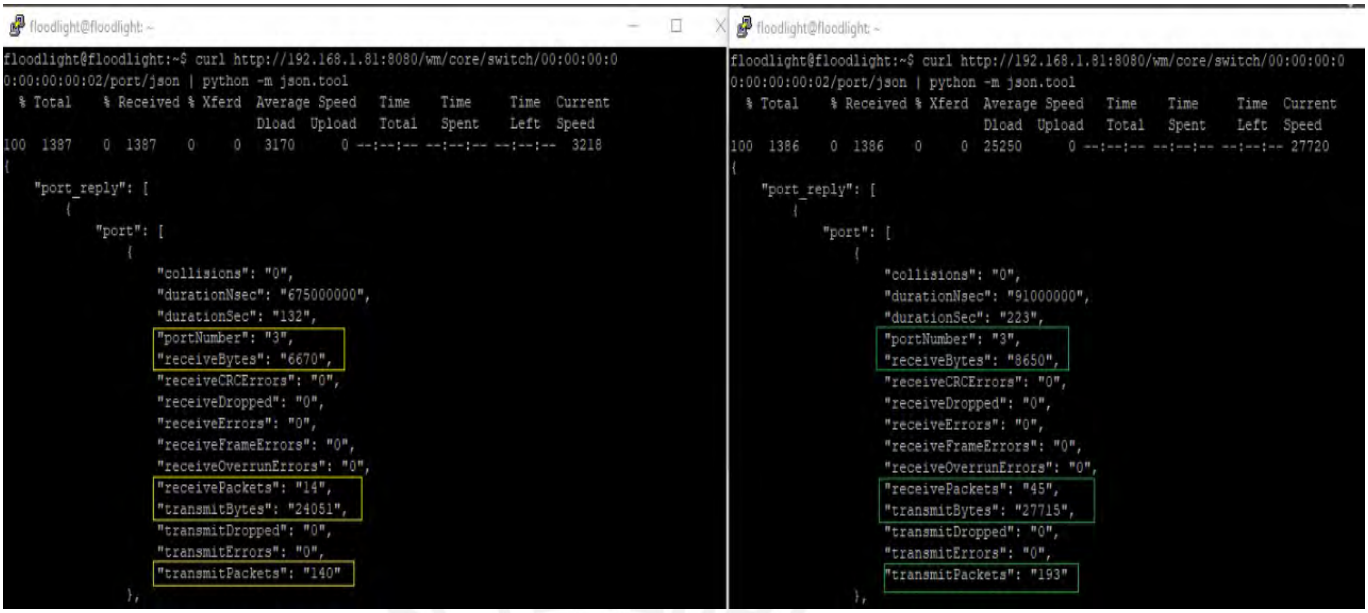

Figura 49: Monitoreo de Traceroute en el Switch 2 [Elaboración Propia].

4.3. Análisis de resultados

De los resultados obtenidos en las secciones anteriores se procede al análisis correspondiente, el cual luego de su debida extrapolación se observa que en todos los escenarios de prueba se obtiene una correcta recopilación de datos de toda la red. Todo esto, de acuerdo a las políticas inyectadas por REST API al controlador SDN.

En el monitoreo de cantidad de hosts, conectividad y rango de puertos se puede comprobar con claridad que el controlador floodlight es capaz de brindarnos la información de los parámetros más importantes de la red como la cantidad de switches, servidores, puertos y hosts. Esto pone en evidencia que la tecnología SDN resultaría muy valiosa para este tipo de redes de acceso, ya que nos permite obtener la recopilación de información de la red de forma rápida y eficiente. También, cabe recalcar que el controlador está bien equipado para estas tareas, ya que como se pudo comprobar en la prueba de rango de puertos, este es capaz de utilizar el protocolo LLDP, el cual anuncia las capacidades de cada dispositivo de la red, lo cual si se presentara en un caso real sería de gran relevancia conocer las capacidades de los equipos que se está operando.

Por otro lado, también se pudo comprobar que el controlador es capaz de brindarnos información sobre ciertas características de la red, como es el caso del ancho de banda. Así mismo, logro mostrar el tráfico que se genera en las interfaces de la red junto con las estadísticas de estas, evidenciando de esta manera que el controlador SDN floodlight es capaz de igualar a varios sistemas de monitorización, los cuales son utilizados para poder gestionar la red con la obtención de sus parámetros. Todo esto, se puede resaltar en las pruebas realizadas de monitorización de ancho de banda, tráfico de interfaces de red y estadísticas de interfaces.

Finalmente, estos resultados comprueban la escalabilidad, programabilidad y versatilidad de este nuevo paradigma de redes y su beneficio de utilizarse en una red de acceso de última milla. Además, queda como trabajo pendiente aplicar el paradigma SDN a redes inalámbricas.

5. Conclusiones

- El presente trabajo de tesis demuestra el funcionamiento de la aplicación de la tecnología SDN para la gestión de redes de acceso de última milla mediante la recopilación de los parámetros más importantes de una red, entre uno de los principales es la demanda del tráfico de datos y acceso a internet, similar a lo que suelen hacer varios de los mejores sistemas de monitorización y gestión de redes de otros fabricantes, los cuales buscan problemas de sobrecarga que suceden en la infraestructura de una red.

- También se presenta un marco basado en SDN para redes inalámbricas con capacidades de control de red para permitir monitoreo de tráfico en las interfaces de la red y calidad de servicio de extremo a extremo.

- La implementación de una red de acceso de última milla basado en SDN de acuerdo a la topología de pruebas diseñada en la Figura 19 y desplegada en el entorno de simulación de redes Mininet permitió monitorear el tráfico de datos en las interfaces de la red generado desde varios clientes situados en la misma red. También permitió la obtención de otros parámetros como número de clientes, número de switches, servidores, puertos, etc. Por último, permitió observar el número de paquetes junto con la cantidad bytes que recorrían por las interfaces cuando se le aplicaba una instrucción de envío de datos hasta otro punto de la red o a una parte externa a ella.

- Tal como se aprecia en la sección de Análisis de resultados del capítulo 4 de la presente tesis, se logró el resultado final de la tesis, verificando el objetivo principal de la tesis, así como la comparación de los resultados de SDN y las redes convencionales, logrando mejores resultados en la solución propuesta y adicionalmente se verificó que presenta una adecuada escalabilidad, programabilidad y versatilidad de este nuevo paradigma de redes y sus ventajas al ser utilizadas en una red de acceso de última milla.

6. Referencias

[1] "Red Dorsal Nacional de Fibra Óptica", Ministerio de Transportes y Comunicaciones. https://portal.mtc.gob.pe/comunicaciones/concesiones/red_dorsal/red_dorsal.html (consultado may 14, 2021.)

[2] "Red Dorsal de Fibra Óptica", ProInversión, 2012. [En línea]. Disponible en:https://www.proinversion.gob.pe/MODULOS/LAN/landing.aspx?are=0&pfl=1&lan=13 &tit=red-dorsal-de-fibra- %C3 %B3ptica (consultado may 03, 2021).

[3] H. Campodónico, "El fracaso de la Red Dorsal de Fibra Óptica y la obsesión por las APP", OtraMirada, sep. 30, 2019. http://www.otramirada.pe/el-fracaso-de-la-red-dorsal-de-fibra- %C3 %B3ptica-y-la-obsesi %C3 %B3n-por-las-app (consultado may 14, 2021).

[4] E. San Roman y C. San Roman, "Redes de acceso y transmisión de Fibra Óptica: alternativas de políticas y regulaciones", Pontif. Univ. Católica Perú, pp. 39-48.[En línea]. Disponible en: http://revistas.pucp.edu.pe/index.php/derechoadministrativo/article/download /13517/14143/.

[5] N. E. E. Tarapuez, "Diseño y simulacíon de una Red definida por Software (SDN)", Univ. Cent. Ecuad., may 2016, Consultado: abr. 21, 2021. [En línea]. Disponible en: http://www.dspace.uce.edu.ec/bitstream/25000/6505/1/T-UCE-0011-265.pdf

[6] L. E. Salao Jurado, "Virtualización de las redes FTTH a través de SDN para optimizar su administración", Esc. Super. Politécnica Litoral, Tesis de licenciatura 2017, Consultado: abr. 21, 2021. [En línea]. Disponible en: https://www.dspace.espol.edu.ec/retrieve/98718/D-106126.pdf

[7] A. Chamlian, G. Telfeyan, y N. Piquerez, "Software Define Networking en redes inalámbricas", Univ. Repúb., 2016, Consultado: may 07, 2021. [En línea]. Disponible en: https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/19026

[8] G. J. Cuba Espinoza y J. M. A. Becerra Ávila, "Diseño e implementación de un controlador SDN/openflow para una red de campus académica", Pontif. Univ. Católica Perú, ago. 2016, Consultado: abr. 21, 2021. [En línea]. Disponible en: http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/7149

[9] P. Göransson y C. Black, Software Defined Networks A Comprehensive Approach. Waltham: Elsevier Inc. 2014.

[10] Y. Jarraya, T. Madi, y M. Debbabi, "A Survey and a Layered Taxonomy of Software-Defined Networking", IEEE Commun. Surv. Tutor., vol. 16, núm. 4, pp. 1955–1980, Fourthquarter 2014, doi: 10.1109/COMST.2014.2320094.

[11] "What's Software-Defined Networking (SDN)?", SDxCentral, ago. 26, 2016. https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/ (consultado may 15, 2021).

[12] M. Josan, "Cómo funciona una conexión de fibra. GPON y FTTH", NASeros, mar. 13, 2017. https://naseros.com/2017/03/13/como-funciona-una-conexion-de-fibra-gpon-y-ftth/ (consultado may 06, 2021).

[13] P. Parol y M. Pawlowski, "Towards networks of the future: SDN paradigm introduction to PON networking for business applications", en 2013 Federated Conference on Computer Science and Information Systems, sep. 2013, pp. 829–836.

[14] Z. Chen, Z. Luo, X. Duan, y L. Zhang, "Terminal handover in software-defined WLANs", EURASIP J. Wirel. Commun. Netw., vol. 2020, núm. 1, p. 68, mar. 2020, doi: 10.1186/s13638-020-01681-w.

[15] J. Salazar Soler, Redes inalámbricas. European Virtual Learning Platform for Electrical and Information Engineering, 2016. Consultado: jun. 19, 2021. [En línea]. Disponible en: https://upcommons.upc.edu/handle/2117/100918

[16] B. Zavala y T. Iván, "Diseño de una red inalámbrica para una empresa de Lima", Pontif. Univ. Católica Perú, oct. 2011, Consultado: jun. 19, 2021. [En línea]. Disponible en: http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/809

[17] A. C. López Baena, L. A. Caballero García, y J. C. Slagle Restrepo, "Diseño de una red inalámbrica para el acceso a internet de la institución educativa departamental José Benito Vives de Andreis de la zona bananera desde la Universidad Cooperativa de Colombia sede Santa Marta.", Alcadía Zona Banan. 2018 Inf. Munic. Obtenido Httpwwwzonabananera-MagdalenagovcoMiMunicipioPaginasInformaciC3B3n-Munic., 2018, Consultado: jun. 29, 2021. [En línea]. Disponible en: https://repository.ucc.edu.co/handle/20.500.12494/8522

[18] A. Jiménez y J. Mauricio, "'Análisis y diseño de una red inalámbrica de larga distancia para proveer acceso a internet a zonas rurales. Caso de estudio sector rural de los cantones Pujilí y Saquisilí de la provincia de Cotopaxi'", 2018, Consultado: jun. 29, 2021. [En línea]. Disponible en: http://repositorio.puce.edu.ec:80/xmlui/handle/22000/15084

[19] V. León y A. Emilio, "Estudio de factibilidad y diseño de una red de Acceso 'Punto – Multipunto' para brindar servicios de Internet y Telefonía en el sector Los Almendros Sur de la ciudad de Guayaquil en Banda no Licenciada de 5 GHz.", mar. 2017, Consultado: jun. 29, 2021. [En línea]. Disponible en: http://repositorio.ucsg.edu.ec/handle/3317/7681

[20] C. I. Quispe Ordoñez, "Diseño e implementación de un balanceador de carga para la optimización de los recursos de protección en una red Enterprise mediante un banco de Firewalls N: 1 controlado vía SDN", Pontif. Univ. Católica Perú, dic. 2019, Consultado: jun. 13, 2021. [En línea]. Disponible en: http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/15579

[21] H. Vaca y A. Javier, "Diseño y optimización de una red GPON a través de una red SDN para la Facultad Técnica para el Desarrollo.", Univ. Católica Santiago Guayaquil, sep. 2020, Consultado: abr. 21, 2021. [En línea]. Disponible en: http://repositorio.ucsg.edu.ec/handle/3317/15577

[22] Y. L. Montoya Álzate y J. M. Avendaño Ocampo, "Emulación del proceso de conmutación / apilamiento de etiquetas en redes MPLS, mediante una herramienta de simulación para redes definidas por software.", Univ. Católica Pereira, abr. 2016, Consultado: jun. 15, 2021. [En línea]. Disponible en: http://repositorio.ucp.edu.co/handle/10785/3674

[23] E. de la Torre y R. Zadiel, "Aplicaciones de SDN/NFV en redes inalámbricas de área local", Thesis, Universidad Central "Marta Abreu" de Las Villas, Facultad de Ingeniería Eléctrica, Departamento de Electrónica y Telecomunicaciones, 2017. Consultado: jun. 27, 2021. [En línea]. Disponible en: http://dspace.uclv.edu.cu:8089/xmlui/handle/123456789/7955

[24] C. S. Hong, S. M. Ahsan Kazmi, S. Moon, y N. Van Mui, "SDN Based Wireless Heterogeneous Network Management", en AETA 2015: Recent Advances in Electrical Engineering and Related Sciences, Cham, 2016, pp. 3–12. doi: 10.1007/978-3-319-27247-4_1.

[25] C. S. Hong, S. M. Ahsan Kazmi, S. Moon, y N. Van Mui, "SDN Based Wireless Heterogeneous Network Management", en AETA 2015: Recent Advances in Electrical Engineering and Related Sciences, Cham, 2016, pp. 3–12. doi: 10.1007/978-3-319-27247-4_1.

[26] G. Pereira y E. Gamess, "Lineamientos para el Despliegue de Redes SDN/OpenFlow", vol. 4, pp. 21–33, dic. 2017.

[27] "gRPC", gRPC, 2020. https://grpc.io/ (consultado jun. 27, 2021).

[28] "Next Generation Network Engineers", THAI CPE, 2018. https://www.thaicpe.com//discussion/14295/next-generation-network-engineers/p1 (consultado jun. 27, 2021).

[29] F. Ramírez Navia, "Cacti, Monitoreo de Red y Reportes Gráficos OpenSource", ITSoftware, sep. 22, 2017. [En línea]. Disponible en: https://itsoftware.com.co/content/cacti-sistema-recoleccion-datos-graficas/ (consultado nov. 28, 2021).

[30] "Cacti®: la solución gráfica completa basada en RRDTool", Cacti®. [En línea]. Disponible en: https://www.cacti.net/ (consultado nov. 28, 2021).

[31] "NAGIOS: Herramienta para gestión-diagnóstico de Red en Linux", Grupo Spi, jul. 12, 2007. [En línea]. Disponible en: https://www.spri.eus/euskadinnova/es/enpresa-digitala/agenda/nagios-herramienta-para-gestion-diagnostico-linux/3909.aspx (consultado nov. 28, 2021).

[32] "PRTG Network Monitor: supervisión de red para profesionales", PAESSLER THE MONITORING EXPERTS, 1997. [En línea]. Disponible en: https://www.paessler.com/es/network_monitoring_tool (consultado nov. 28, 2021).

[33] S. Córdoba López, "Estudio de redes SDN mediante Mininet y MiniEdit", Proyecto/Trabajo fin de carrera/grado, Universidad Politècnica de València, 2019. Consultado: jun. 15, 2021. [En línea]. Disponible en: https://riunet.upv.es/handle/10251/127877

[34] "MININET", Open Networking Foundation. https://opennetworking.org/mininet/ (consultado jun. 09, 2021).

[35] "Mininet". http://mininet.org/news/ (consultado jun. 23, 2021).

[36] A. de J. Merino Gala y C. Quispe Ordoñez, "Manual de usuario del emulador de Redes: VNRT". AINET Solutions S.A.C, jul. 2018.