ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ

DIRECCIÓN DE POSGRADO Y FORMACIÓN CONTINUA

INFORME DE INVESTIGACIÓN

PREVIA A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN MENCIÓN REDES Y SISTEMAS DISTRIBUIDOS

MODALIDAD: TRABAJO DE TITULACIÓN

PROPUESTA DE UNA SOLUCIÓN VIRTUALIZADA UTILIZANDO SOFTWARE LIBRE COMO INFRAESTRUCTURA EN LA RED DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ

AUTORES:

MERCY KARINA MENDOZA ZAMBRANO

JONATHAN ALEXANDER MORAN MACIAS

TUTOR:

DR. C. JAVIER HERNÁN LÓPEZ ZAMBRANO

CALCETA, NOVIEMBRE 2023

DERECHOS DE AUTORÍA

Mercy Karina Mendoza Zambrano y Jonathan Alexander Moran Macias, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, que se han respetado los derechos de autor de terceros, por lo que asumimos la responsabilidad sobre el contenido del mismo, así como ante la reclamación de terceros, conforme a los artículos 4, 5 y 6 de la Ley de Propiedad Intelectual.

A través de la presente declaración cedemos los derechos de propiedad intelectual a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, según lo establecido en el artículo 46 de la Ley de Propiedad Intelectual y su Reglamento.

Firmado electrónicamente por:
JONATHAN ALEXANDER MORAN MACIAS

Mercy Karina Mendoza Zambrano	Jonathan Alexander Moran Macias

CERTIFICACIÓN DEL TUTOR

Dr.C. Javier Hernán López Zambrano, certifica haber tutelado el trabajo de titulación Propuesta de una solución virtualizada utilizando software libre como infraestructura en la red de la Universidad Técnica de Cotopaxi Extensión La Maná, que ha sido desarrollado por Mercy Karina Mendoza Zambrano y Jonathan Alexander Moran Macias, previo la obtención del título de Magíster en Tecnologías de la Información Mención Redes y Sistemas Distribuidos, de acuerdo al Reglamento de unidad de titulación de los programas de Posgrado de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

Firmado digitalmente por JAVIER HERNAN LOPEZ ZAMBRANO

Dr. C. Javier Hernán López Zambrano

APROBACIÓN DEL TRIBUNAL

Los suscritos integrantes del tribunal correspondiente, declaramos que hemos **APROBADO** el trabajo de titulación Propuesta de una solución virtualizada utilizando software libre como infraestructura en la red de la Universidad Técnica de Cotopaxi Extensión La Maná, que ha sido propuesto, desarrollado y sustentado por **Mercy Karina Mendoza Zambrano** y **Jonathan Alexander Moran Macias**, previa la obtención del título de Magíster en Tecnologías de la Información Mención Redes y Sistemas Distribuidos de acuerdo al Reglamento de la unidad de titulación de los programas de Posgrado de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

Ramón Varela

Firmado digitalmente porRAMON AGUSTIN VARELA MUNOZ

M.Sc. Ramón Varela Muñoz
MIEMBRO

GUSTAVO GABRIEL MOLINA GARZON

Firmado digitalmente por GUSTAVO GABRIEL MOLINA GARZON
DN: CN=GUSTAVO GABRIEL MOLINA GARZON, SERIALNUMBER=200623174906, OU=ENTIDAD DE CERTIFICACION DE INFORMACION, O=SECURITY DATA S.A. 2, C=EC
Razón: Estoy aprobando este documento
Ubicación:
Fecha: 2023.10.31 16:19:27-05'00'
Foxit PDF Reader Versión: 2023.2.0

M.Sc. Gustavo Molina Garzón
MIEMBRO

Firmado electrónicamente por:
RAMON JOFFRE MOREIRA PICO

M.Sc. Joffre Moreira Pico
PRESIDENTE

AGRADECIMIENTO

Este informe es el resultado del esfuerzo dedicado día a día para lograr cada una de mis metas propuestas.

Por esto agradezco a mis padres, esposa y hermanos quienes a lo largo de toda mi vida me han apoyado y motivado mi formación académica, creyeron en mí en todo momento y no dudaron de mis habilidades.

A mi tutor el Dr. Javier López, quien con sus conocimientos y orientaciones me ayudó en el desarrollo del proyecto de titulación para culminar satisfactoriamente.

Expresar un eterno agradecimiento a esta prestigiosa institución como es la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, que me permitió ser parte de ella preparándome académicamente dentro de sus aulas.

JONATHAN ALEXANDER MORAN MACIAS

AGRADECIMIENTO

A Dios ante todo por mantenerme con salud y brindarme la fortaleza para concluir este proyecto de titulación.

A mi madre y hermanos quienes me impulsaron y guiaron para que pueda cumplir con mi objetivo y lo pueda hacer realidad, a las personas que me apoyaron e incentivaron a continuar con nuestro trabajo.

A todos y cada uno de los docentes que han contribuido con nuestros conocimientos y actitudes, y así poder culminar con éxito una etapa de mi vida.

MERCY KARINA MENDOZA ZAMBRANO

DEDICATORIA

El presente trabajo quiero dedicárselo a las personas que de una u otra manera estuvieron conmigo en los momentos difíciles, brindándome su amor y sabiduría para poder culminar mi trabajo.

A mis padres Francisco y Lucia que siempre han estado presentes brindándome su apoyo incondicional para realizar todo lo que me proponga.

A mi esposa Ambar que ha sido un pilar fundamental en el proceso de estudio ya que en todo momento estuvo brindándome su apoyo para que continúe hasta finalizar la meta propuesta.

A mis hermanos Luis y Nahomi, por estar siempre presentes y acompañarme en el cumplimiento de este objetivo de vida planteado.

JONATHAN ALEXANDER MORAN MACIAS

DEDICATORIA

Dedico de manera especial a mi madre y hermanos ya que han sabido formarme con buenos valores, lo cual me ha ayudado a seguir adelante en los momentos difíciles.

A mi familia por su comprensión y estímulo constante, además de su apoyo incondicional a lo largo de mis estudios.

Y a todas las personas que de una u otra forma me apoyaron en la realización de este trabajo.

MERCY KARINA MENDOZA ZAMBRANO

CONTENIDO GENERAL

CONTENIDO DE FIGURAS

CONTENIDO DE TABLAS

RESUMEN

Las Redes Definidas por Software constituyen una oportunidad para ofrecer servicios de calidad, basados en el desacoplamiento de los planos de control y datos, que satisfagan las exigencias actuales de las telecomunicaciones. Esta investigación propone satisfacer las limitaciones que se detectaron en el funcionamiento, rendimiento y calidad del servicio en la Universidad Técnica de Cotopaxi Extensión La Maná. Se usó el método cuasi experimental con un enfoque cualitativo y la metodología PPDIOO que contempla las fases necesarias para la obtención de requerimientos. La propuesta de virtualización utilizó software libre, se seleccionaron los simuladores y controladores y se eligió a Mininet como emulador de la red, el controlador Aruba Van SDN, mediante el protocolo OpenFlow, fueron utilizadas como herramientas tecnológicas de virtualización y comunicación VMware Workstation y Mikrotik. Como resultado se logra la gestión, el monitoreo de datos en la red diseñada y se comprueba su correcto funcionamiento, ofreciendo múltiples opciones para la gestión directa y centralizada de la red a un alto nivel y el desarrollo de aplicaciones o APIs para una gestión personalizada, que ofrezca información o permita cambios en tiempo real del comportamiento de la red.

PALABRAS CLAVE

Redes Definidas por Software, OpenFlow, Mininet, virtualización, controladores.

ABSTRACT

Software Defined Networks are an opportunity to offer quality services, based on the decoupling of control and data planes, which meet the current demands of telecommunications. This research proposes to satisfy the limitations that were detected in the operation, performance and quality of the service at the Technical University in Cotopaxi Extension La Maná. The quasi-experimental method was used with a qualitative approach and the PPDIOO methodology that contemplates the necessary phases to obtain requirements. The virtualization proposal used free software, simulators and controllers were selected and Mininet was chosen as the network emulator, the Aruba Van SDN controller, through the OpenFlow protocol, VMware Workstation and Mikrotik were used as virtualization and communication technology tools. As a result, management is achieved, data monitoring in the designed network and its proper functioning is verified, offering multiple options for direct and centralized management of the network at a high level and the development of applications or APIs for personalized management that offers information or allows changes in real time of the behavior of the network.

KEYWORDS

Software Defined Networks, OpenFlow, Mininet, virtualization, controllers.

CAPÍTULO I. ANTECEDENTES

1.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

Desde el surgimiento de las tecnologías ha existido la necesidad del intercambio de datos, lo cual se logró con la llegada de la computación y las comunicaciones, las cuales implementaron varias técnicas y mecanismos para conseguir su funcionamiento, inicialmente utilizando sistemas primarios hasta llegar a otros más complejos y de mayor alcance, por lo cual en 1969 surgen las redes de computadoras con la creación de ARPANET (Oviedo Bayas et al., 2021).

Con base en las redes de comunicaciones, actualmente se facilita un enlace excepcional, innovando a nivel de arquitecturas tanto como de tecnologías, es decir, que las redes están sujetas a un sin número de exigencias para las cuales no fueron inicialmente diseñadas e implementadas, por lo que constituye un reto migrar a un medio de comunicación para que proporcione mayor rendimiento en la misma (Quijada, 2021).

Las redes de computadoras han venido evolucionando y trabajando de manera eficaz, sin embargo, algunas redes corporativas, de instituciones y organismos diversos, cuentan con varios años de haber sido diseñadas e implementadas y, en ocasiones, ya no satisfacen los requerimientos de los servicios que necesita la organización en la actualidad, como la de implementar servicios de red personalizados, realizar cambios en los patrones de tráfico en la red de manera dinámica, flexible y escalable permitiendo un acceso rápido, ágil y eficaz a los servicios. Por ello se requiere un diseño más flexible que permita realizar cambios de manera sencilla, de manera dinámica, con interfaces programables, gestionadas por software y con funciones de virtualización de red (Núñez, 2015).

La Universidad Técnica de Cotopaxi Extensión La Maná, al igual que otras instituciones, enfrenta problemáticas en su arquitectura tradicional de red,

evidenciándose limitaciones en la calidad del servicio y la administración. Al aumentar el número de usuarios y, por ende, las actividades, se ha generado una saturación del tráfico en la red, retardos e interrupción en los procesos académicos y administrativos que se realizan en la institución, con la consiguiente afectación al correcto desarrollo de sus actividades académicas y de gestión.

Por otro lado, la arquitectura actual de la red institucional se ve afectada por las limitaciones en la adquisición de equipamiento; dificultades para administrar y dar un mantenimiento correctivo o preventivo, debido a la variedad del equipamiento y a las dificultades para proporcionar mantenimiento a los servicios; mejorar la seguridad; realizar respaldos de manera fácil y rápida; recuperar fallos y dar soportes técnicos más efectivos para el área de telemática de la universidad, lo cual hace visible la necesidad de desarrollar e implementar una solución que no solo corrija los aspectos negativos de la red tradicional, sino que permita la administración, mantenimiento y actualización de la red de manera sencilla.

Teniendo en cuenta la necesidad de mejorar la calidad, el funcionamiento y el servicio que ofrece la red a la comunidad universitaria, así como los antecedentes a nivel mundial que proponen soluciones a situaciones similares, se define en esta investigación el siguiente problema científico: ¿Cómo el despliegue de una Red Definida por Software puede mejorar el funcionamiento, rendimiento y calidad del servicio en la Universidad Técnica de Cotopaxi Extensión La Maná?

1.2. JUSTIFICACIÓN

A nivel mundial se han desarrollado investigaciones que proponen diversas soluciones, entre ellas la implementación de Redes Definidas por Software (Software Defined Networking, SDN) que facilitan la administración centralizada de la red a través de una plataforma de software (Dixon, 2016). Empresas de tecnología a nivel mundial se encuentran desarrollando proyectos con este modelo, como la interconexión entre los data center de Google. Por su parte, Facebook, Amazon Web Services y Microsoft son contribuyentes activos en los

avances en esta área tanto en el ámbito epistemológico como práctico (Roncero, 2014). Estos trabajos sirven como antecedentes a la presente investigación.

Los resultados obtenidos en la implementación de redes SDN, demuestran las bondades, los beneficios, los avances y el desarrollo en este campo, lo que justifica su utilización por empresas importantes del mundo de las telecomunicaciones como: Cisco, Juniper, HP, entre otras (Roncero, 2014).

En Latinoamérica se han desarrollado casos de estudio para analizar la aplicabilidad de este tipo de arquitectura, que cada vez es más usada en el área de las telecomunicaciones. Ramírez Giraldo y López Echever (2018), analizan el nuevo paradigma de las redes, su arquitectura, componentes y funcionamiento y su impacto en el ámbito académico. Duarte y Lobo (2015), por su parte, presentan un análisis relacionado con la importancia de migrar hacia una red SDN por las ventajas que reporta a la satisfacción de las demandas de las universidades, sus docentes, estudiantes e investigadores.

Investigaciones desarrolladas en Ecuador coinciden con lo planteado por los autores antes mencionados, al comprobar la mejora en la calidad de los servicios que las redes SDN han tenido en diversos sectores (Pérez y Marín, 2021). Es por ello que son varias las universidades que se encuentran en investigaciones y desarrollo de prototipos para migrar su infraestructura de red tradicional a SDN o mantener un entorno híbrido. Bernal y Mejía (2016), de la Escuela Politécnica Nacional, plantean la importancia de la selección del controlador, elemento principal de una red SDN, y de la integración de un simulador para obtener resultados que aporten al despliegue de estas soluciones basadas en software.

Para la Universidad Técnica de Cotopaxi Extensión La Maná la realización de este proyecto no solo representa un beneficio para los administradores e investigadores que lo ejecutan, al permitirles desarrollar sus habilidades prácticas y el aprendizaje de conceptos y modelos innovadores en el campo de las redes,

sino que constituye un gran impacto social al beneficiar a todos los usuarios de la comunidad universitaria del cantón La Maná.

La viabilidad del proyecto se corroboró con la aceptación por parte de las autoridades universitarias, para diseñar una propuesta de solución virtualizada, utilizando software libre, como infraestructura en la red. Las soluciones de virtualización son alternativas, además, para la reducción de los costos operativos, para una mejor utilización de recursos y requisitos de gestión más fáciles, contribuyendo a la seguridad, a brindar soportes técnicos más efectivos para el área de telemática de la universidad, facilitando que los usuarios tengan estaciones de trabajo estandarizadas, acceso a las aplicaciones que se autoricen y se minimiza la presencia de virus.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Diseñar una propuesta de una solución virtualizada utilizando software libre como Infraestructura en la Red de la Universidad Técnica De Cotopaxi Extensión La Maná para mejorar la administración y calidad de servicio de la red.

1.3.2. OBJETIVOS ESPECÍFICOS

1. Indagar sobre los diferentes tipos de redes gestionadas por software (SDN), sus características y ventajas.
2. Simular en un ambiente controlado los diferentes tipos de redes SDN.
3. Analizar los resultados obtenidos en la simulación.
4. Elaborar una propuesta idónea basada en redes SDN, que se adapte a la realidad y necesidad de la UTC extensión La Maná en cuanto a la gestión de los servicios de red.

1.4. IDEAS A DEFENDER

La propuesta de una solución virtualizada, utilizando software libre como infraestructura en la red de la Universidad Técnica de Cotopaxi Extensión La Maná contribuirá a mejorar la administración y calidad de servicio prestado.

CAPÍTULO II. REVISIÓN BIBLIOGRÁFICA

2.1. REDES DEFINIDAS POR SOFTWARE

En la actualidad los servicios de telecomunicaciones y las arquitecturas de red presentan un nivel alto de complejidad, por lo que no cumplen con las expectativas del usuario para mejorar los servicios, por lo que la industria de las Tecnologías de la Información y la Comunicación (TIC) se han visto en la necesidad de reevaluar los conceptos de las redes tradicionales (Oviedo Bayas et al., 2021).

En este contexto, las redes SDN han pasado a ser una opción a considerar por muchas organizaciones reconocidas a nivel global (Marín Muro y Alvarez Paliza, 2016). La Open Networking Foundation (ONF, por sus siglas en inglés) “es un consorcio sin ánimo de lucro dedicado al desarrollo, estandarización y comercialización de las SDN. Esta organización es la que ha presentado la visión más aceptada y está conformada por todas las grandes instituciones y empresas de las TIC” (Pérez Tardío et al., 2019).

La O. N. Fundation (2012), como se citó en Jiménez Morales (2018, p. 5) plantea que “las redes SDN se definen como una arquitectura de red dinámica, gestionable, adaptable, de costo eficiente, que la hace ideal para las altas demandas de ancho de banda y la naturaleza dinámica de las aplicaciones actuales. Esta arquitectura desacopla el control de la red y la funcionalidad de reenvío de información, permitiendo que el control de la red pueda ser completamente programable, logrando que las aplicaciones y servicios de red se abstraigan de la infraestructura de red subyacente”.

Es necesario, además, considerar que la seguridad es esencial en todos los tipos de redes existentes, incluyendo las SDN, teniendo en cuenta la necesidad de garantizar a los clientes servicios que satisfagan sus exigencias sobre la

información que se transmite. Por otra parte, son efectivas para asegurar su despliegue en cualquier momento y sitio (Barrera Pérez et al., 2018).

2.2. EVOLUCIÓN DE LAS REDES PROGRAMABLES

Al analizar el surgimiento y evolución de las redes SDN se destacan diversos factores como los planteados por Spera (2013), como se citó en Ríos (2016, p. 29), “es importante reconocer que la masividad de los dispositivos móviles, el nuevo contenido audiovisual, la virtualización de servidores y la integración de los servicios en la nube, son algunas de las tendencias que están impulsando a la industria a repensar las arquitecturas tradicionales de red, que fueron diseñadas basadas en una arquitectura jerárquica, que tiene sentido en un ambiente cliente-servidor”.

Este mismo autor considera, además, que la arquitectura tradicional no responde al almacenamiento y a las soluciones dinámicas que demandan en la actualidad los diferentes nodos de cómputo, debido a los cambios en el tráfico, en el acceso a la información y a las diferentes aplicaciones. En correspondencia el acceso y, por tanto, el tráfico generado por los usuarios también se ha modificado.

Teniendo en cuenta estas demandas y la necesidad de ofrecer soluciones óptimas, surgen las llamadas redes inteligentes que basan su diseño en la implementación de redes SDN y la Virtualización de las Funciones de Red (Network Function Virtualization, NFV por sus siglas en inglés) (Pérez Tardío et al., 2019). Pueden cambiar las reglas de los conmutadores de la red, agregando o eliminando prioridades, bloqueando o permitiendo el tráfico de paquetes específicos con un control detallado (Badotra y Panda, 2020).

Las redes SDN, modifican, según PromonLogicalis (2013) como se citó en Ríos (2016, p. 30), “el paradigma de la red tradicional de datos que ha estado en gran medida centrada en hardware, requiriendo configuración dispositivo por dispositivo; y, de la arquitectura tradicional de los dispositivos de red, mostrado en la Figura 1(a), en el cual “... los controles están en el nivel de los elementos de red,

y, por tanto, basados en sistemas propietarios desarrollados por los fabricantes de equipos"; y, la propuesta de uno nuevo, representado en la Figura 1(b), en el que se "prevé una separación entre los planos de control (la inteligencia de un elemento de red, por ejemplo, el software responsable de definir los procesos de enrutamiento, políticas de seguridad, ingeniería de tráfico) y el plano de datos (responsable por realizar el encaminamiento de los paquetes, esto es, la base del reenvío de información); es decir, el principal cambio está determinado por la alteración de la capa de red en la cual se controla el tráfico, que pasa a estar distribuida entre el hardware y la capa de software. Así, los elementos de red pasan a ser responsables tan solo del encaminamiento físico de los paquetes, en tanto que todo el control del enrutamiento es ejecutado por medio de software, en una capa superior".

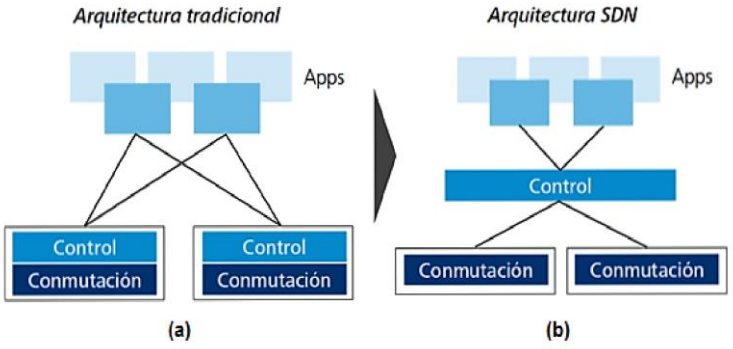


Figura 1. Desacoplo de la capa de control y datos. a) Arquitectura tradicional; b) Arquitectura SDN
Fuente: (Deloitte.com, 2019)

De esta manera se pueden redefinir las políticas de tráfico en la capa control y no es necesario reconfigurar los equipos de hardware (switch y enrutadores) (Anexo 1). Se coincide con Ríos (2016), al considerar que esto "hace factible la interacción de los aplicativos con los elementos de red, permitiendo que el comportamiento de la infraestructura se defina automáticamente con base en la aplicación".

La ONF considera a las redes SDN "como una arquitectura emergente, dinámica, manejable, rentable y adaptable, ideal para la naturaleza de banda ancha de las aplicaciones actuales" (Herrera Vaca, 2020). Esta arquitectura

desvincula las funciones de control y reenvío de la red, permitiendo hacer programable directamente el control de la misma y quedando abstraída la infraestructura subyacente para las aplicaciones y los servicios que ofrece (Citrix.es, 2019).

Se coincide con Gonzalez et al. (2018), Ríos (2016) y Salazar Chacón (2021), en el análisis que realizan sobre las redes programables, considerando, además, las ventajas que, a partir de esa arquitectura se obtienen, ya que “los elementos de red (routers, switches, firewall, entre otros) pasan a tener Interfaces de Programación de Aplicaciones (APIs) en su sistema operativo que crearán la posibilidad de que las aplicaciones no desarrolladas por los fabricantes de hardware interactúen con el plano de control del sistema, tomando decisiones de ingeniería de tráfico basados en patrones inusuales, como temperatura, costo del enlace, consumo de energía, entre otros” (Ríos, 2016, p. 30). Es por ello que, en el contexto de SDN, la virtualización de red se refiere a la creación de redes virtuales, lógicas que son desvinculadas del hardware de red subyacente y que pueden ser controladas mediante programación” (Ríos, 2016).

Según Ahuja et al. (2021), SDN ayuda a los ingenieros de redes a monitorear la red rápidamente, controlarla en un punto central, identificar el tráfico malicioso y fallas en los enlaces de manera fácil y eficiente. Por su parte, Javanmardi et al. (2021), deducen que las redes SDN poseen una gran capacidad para administrar los flujos de los mismos, en baja latencia a las aplicaciones de los usuarios para Internet de las cosas (IoT). Mientras los autores Karakus y Guler (2020), hacen énfasis en la ayuda a los administradores de red al proporcionar rutas de calidad de servicio (QoS) garantizadas de extremo a extremo (E2E) para los flujos entre redes, a la vez que proporcionan una gestión de flujo más detallada junto con una vista global de la red. A su vez, Al Mahdawi y Salih (2021), argumentan la separación de los planos de control y de datos de los dispositivos de red al concentrar el primero en dispositivos centralizados de alto nivel y supervisores eficientes, llamados controladores, como una gran ventaja al momento de administrar la red.

Para Garrich et al. (2020), las redes metropolitanas han experimentado un gran avance tecnológico al aprovechar las capacidades de las SDN y la NFV, pero lejos de limitarse a estas redes, este avance se puede replicar en las redes LAN y WAN para hacer de su administración un proceso más simple, además de asegurar la calidad del servicio de las mismas.

Es por ello que en esta investigación se asume la propuesta de diseño de una red SDN ya que “supone una mejora considerable de la capacidad de gestión y la flexibilidad de la red, permite la gestión automatizada del tráfico, mejora el ancho de banda y posibilita adaptar la red a las necesidades del cliente” (Mahabir, 2023). Para las organizaciones del sector de las telecomunicaciones y las instituciones que usan estos servicios brinda una vista central de la red; contribuye a la reducción de costes, al no tener que invertir en dispositivos de precios elevados; “se trabaja con normas abiertas; brinda mayor velocidad y agilidad; reducción del tiempo de inactividad; mejora la seguridad; mayor facilidad para utilizar los recursos de la nube y la capacidad para probar varias configuraciones de red sin realizar cambios en la misma” (Mahabir, 2023).

Por su parte la NFV pretende resolver el problema del espacio y la corta vida útil de las tecnologías debido a la innovación; conecta diferentes equipos de red a servidores, conmutadores y almacenamiento de alto volumen estándar de la industria. Su ventaja consiste en implementar funciones de red en software que se ejecutan en servidores estándares de la industria, y que pueden trasladarse o actualizarse en diferentes ubicaciones de la red, sin necesidad de instalar nuevos equipos” (Mahabir, 2023).

2.2.1. ARQUITECTURA SDN

Se coincide con Liang y Qiu (2017), cuando enfatizan que la funcionalidad de las redes SDN introduce el concepto de desacoplar el control desde el plano de datos y realizarlo a través de un controlador centralizado. En este sentido, el modelo SDN de referencia propuesto por el Grupo de Trabajo de Arquitectura de

Open Networking Foundation (2016), está compuesto por tres capas: Capa de Aplicación, Capa de Plano de Control y Capa de Plano de Datos (Figura 2).

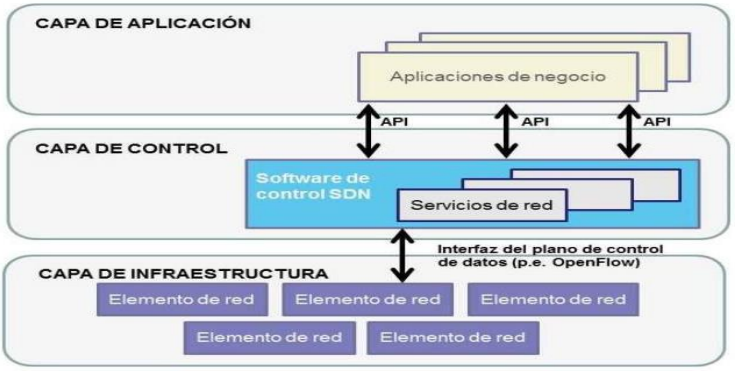


Figura 2. Arquitectura SDN
Fuente: (Millán, 2014)

Capa de Aplicación: Está conformada por las aplicaciones del negocio y por las de servicios de red. Por lo tanto, las aplicaciones del SDN identifican las necesidades, políticas, requisitos y sugerencias de la red al controlador SDN a través de interfaces de aplicación (protocolos de control de red SDN). También, construye una visión abstracta de la red para sus propios fines de programación interna (Open Networking Foundation, 2016).

Capa de Control: Esta capa “contiene los controladores SDN encargados de gobernar y dirigir la manera en que se transportan los datos en los dispositivos SDN. Se encarga de todas las funciones complejas de enrutamiento, manejo de políticas, monitoreo y chequeos de seguridad de la red” (Pereira y Gamess, 2017, p. 42). De este modo, un controlador SDN se lo reconoce como una entidad de software que se encarga del control exclusivo sobre un conjunto abstracto de datos, también puede ofrecer una instancia de modelo de información abstraída para, al menos, un cliente (ONF, 2016).

Huang et al. (2018) se citó en Ruipérez Cuesta (2021, p. 6), consideran que “el plano de control de una configuración SDN consiste en uno o más controladores que usan APIs abiertas para ejercer control sobre los switches de la red. Además de impulsar las reglas de reenvío a los switches, los controladores también vigilan el medio, de esta manera tienen la capacidad de tomar decisiones

de reenvío integradas con la gestión del tráfico en tiempo real". Usan tres interfaces para comunicarse con los diferentes componentes de la infraestructura SND: Southbound, sur; Northbound, norte; y Westbound/Eastbound, este y oeste.

- La interfaz Southbound permite que el controlador se comunique, interactúe y maneje los elementos de envío. Aunque existen otras soluciones patentadas como OnePK (Cisco) y Contrail (Jupiter Networks), OpenFlow es la implementación más común (Open Networking Foundation, 2017).
- La interfaz Northbound posibilita a las aplicaciones que se encuentran en la capa de aplicación programar los controladores, haciendo modelos de datos abstractos y otras funcionalidades disponibles para ellos (Pfaff y Davie, 2018).
- Las interfaces Westbound/Eastbound permiten la comunicación entre grupos de controladores.

El controlador SDN cuenta con las siguientes opciones de APIs Soutbound abiertas, además de OpenFlow, entre ellas se encuentran Open vSwitch Database (OVSDB) (Pfaff y Davie, 2018) y Forwarding and Control Element Separation (ForCES) (Doria et al., 2019). También existen las opciones de plugins como Border Gateway Protocol (BGP), Simple Network Management Protocol (SNMP) y Network Configuration Protocol (NETCONF) (Enns et al., 2018), entre otros. Sin embargo, como se expuso anteriormente y coincidiendo con Pereira y Gamess (2017, p. 61), "OpenFlow es la API Southbound estándar de la industria de las redes para entornos SDN. A través de ella un controlador puede crear, actualizar, modificar y eliminar entradas en las tablas de flujos de los dispositivos SDN y obtener información de estadísticas de estos".

Capa del Plano de Datos: Según Ruipérez Cuesta (2021, p. 5), "esta es la primera capa en la arquitectura SDN y se utiliza para el reenvío de un conjunto de paquetes basado en dispositivos de red que componen la infraestructura de la misma, cuyas funciones principales sirven para hacer cumplir las acciones de reenvío de paquetes de flujo de acuerdo con las instrucciones correspondientes proporcionadas por el controlador y para informar el estado de la red cuando lo

soliciten las aplicaciones de la misma". Los dispositivos de red más comunes que suelen estar presentes en esta capa son switches y routers.

En una red SDN los controladores se comunican con las aplicaciones externas mediante APIs Northbound abiertas, incluyendo el protocolo OpenFlow. Una red SDN también puede ser vista como una composición de capas y sistemas (Figura 2). Algunas capas se encuentran presentes en todos los despliegues SDN incluyendo: controladores, infraestructura de red y APIs Southbound, mientras que otras son opcionales, como las APIs Northbound, aplicaciones, hipervisores y virtualización de redes" (Pereira y Gamess, 2017, p. 22).

2.2.2. PROTOCOLO OPENFLOW

El protocolo OpenFlow fue creado en la Universidad de Stanford, California y posteriormente estandarizado por ONF (Mocha Guacho y Celleri Pacheco, 2020; Ramos Suavita, 2021). Para Pérez Tardío et al. (2019, p. 9), "es considerado el estándar de código abierto más utilizado en las aplicaciones de la interfaz hacia el sur, Southbound, el cual permite la comunicación entre el plano de control y de datos de la arquitectura SDN. Muchas de las grandes corporaciones dentro del mundo de las telecomunicaciones han apoyado a OpenFlow, como es el caso de IBM, Google y HP".

La arquitectura "comprende tres componentes principales, como se muestra en la figura 3: a) los interruptores compatibles con OpenFlow constituyen el plano de datos; b) el plano de control tiene uno o más controladores OpenFlow; c) el plano de control está conectado con los interruptores a través de un canal de control seguro, es decir la interfaz OpenFlow" (Ruipérez Cuesta, 2021, p. 7). Esto se debe a que "OpenFlow desacopla el plano de control del de datos y es el protocolo que más se usa habitualmente para la interfaz Southbound" (Ruipérez Cuesta, 2021).

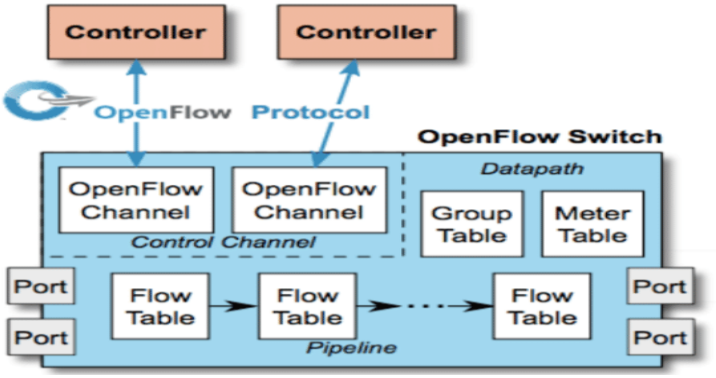


Figura 3. Arquitectura SDN. Protocolo OpenFlow.
Fuente: (Ruipérez Cuesta, 2021)

Como se aprecia en la anterior figura, la arquitectura SDN se divide en, al menos, tres partes (Braun y Menth, 2018). La tabla de flujos, compuesta por registros que contienen “campos en los que debe buscar coincidencias con los paquetes entrantes, contadores e instrucciones sobre qué hacer con los paquetes que coinciden” (Ruipérez Cuesta, 2021). El canal seguro, que “conecta el dispositivo al controlador, permitiendo el envío de comandos y paquetes entre ambos mediante el protocolo OpenFlow y el protocolo OpenFlow proporciona una forma abierta y estandarizada en la comunicación entre el conmutador y el controlador, permitiendo al controlador añadir, eliminar, modificar y buscar en las entradas de la tabla de flujos a través del canal seguro”. (Ruipérez Cuesta, 2021, p. 7)

En la presente investigación se coincide con Pérez Tardío et al. (2019), al considerar que “OpenFlow no sustituye a ninguno de los protocolos de la pila TCP/IP pues cada mensaje va encapsulado en un segmento TCP y sus correspondientes paquetes IP y trama Ethernet, …así se puede lograr un despliegue mucho mayor de aplicaciones de red y alcanzar mayor como calidad de servicio, seguridad de la red o ingeniería de tráfico” (p. 10).

Open vSwitch (OVS) “es un conmutador multicapa, de código abierto. Ha sido desarrollado para la implementación en entornos de producción para la automatización de la red de forma programable” (Pérez Tardío et al., 2019).

Además, fue diseñado para soportar distribuciones a través de múltiples servidores físicos y diversas tecnologías de la virtualización basada en Linux, incluyendo Xen/XenServer (Kurth, 2013), Kernel Virtual Machine (KVM por sus siglas en inglés) (KVM.org, 2019), Proxmox (Kurth, 2015) y VirtualBox (Kreutz et al., 2015). Se coincide con Sayans Cobo (2018), al considerar que por sus funcionalidades lo convierten en una excelente solución para implementaciones SDN.

2.2.3. CONTROLADORES

Según Pérez Tardío et al. (2019, p. 15) “el controlador es el elemento central o principal de la arquitectura SDN, pues se encarga de manejar todos los protocolos, políticas de la red e instrucciones que se envían a los dispositivos de red. El controlador centraliza la inteligencia de la red y gestiona las tablas de flujo de los conmutadores al agregar y eliminar las entradas de flujo sobre el canal seguro, utilizando el protocolo OpenFlow. En una red tradicional, se configurarán los dispositivos conmutadores y enrutadores con la información necesaria para encaminar el tráfico de la red. Para alcanzar una mayor flexibilidad en las redes SDN, resulta más factible que sea el controlador el encargado de reconfigurar todos los dispositivos de reenvío. Además, será responsable de determinar cómo manejar paquetes sin entradas de flujo válidas”.

Tener múltiples controladores mejora la confiabilidad pues el conmutador puede continuar operando en modo OpenFlow si falla la conexión del controlador principal (Dueñas Santos et al., 2017; Jammal et al., 2017). Los controladores coordinan la administración del conmutador entre ellos mismos, y el objetivo de la funcionalidad de múltiples controladores es solo ayudar a sincronizar el traspaso entre ellos (Singh y Jha, 2017).

Existen varios tipos de controladores SDN como NOX, desarrollado por N.O.X. Repo. (2019a); POX por N.O.X. Repo. (2019b); Beacon de Erickson (2013); ONOS por Open Network Operating System (2019); Ryu diseñado por Osrg/Ryu.org (2019) y OpenDaylight (ODL) de OpenDaylight.org (2019). Para

Pérez Tardío et al. (2019, p. 6), “estos permiten la gestión de las redes definidas por software, pero poseen diferencias que determinan su utilización, como es el caso de la versión de OpenFlow soportada, el lenguaje de desarrollo, la interfaz gráfica, el soporte de plataformas, entre otras. Existe la necesidad de un análisis comparativo extenso pues no se conoce a priori una diferencia marcada entre uno y otro”. (Anexo 2).

2.2.3.1. OPENDAYLIGHT (ODL)

OpenDaylight Project es una plataforma utilizada en redes SDN, ideal para obtener diversas opciones de configuración, además existen nuevas aplicaciones construidas sobre su plataforma lo que hace la transición a SDN aún más fácil. El proyecto ODL cuenta con el apoyo de diferentes compañías como son Brocade, Red Hat, NEC, Microsoft, Hp, Dell, IBM, entre otras.

Ruipérez Cuesta (2021), reporta que “es un proyecto impulsado por la Linux Foundation y que, según su página oficial, es el controlador SDN de código abierto más implementado en la actualidad. Cuenta con una arquitectura de tres capas principales, las mismas que se encuentran separadas por dos interfaces que son las que permiten la comunicación entre ellas, además el controlador posee una interfaz gráfica avanzada que brinda varias ventajas, a diferencia de los otros controladores como POX que utiliza líneas de comando y Floodlight que posee menos funcionalidades (Figura 4) (OpenDaylight.org, 2019).

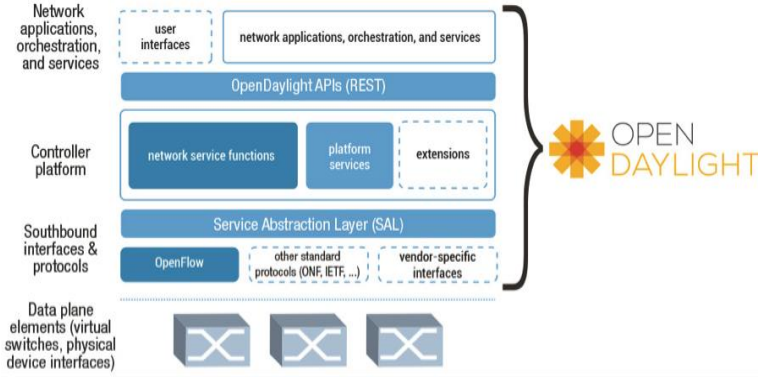


Figura 4. Arquitectura controlador OpenDaylight.
Fuente: (Hewllet Packard Enterprise, 2022)

OpenDaylight permite a) centralizar el control de los dispositivos virtuales y físicos que posee la red; b) controlar los dispositivos con protocolos abiertos y estándares y c) proveer abstracción de alto nivel para que las personas encargadas de redes y desarrolladores creen aplicaciones que permitan personalizar la administración y configuración en las redes (Barona et al., 2014)

La plataforma SDN OpenDaylight soporta una extensa colección de protocolos para redes clásicas, como OpenFlow, patrones de tipo tabla, extensión de OpenFlow, y otros como BGP/PCEP, CAPWAP y NETCONF. Junto con esto también consolida Open Vswitch y Openstack a través del plan de integración OVSDB (Prakruthi y Patil, 2016).

2.2.3.2. SDN VAN Controller

El controlador SDN VAN Controller, desarrollado por el fabricante de dispositivos de red HP ofrece un espacio de control unificado en la red, la misma que se encuentra habilitada para protocolos Openflow, lo que permite una administración más simplificada, así como el aprovisionamiento, la organización y una nueva generación de servicios de la red basados en aplicaciones (Contreras Pardo, 2014).

Entre las características del controlador se encuentran que a) posee interfaces abiertas y programables; b) proporciona un control flexible y centralizado; c) ofrece una alta disponibilidad y mayor escalabilidad y d) posee robustez en la seguridad (Figura 5).

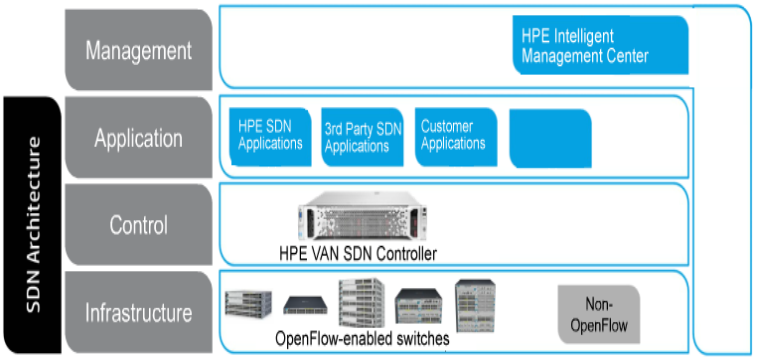


Figura 5. Arquitectura controlador SDN VAN Controller.
.**Fuente:** (Hewllet Packard Enterprise, 2022)

Además, es de código abierto y posee un módulo de topología de servicio optimizado, el cual permite buscar hosts, switch Openflow o algún usuario en la GUI.

2.2.3.3. OPEN NETWORK OPERATING SYSTEM (ONOS)

Open Network Operating System más conocido por sus siglas ONOS, se define como un sistema operativo que se utiliza para gestionar equipos de red mediante un entorno SDN y fue desarrollado por la empresa Open Networking Lab en el año 2014. El principal objetivo de dicho controlador es facilitar el despliegue y desarrollo de soluciones NFV y SDN. Además, es un sistema open source que es apoyado por empresas privadas como AT & T, Intel, Huawei entre otras.

El sistema operativo ONOS adopta una arquitectura distribuida para disponibilidad y escalabilidad horizontal. Proporciona una vista de red global a las aplicaciones, que está lógicamente centralizada, aunque se distribuye físicamente en varios servidores (Berde et al., 2014). Este sistema tiene definidas dos especificaciones de prototipos, uno se enfoca en la construcción de una arquitectura de red que provea una vista global de la red con tolerancia a fallos y características de escalabilidad y el otro busca mejorar el rendimiento general del sistema (Paliwal et al., 2018).

La arquitectura de ONOS se divide en tres capas que son: Interfaz Southbound, core e interfaz Northbound (Figura 6). Posee las siguientes características: a) constituye una plataforma que integra un conjunto de aplicaciones que actúan como un controlador SDN distribuido, extensible y modular; b) permite la gestión, configuración e implementación, permitiendo simplificar la integración de nuevos software, hardware y servicios y c) posee un entorno sumamente versátil e interoperable, el cual es compatible con diferentes marcas o configuraciones de equipos de red.

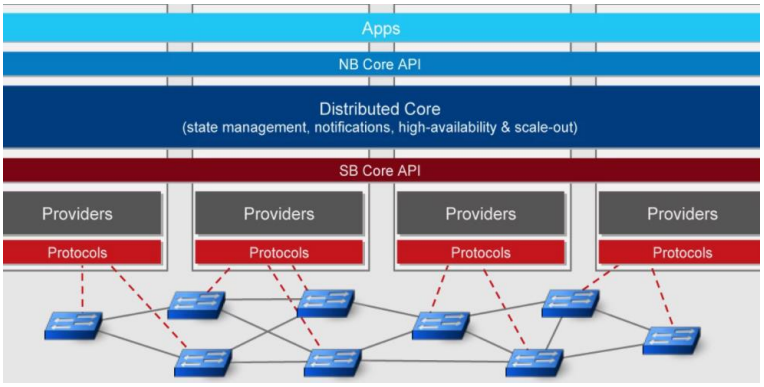


Figura 6. Arquitectura controlador ONOS
Fuente: (Open Networking Foundation, 2016)

2.3. MININET

En la actualidad la simulación de procesos de optimización de redes, de automatización, de prueba de nuevos protocolos, entre otros, posibilita depurar errores en entornos seguros, sin riesgos que impliquen altos costos (Córdoba López, 2018). Con este fin surge Mininet, como "una plataforma que permite crear redes virtuales a gran escala de forma rápida y eficiente" (Pérez Tardío et al., 2019). Se diseña en la Universidad de Stanford, en principio para apoyar los procesos de investigación y de enseñanza-aprendizaje de las disciplinas relacionadas con las redes de computadoras.

Permite crear topologías complejas que respondan a las necesidades de los servicios de la red, usando scripts desarrollados en Python. Una ventaja a tener en cuenta la constituye el ser un proyecto de código abierto, lo que permite mantenerlo en desarrollo continuo (Córdoba López, 2018). Además, en un ordenador se pueden crear redes complejas y comprobar la escalabilidad de los sistemas (Jha y Tech, 2017).

2.4. HERRAMIENTAS TECNOLÓGICAS DE VIRTUALIZACIÓN Y COMUNICACIÓN

Se coincide con Bernal y Mejía (2016), cuando plantean que "las SDN y NFV son complementarias, pero no depende una de la otra. Por ejemplo, los servicios de red pueden ser desarrollados en software y ejecutados o emulados en recursos lógicos sin la necesidad de desplegar una SDN, y viceversa. Sin

embargo, su combinación puede crear un entorno de recursos virtuales, interconectados por enlaces virtuales que se configuran o destruyen fácilmente para servir a múltiples aplicaciones" (Bernal y Mejía, 2016, p. 7). Ambas permiten garantizar redes más flexibles, potentes, con mayor capacidad de programación, a la vez que facilitan ampliar y ofrecer nuevos servicios con menor costo.

Las SDN y NFV hacen que las infraestructuras de telecomunicaciones sean más omnipresentes (pervasive), flexibles y capaces de soportar todos los terminales que se conecten a las redes (Pretz, 2016; Sayans Cobo, 2018; Flores, 2022).

2.4.1. VIRTUAL BOX

Virtualbox es un software utilizado para el desarrollo de contenidos virtuales en una máquina host física, también guarda una estrecha relación con las distribuciones de Windows (Flores, 2022, p. 174).

2.4.2. VMWARE

VMmare es un sistema que posee una versión gratuita y no requiere de instalaciones adicionales de un software externo, permite la interconexión entre el equipo virtual y el host, cuya función es manejar el sistema operativo de Windows y Linux. El Vmware tiene como funcionalidad procesar la información de forma inmediata en el sistema operativo de las máquinas virtuales, y con ello genera la adaptabilidad de la TI (García, 2022).

2.4.3. OPEN NEBULA

La arquitectura física que asume OpenNebula es un clúster clásico porque las máquinas virtuales creadas residen en el mismo host y existe, al menos, una red física que conecta todos estos servidores. Actúa como un entorno global de gobierno de nube pública y privada, lo que permite la interoperabilidad entre entorno de virtualización como VMware, KVM y contenedores LXD a través del panel de control (Pereira y Gamess, 2017).

2.4.4. OPENSTACK

OpenStack es una solución IaaS para recursos de hardware y un mecanismo para administrar e implementar máquinas virtuales específicas, al tiempo que permite el crecimiento horizontal al agregar nodos de cómputo a la solución cuando sea necesario, basado en la nube que puede usar para configurar y ejecutar su infraestructura informática y de almacenamiento (Kumar y Deepa, 2017).

2.4.5. HYPERV

Hyper-V proporciona una plataforma de virtualización simple y confiable que mejora la utilización del servidor, al mismo tiempo que reduce los costos, lo que permite a los clientes empresariales crear nubes privadas e implementar modelos operativos de servicios. Su función consiste en ejecutar varias máquinas virtuales bajo pruebas con un mismo hardware, con la finalidad de evitar problemas en el acceso del sistema base (IONOS, 2020).

2.4.6. ROUTER MIKROTIK

Mikrotik RouterOS es el sistema operativo que utiliza el hardware Mikrotik RouterBOARD, el cual está basado en el kernel de Linux v3.3.5. Cuenta con una interfaz sencilla de operar, proporcionando a todas las funciones una instalación rápida y sencilla. Puede ejecutarse desde discos IDE o módulos de memoria flash, cuenta con un diseño modular, con módulos actualizables y una interfaz gráfica amigable (Jiménez Contreras, 2018).

2.4.7. ROUTER CISCO

El enrutador Cisco proporciona conectividad de la capa de red en la interconexión del sistema abierto, a diferencia de un conmutador, que proporciona conectividad y reduce el tiempo en el servidor. Permite el reenvío de los mensajes de difusión y multidifusión a los usuarios, mediante la retransmisión en el servicio de UDP, y permite que los mensajes no tengan ninguna modificación (Jiménez, 2019).

2.5. PROTOCOLOS DE ENRUTAMIENTO DINÁMICO

Los protocolos de enrutamiento dinámico son utilizados en redes de gran tamaño con el objetivo de facilitar la sobrecarga operativa y administrativa (Priano, 2021).

2.5.1. PROTOCOLO DE INFORMACIÓN DE ENCAMINAMIENTO (ROUTING INFORMATION PROTOCOLO, RIP)

Este protocolo realiza un encaminamiento interno, se le utiliza para la parte interna de la red que no está conectada directamente al backbone de Internet. Utiliza básicamente un algoritmo de vector de distancia en el cual establece la métrica que va a utilizar, definiendo el número de saltos que realizará entre los routers ya que como máximo establece un número de 15 saltos antes de retirar el paquete enviado (Alvarado Cadena, 2013).

2.5.2. PROTOCOLO DE ENRUTAMIENTO DINÁMICO OPEN SHORTEST PARTH FIRST (OSFP)

El protocolo OSFP, el camino más corto primero, fue desarrollado en reemplazo del protocolo RIP ya que OSFP es un protocolo que no se basa en vectores de distancia, sino que mantiene la topología de la red, teniendo una visión global, lo cual le permite seleccionar el camino más corto para enviar la información. Fue diseñado para utilizarlo en entornos de Internet y sus pilas de protocolos TCP/IP, ya que se usa como un protocolo de ruteo interno, es decir, que distribuye información entre los router del mismo sistema autónomo.

2.5.3. PROTOCOLO DE ENRUTAMIENTO DINÁMICO BORDER GATEWAY PROTOCOL, PROTOCOLO DE LA PASARELA EXTERNA (BGP)

BGP se conoce como un protocolo de puerta de enlace EGP exterior que es utilizado para el intercambio de información de encaminamiento entre los routers de distintos sistemas autónomos. Posee información relacionada con la ruta de cada destino, además utiliza esta información para mantener una base de datos sobre el alcance de la red, que es intercambiada con otros sistemas BGP. Usa políticas de enrutamiento y directivas para escoger entre varios caminos hasta

un destino, controlando la redistribución de la información de enrutamiento (Juniper Networks, 2023).

Usa diferentes parámetros como son: ancho de banda, saturación de la red, precio de la red, denegación de paso de paquetes, entre otros.

En este capítulo se analizaron las características y funcionalidades de los principales componentes que conforman la red SDN, realizando una comparación con las redes tradicionales, con lo que se logran sustentar los fundamentos teóricos de la presente investigación.

Se analizaron los simuladores y controladores que se utilizarán en la propuesta. Se elige a Mininet como emulador de la red SDN, el controlador Aruba Van SDN, el protocolo OpenFlow, las herramientas tecnológicas de virtualización y comunicación VMware Workstation y el router Mikrotik.

CAPÍTULO III. DESARROLLO METODOLÓGICO

3.1. TIPO Y DISEÑO DE LA INVESTIGACIÓN

El objeto de estudio de la presente investigación lo constituye una propuesta de solución virtualizada SDN, utilizando software libre como controlador de infraestructura de red para la Universidad Técnica de Cotopaxi extensión la Maná. El tipo de investigación es cuasi experimental con un enfoque cualitativo, diseñado para guiar el proceso de investigación científica en cada etapa, permitiendo la determinación de los fundamentos teóricos, la selección de las herramientas y la propuesta de solución que valida los resultados alcanzados.

3.2. METODOLOGÍA PPDIOO

Se utilizó la metodología PPDIOO que ofrece las facilidades para el desarrollo e implementación de redes, a un costo reducido, la cual es propuesta por CISCO, teniendo en cuenta elementos como la organización y contempla las fases necesarias para la obtención de requerimientos, como se presentan en la figura 7, convirtiéndose en un estándar en la actualidad por sus beneficios, permitiendo verificar los requisitos, aplicaciones o servicios que se requieran, los "dispositivos de red actuales y los nuevos que se van a colocar, realizar un análisis de la red actual y de la propuesta para el cambio de red" (Aguilar Peña, 2021).

Figura 7. Metodología PPDIOO.
Fuente: (Cisco System, 2016)

3.2.1. FASE DE PREPARACIÓN

Consistió en investigar y analizar las diferentes tecnologías y herramientas de comunicación, router, controladores y sistemas programables para redes y centros de datos definidos por software disponibles en el mercado, sean estas tanto privados como libres, con el propósito de minimizar costos de implementación y despliegue al momento de la puesta a punto de la red definida por software SDN.

Como se ha expuesto en el capítulo anterior, los avances y costos elevados que representa el diseño y construcción de una red física, están incitando a que los administradores de red opten por contratar recursos virtuales, siendo las redes SDN la opción más eficaz, integrando las redes programables con el sistemas de gestión y administración como controladores SND, los cuales pueden ser escalables, dependiendo de los requerimientos del cliente, por lo que el despliegue de toda esta infraestructura será a nivel de SND, además se hará una prueba con el sistema Mininet y el controlador SDN definido por software.

En esta fase se definen los componentes más importantes para la elaboración de la arquitectura virtualizada SDN en un laboratorio controlado (Tabla 1).

Tabla 1. Componentes de la infraestructura de red.

Infraestructura Tecnológica
a. Mikrotik
b. VMware Workstation
c. Controlador SDN (HPE Linux / Aruba Van SDN Controller)
d. Mininet (Python, Topología, Lineal, Plana, Estrella)
e. Sistemas Operativos Windows (Windows 7 y Windows 10)
f. Sistemas Operativos Linux / Ubuntu (Ubuntu 14.04.4 y Ubuntu 22.04)

Fuente: Autores

En tal sentido se describe en la Tabla 2, toda la tecnología existente en la Universidad Técnica De Cotopaxi Extensión La Maná, por un valor total de $35538.25 en el mercado, de esta forma, no se pretende hacer nuevas

inversiones, sólo modificar el enfoque de la red tradicional a una red SDN, desde la división de la Capa de Control de los dispositivos para la comunicación con la capa subyacente, permitiendo su control y gestión centralizada, apoyado en los beneficios ya descritos, así como su rápida implementación y ahorro de costos.

Tabla 2. Análisis de costo de los dispositivos de la red

Cantidad	Descripción	Valor Unidad	Valor Total
1	ACCESS POINT UBIQUITI NANOSTATION 2	68.00	68.00
10	ACCESS POINT 2.4GHZ, NANOSTATION 2 UBIQUITI	246.4	2464.00
1	BIENES SUJETOS A CONTROL/SWITCH	481.6	481.6
1	DISPOSITIVO DE AUTENTICACIÓN FAC-400E. CUR 2531	12,004.4	12,004.4
1	DISPOSITIVO DE AUTENTICACIÓN FAC-400E. CUR 2531	19,575.3	19,575.3
35	DISPOSITIVOS DE ACCESO INALÁMBRICO FortiAP-321C, INCLUYE (5) 1-PORT GIGABIT PoE POWER INJECTOR, 802.3af 15.4Watts. CUR 2531	820.81	28,728.35
8	DISPOSITIVOS DE ACCESO SWITCHES. CUR 2531	420.34	3,362.72
7	SWITCH HP 24 PUERTOS 10/100/1000 ADMINISTRABLES HP 1910 -24 G	483.84	3,386.88
1	SWITCH WEB MANAGED 24XRJ45 10/100/1000 4 PORTS - JG924A HP 1920-24G (CUR 5368)	455.98	455.98

Fuente: Autores

3.2.2. FASE DE PLANIFICACIÓN

La planificación supone la definición de requisitos para la creación de una infraestructura que permita la interconexión armónica de todos los componentes de la red y por consiguiente la prestación de servicio a todos los usuarios de la Universidad Técnica De Cotopaxi Extensión La Maná. Inicialmente se realiza una identificación de los requerimientos de la red, haciendo una evaluación y caracterización de la misma, basándose en el análisis de los puntos débiles, así como la aplicación del estudio de las mejores prácticas.

Como parte del desarrollo de esta etapa, se tienen en cuenta los resultados que se pretenden obtener y los elementos necesarios que participan en el proceso de implementación (Topología, Bloques IP, Routers, etc.) para la virtualización del laboratorio controlado desde donde se harán las diferentes pruebas que justifican la propuesta.

Para el despliegue de la investigación es necesario el cumplimiento de los siguientes requerimientos descritos en la Tabla 3.

Tabla 3. Componentes de la infraestructura de red

Herramienta	Tecnología	Versión	Libre/ propietaria
Mikrotik CHR 49	Mikrotik chr-6.49.4	6.49.4 (x86_64)	Libre
Mikrotik CHR 49	Mikrotik chr-6.49.4	6.49.4 (x86_64)	Libre
Mikrotik CHR 49	Mikrotik chr-6.49.4	6.49.4 (x86_64)	Libre
SDN VAN CONTROLLER	Aruba Van HP	HPE-van-sdn-ctlr-2.8.8	Libre
Mininet	Linux Ubuntu	2.3.0 - 210211	Libre
Windows 7	Windows	Windows 7 sp2	Propietaria
Windows 10	Windows	Windows 10 22H2	
Ubuntu	Linux	20.04.1	Libre
Plataformas y Servicios			
Infraestructura robusta y tolerante a fallos.			
Plataforma de virtualización.			
Integración de sistemas de conmutación en el plano de control y datos			

Fuente: Autores

3.2.3. FASE DE DISEÑO

Se partió del diseño actual de la red de la Universidad Técnica de Cotopaxi Extensión la Maná y se propone uno nuevo, incorporando los componentes necesarios para el despliegue de la red definida por software SDN, con el controlador de gestión y administración mediante el protocolo OpenFlow, en la figura 8 se presenta el diagrama actual de la red.

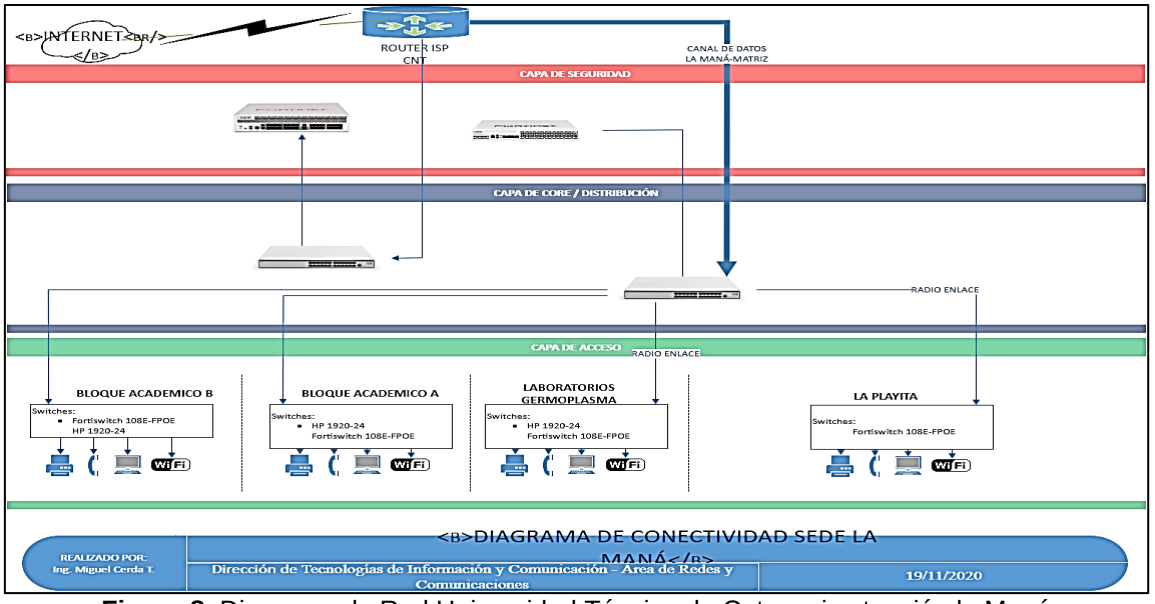


Figura 8. Diagrama de Red Universidad Técnica de Cotopaxi extensión la Maná
Fuente: Autores

Topología SDN

Como propuesta para el desarrollo de la investigación y las fases de implementación de la metodología utilizada, se representa un nuevo esquema de red, en el que todos los elementos que intervienen trabajen de manera síncrona. El cual surge debido a que los centros de datos no pueden responder a patrones de tráfico impredecibles.

Por lo que se tendrían dos alternativas: migrar a una red más costosa, invirtiendo dinero en nuevos equipamientos y tiempo para su configuración, o adaptar el escenario actual a una red definida por software SDN, enfocada a docentes, estudiantes y personal administrativo que requieren cambios dinámicos, rápidos y con costos mínimos.

Para la ejecución de esta investigación se plantea un escenario de direccionamiento IP (Tabla 4), usado en la topología SDN propuesta en el esquema de red que se muestra en la Figura 9.

Tabla 4. Direccionamiento IP.

Herramienta	Dirección Ip	Gateway	Interfaz	Descripción
Mikrotik R1	192.168.5.91/29	none	Ether1	WAN
	192.168.240.2/30	192.168.240.1	Ether2	Internet
	28.8.35.1/24	none	Ether3	DMZ
	10.0.0.1/23	none	Ether3	LAN
Mikrotik R2	192.168.5.92/29	192.168.5.91	Ether1	WAN
	10.0.2.1/23	none	Ether2	LAN
Mikrotik R3	192.168.5.93/29	192.168.5.91	Ether1	WAN
	10.0.4.1/23	none	Ether2	LAN
CONTROLLER	10.0.2.8/23	10.0.2.1	Ether1	LAN
Mininet	10.0.0.10/23	10.0.0.1	Ether1	LAN
WINDOWS	10.0.2.10/23	10.0.2.1	Ether1	LAN
	10.0.4.10/23	10.0.4.1	Ether1	LAN

Fuente: Autores

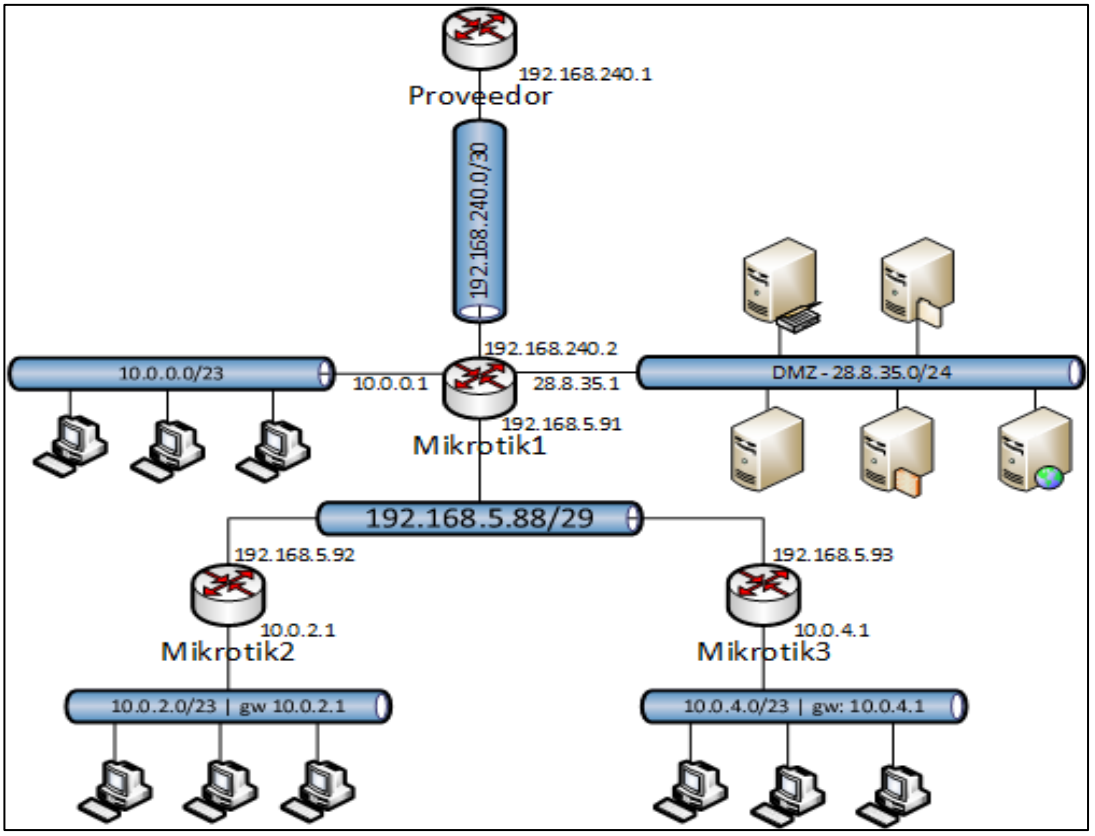


Figura 9. Topología de red propuesta
Fuente: Autores

3.2.4. FASE DE IMPLEMENTACIÓN

Una vez definido los mecanismos de diseño y la respectiva arquitectura de red se proceden a la implementación tecnológica con las herramientas definidas en la fase anterior, como complemento base para el despliegue e implementación de la red SND virtual, netamente definida por software y con la instalación y creación de las máquinas virtuales. Para lograr los objetivos de la presente investigación es fundamental indicar que todos los sistemas utilizados en el laboratorio controlado son de libre distribución, con la excepción de los sistemas operativos de tipo Windows (Tabla 5).

Tabla 5. Máquinas Virtuales y Herramientas Utilizadas

MÁQUINAS VIRTUALES	HERRAMIENTAS
Windows cliente (Windows 10) (1)	Soporte para la virtualización
Windows cliente (Windows 7) (2)	Administración
Router Mikrotik (3)	Rutas, Openflow
Mininet Ubuntu 20.04	Python
SDN VAN CONTROLLER	Data Plane

Fuente: Autores

Para la instalación de los router Mikrotik se procedió a descargar desde la página oficial, la imagen estable CHR en su versión v6.49.4 (Figura 10), cabe indicar que son imágenes muy ligeras que permiten realizar una configuración y gestión muy amplia, además se realizó la integración de las interfaces de red correspondientes para los enlaces entre los demás router y los clientes de la LAN como se propuso en la topología de red presentada anteriormente.

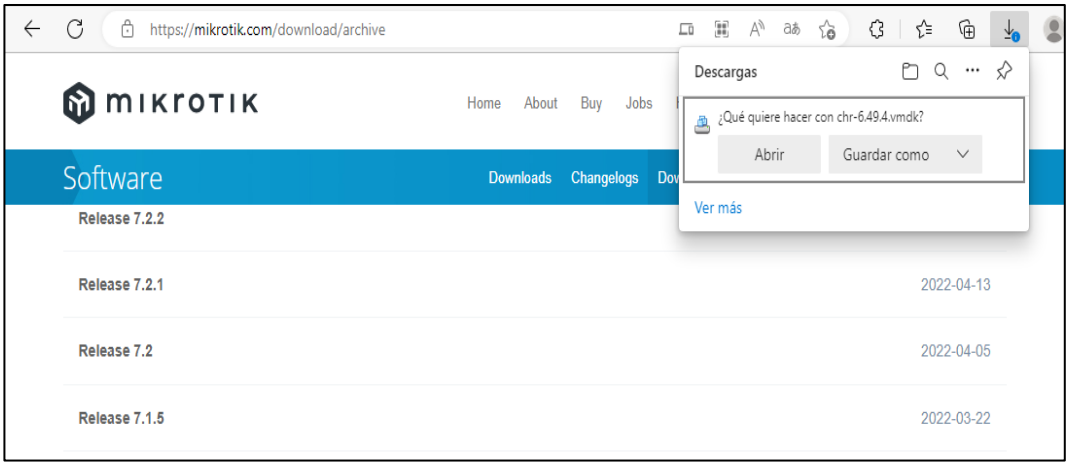


Figura 10. Descarga del Sistema del Router Mikrotik.
Fuente: Autores

Se realiza el proceso de instalación (Figura 11), teniendo en cuenta previamente las características, las prestaciones de cada dispositivo por separado en cuanto a las interfaces de red y tomando como ejemplo el router 2, donde se habilitarán las interfaces correspondientes a la WAN y la LAN, coincidiendo con el esquema para la red SDN propuesta.

Figura 11. Instalación del Router Mikrotik.
Fuente: Autores

Esta acción se realiza en un ambiente completamente controlado por lo que no existen riesgos de dañar ningún dispositivo físico. Mediante el comando *interface print* se pueden comprobar las interfaces establecidas como se observa en la Figura 12.

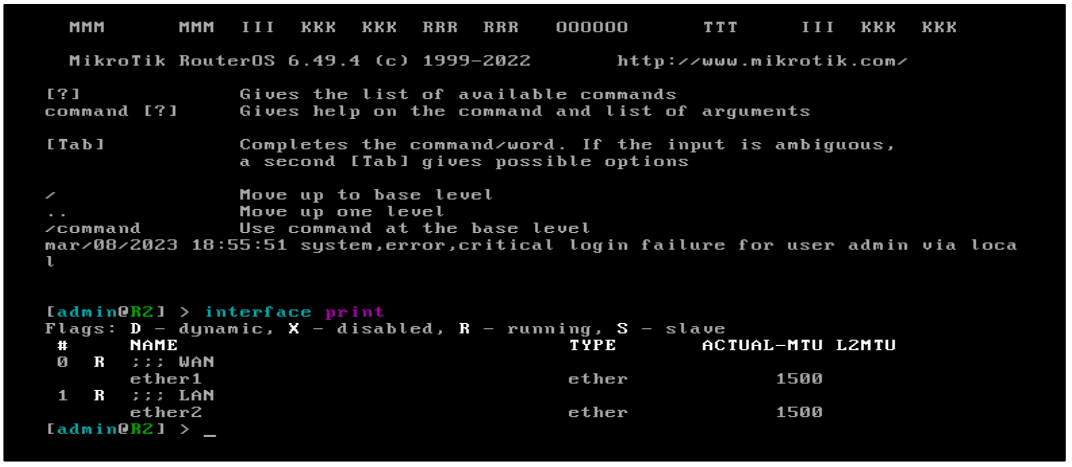


Figura 12. Revisión de interfaces de red Router Mikrotik.
Fuente: Autores

Para la administración del router Mikrotik existen varias formas (desde la propia consola, utilizando el winbox y, una vez configurada al menos una interface, desde una página web), en esta ocasión se realiza la gestión mediante el cliente de administración winbox, como se muestra en la Figura 13.

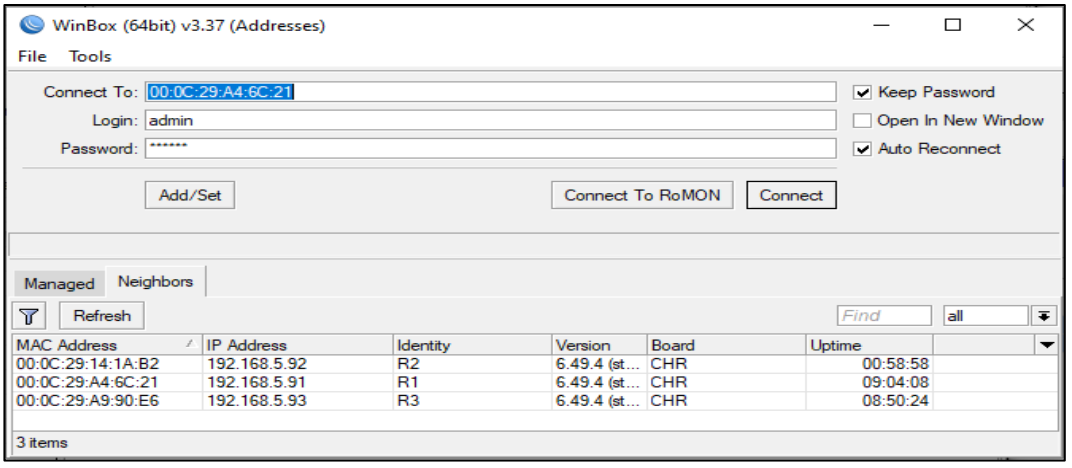


Figura 13. Acceso a Router Mikrotik mediante winbox.
Fuente: Autores

Una vez dentro de la administración del router, haciendo uso de la herramienta winbox, se dispone de varias opciones, esta ventana principal es el Dashboard de Mikrotik y es donde se van a realizar todas las configuraciones respectivas (Figura 14).

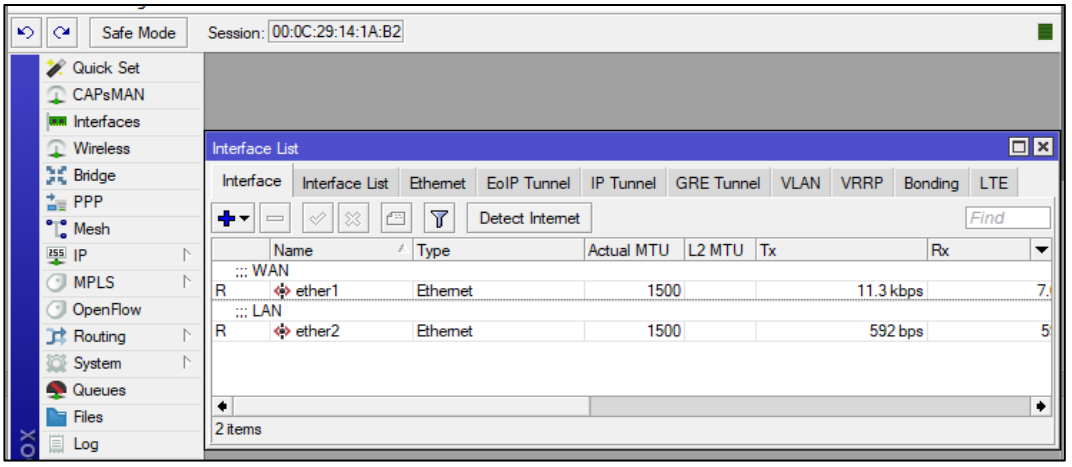


Figura 14. Acceso a Router Mikrotik mediante winbox - Dashboard
Fuente: Autores

Luego se comienza a definir el direccionamiento IP para las interfaces WAN y LAN, en las interfaces ether1 y ether2, tal como indica la Figura 15, donde se precisaron los enlaces con sus segmentos de red.

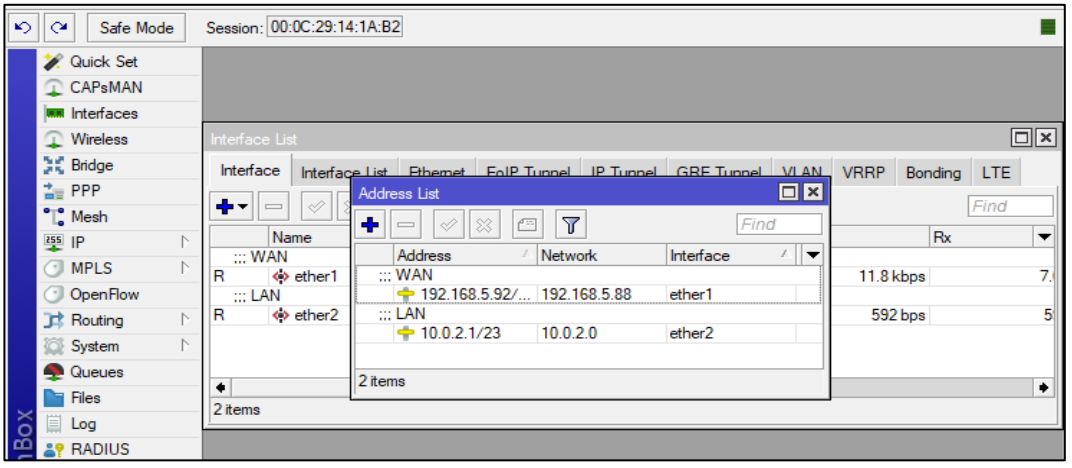


Figura 15. Configuración de interfaces de red Router Mikrotik mediante winbox
Fuente: Autores

Para que las computadoras conectadas a la red LAN del router 2 puedan ver el resto de las computadoras o servicios que están detrás de los otros dos router es necesario configurar las políticas de ruteo, para esto se establecen las rutas y el Gateway correspondiente dentro de la opción IP/Routes, donde se coloca la dirección de la puerta de enlace correspondiente (Figura 16).

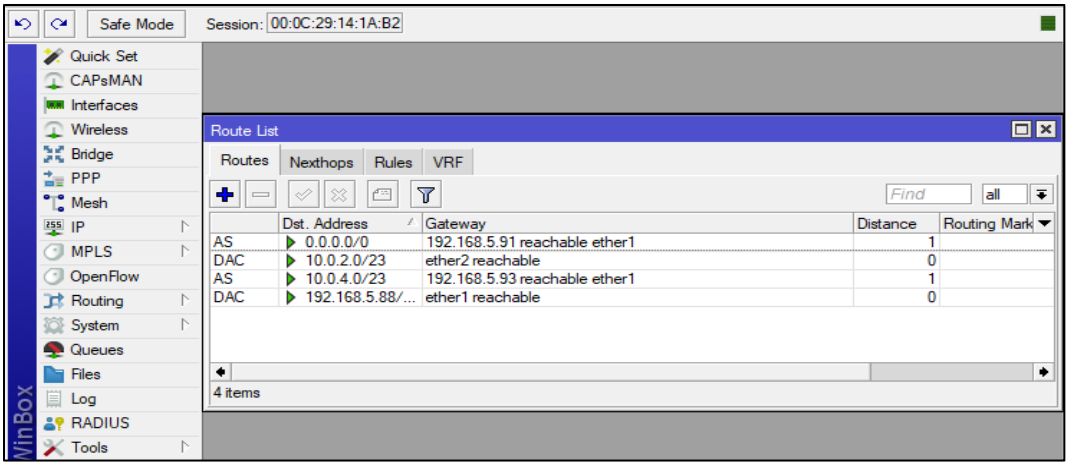


Figura 16. Configuración enrutamiento en router Mikrotik
Fuente: Autores

Posteriormente se procede a instalar el protocolo OpenFlow en cada router Mikrotik. Para ellos se descarga desde el sitio oficial (https://download.mikrotik.com/) el paquete correspondiente para su instalación (routeros/6.49.4/all_packages-x86-6.49.4.zip), en el caso de estudio se descargó la versión que incluye todos los paquetes, seguidamente se extrajo en una carpeta (Figura 17) para luego subirlo al router.

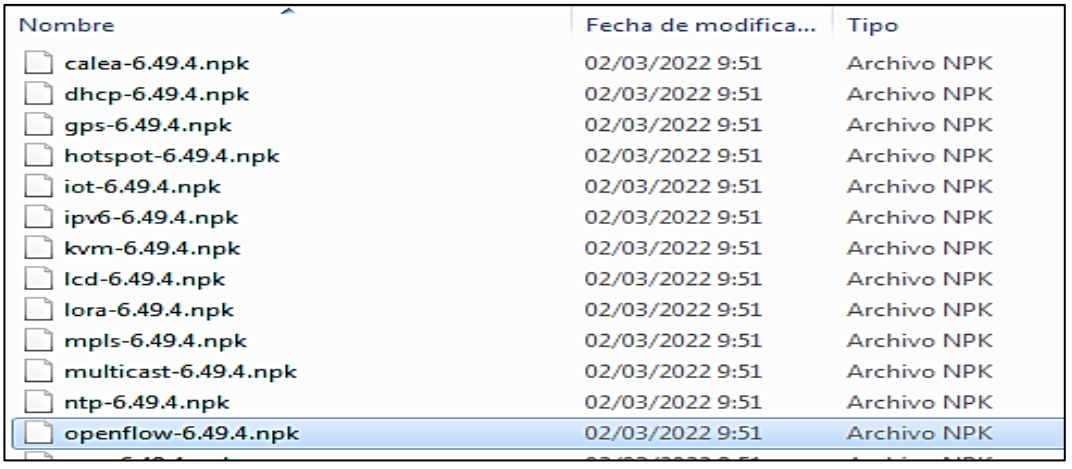


Figura 17. Descarga del paquete para el protocolo OpenFlow en router Mikrotik
Fuente: Autores

Desde el winbox y utilizando la opción Files del menú de la izquierda, se pueden instalar los paquetes deseados, para este caso como se muestra en la Figura 18, se sube el paquete del OpenFlow, a continuación, solo se reinicia el router y queda instalado y listo para usar el OpenFlow.

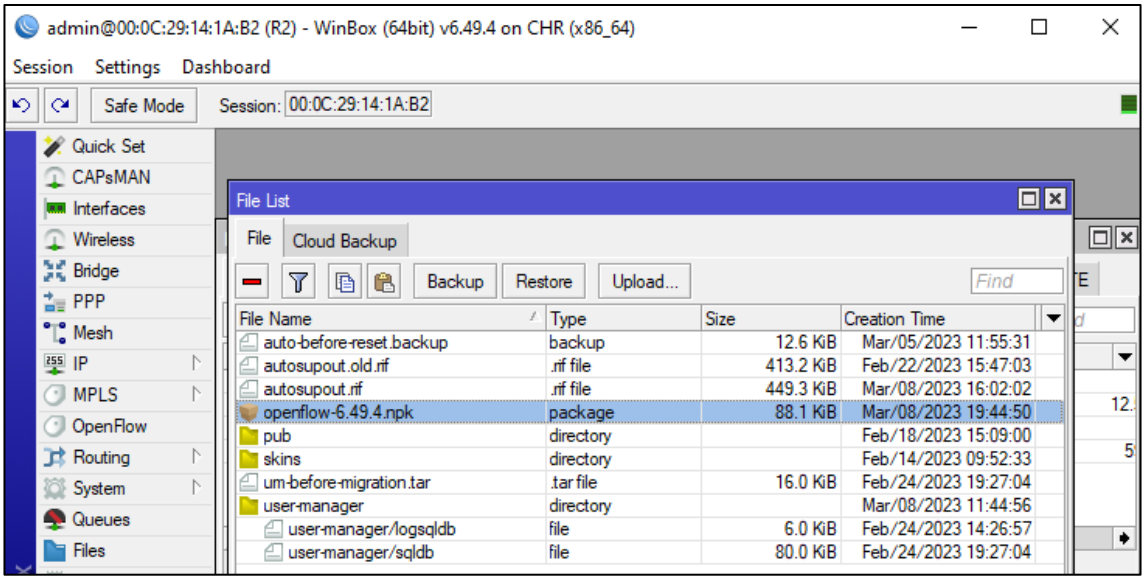


Figura 18. Instalación del paquete OpenFlow en router Mikrotik.
Fuente: Autores

Mediante el protocolo OpenFlow, una red puede ser gestionada de manera centralizada, para esto es necesario conocer el direccionamiento IP del controlador y las interfaces correspondientes que se van a integrar al protocolo tal como indica la Figura 19, permitiendo así la comunicación y gestión con el controlador.

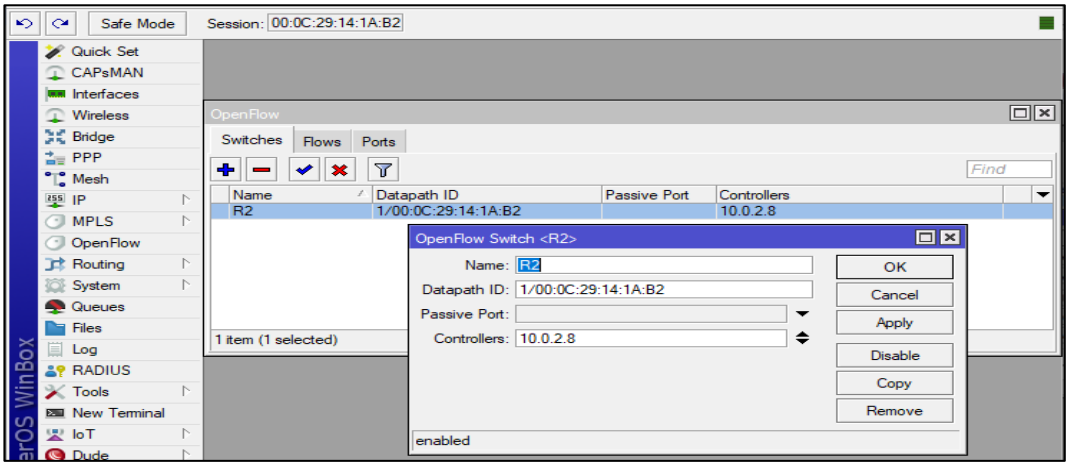


Figura 19. Configuración del protocolo OpenFlow en router Mikrotik.
Fuente: Autores

Es necesario, entonces, la definición de las interfaces de red que van a permitir la integración de las comunicaciones, para este caso las interfaces

establecidas que forman parte de este protocolo son, ether1 que pertenece a la red WAN y ether2 que pertenece a la red LAN, tal como indica la Figura 20.

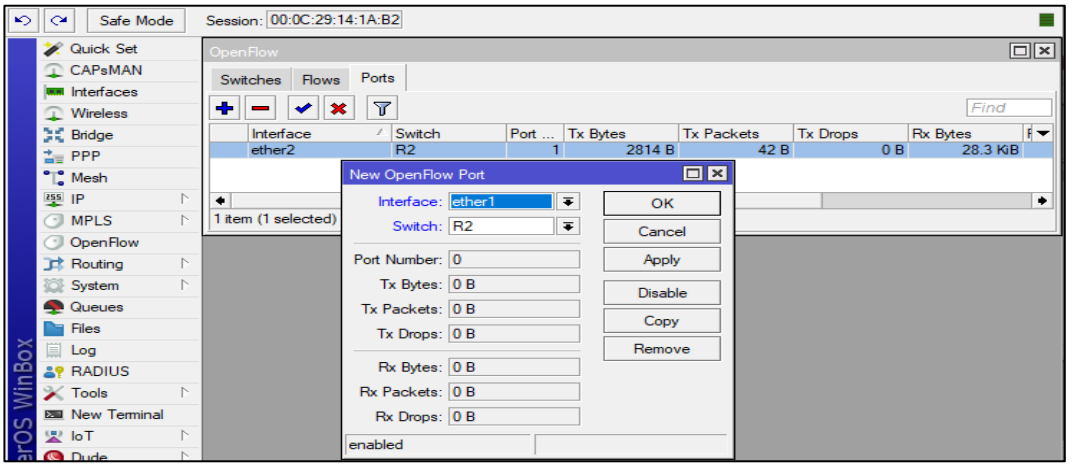


Figura 20. Configuración de las interfaces dentro del protocolo OpenFlow en el router Mikrotik
Fuente: Autores

Una vez integrados los puertos y definido el controlador (SDN Controller) ya se puede revisar el flujo de paquetes, lo que permite la gestión centralizada de una red netamente definida por software como indica la Figura 21.

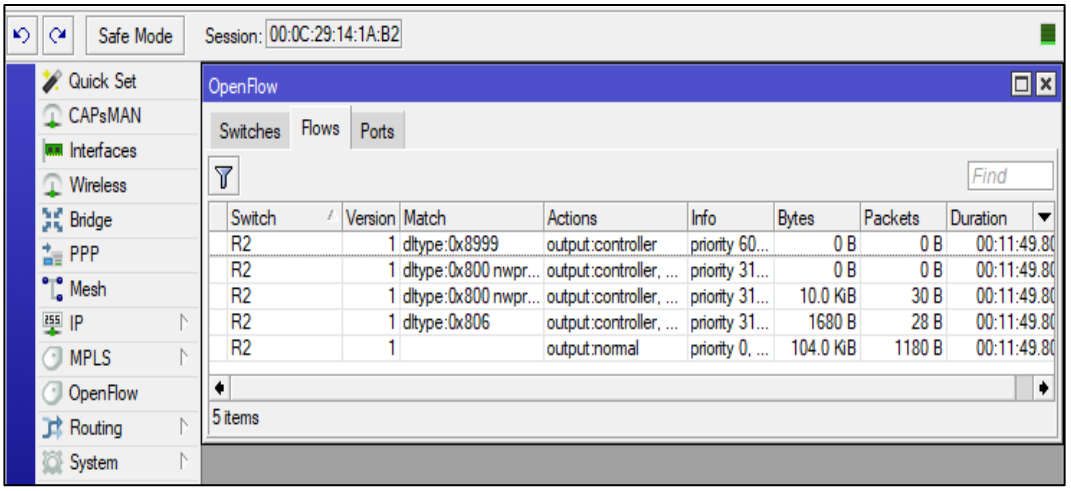


Figura 21. Flujos de paquetes OpenFlow en router Mikrotik.
Fuente: Autores

Es fundamental realizar la misma configuración en cada uno de los router del laboratorio, para este ejemplo se utilizaron 3 routers Mikrotik con los componentes OpenFlow integrados y el respectivo protocolo de enrutamiento.

Instalación de Mininet

Mininet es un emulador para el despliegue de red sobre los recursos de un ordenador en una máquina virtual. Puede trabajar con máquinas preestablecidas o realizar una instalación limpia y nativa, mediante el uso de comandos (Tabla 6), la cual, utiliza el kernel de Linux y otros componentes para emular elementos de la red definida por software SDN con controlador, los switches OpenFlow y los hosts.

Tabla 6. Comandos para la instalación del Mininet.

Comando de instalación del Mininet	
Comandos necesarios para actualizar el sistema operativo	sudo apt-get update sudo apt-get upgrade sudo apt-get dist-upgrade
Comandos necesarios para clonar e instalar Mininet en la máquina.	sudo apt-get install git git clone git://github.com/mininet/mininet cd mininet git tag git checkout -b 2.2.2

Fuente: Autores

Para realizar la instalación limpia de Mininet se deben ejecutar individualmente los pasos con los comandos indicados en la Tabla 5, si el proceso no presenta ninguna novedad, como resultado, el sistema Mininet se ha instalado correctamente (Figura 22).

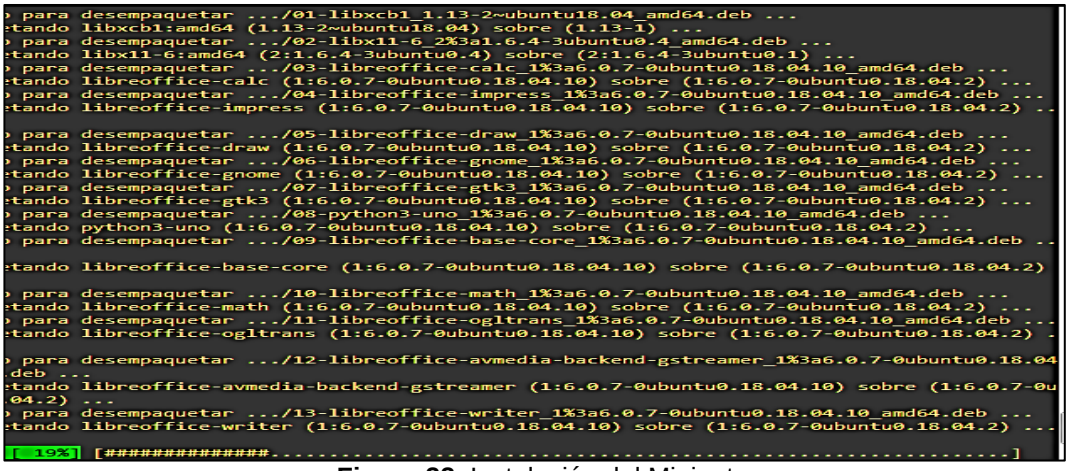


Figura 22. Instalación del Mininet
Fuente: Autores

El procedimiento de creación de una topología de red, haciendo uso de Mininet, es muy sencillo usando un script en Python como indica la Figura 23, donde se programa una red SND, con 3 Switches, 10 hosts y un controlador, adicionalmente se agregan los nodos a la red mediante los links de comunicación.

```
    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                         controller=RemoteController,
                         ip='10.0.2.8',
                         protocol='tcp',
                         port=6633)

    info( '*** Add switches\n')
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch, protocols=["OpenFlow10"])
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch, protocols=["OpenFlow10,OpenFlow13"])
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch, protocols=["OpenFlow10,OpenFlow13"])

    info( '*** Add hosts\n')
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.15', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.20', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='10.0.0.30', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='10.0.0.40', defaultRoute=None)
    h5 = net.addHost('h5', cls=Host, ip='10.0.0.50', defaultRoute=None)
    h6 = net.addHost('h6', cls=Host, ip='10.0.0.60', defaultRoute=None)
    h7 = net.addHost('h7', cls=Host, ip='10.0.0.70', defaultRoute=None)
    h8 = net.addHost('h8', cls=Host, ip='10.0.0.80', defaultRoute=None)
    h9 = net.addHost('h9', cls=Host, ip='10.0.0.90', defaultRoute=None)
    h10 = net.addHost('h10', cls=Host, ip='10.0.0.100', defaultRoute=None)

    info( '*** Add links\n')
    net.addLink(s1, s2)
    net.addLink(s1, s3)
    net.addLink(h1, s1)
    net.addLink(h2, s1)
    net.addLink(h3, s1)
    net.addLink(h4, s1)
```

Figura 23. Creación de red programable en Python con Mininet.
Fuente: Autores

Para la ejecución de este script solo es necesario iniciarlo como se indica en la Figura 24.

```
mininet@mininet-vm:~$ sudo su
root@mininet-vm:/home/mininet# python /home/mininet/mininet/examples/mn_remote_controller.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Results: 0% dropped (90/90 received)
mininet> _
```

Figura 24. Ejecución de script con Python con Mininet.
Fuente: Autores

Instalación de Aruba VAN SDN Controller

El controlador SDN VAN controller proporciona una administración centralizada y unificada habilitada para “redes SDN, lo que simplifica la administración, el desempeño y la organización. Esto permite la gestión de una nueva generación de servicios de red basados en aplicaciones. Ofrece interfaces de programación de aplicaciones (API) abiertas, permitiendo a los desarrolladores crear soluciones transformadoras” (Chafloque Mejia, 2018). Los recursos de instalación para el SDN VAN Controller se definen en la Figura 25.

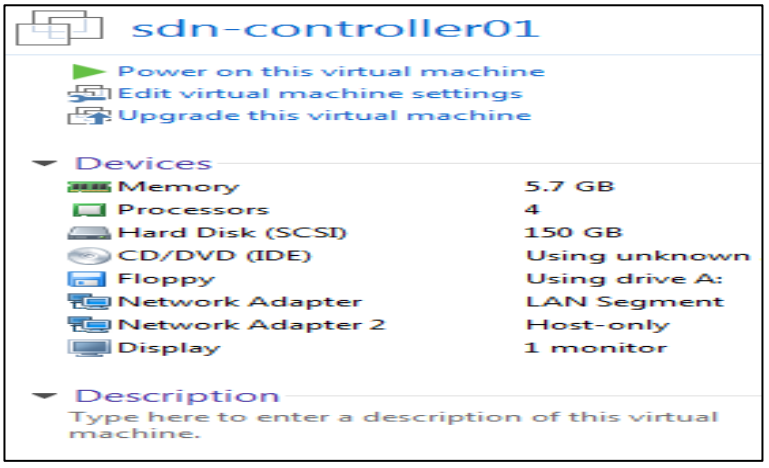


Figura 25. Recursos para la instalación del SDN VAN Controller.
Fuente: Autores

Acceso al SDN VAN Controller

Para acceder al controlador SND es necesario un navegador web, luego se debe de colocar la dirección IP del servidor SDN VAN Controller y seguido el puerto de conexión, que por defecto es el 8443, para este laboratorio la IP que se empleó es la 10.0.2.8 y el puerto de acceso por defecto (8443). Una vez que se accede a la URL definida por IP y al puerto (https://10.0.2.8:8443), se solicita que se ingrese el usuario y la clave, tal como indica la Figura 26.

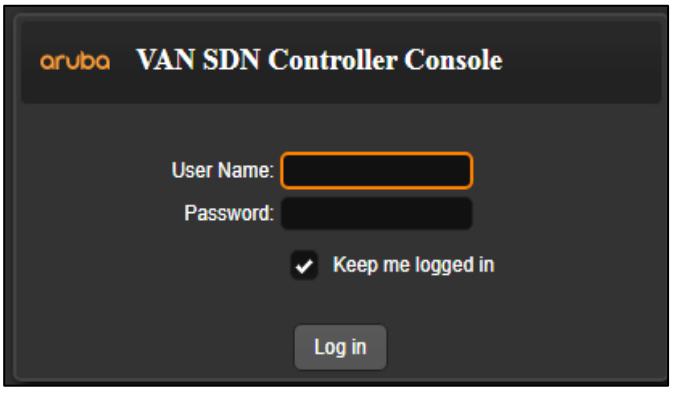


Figura 26. Acceso web al SDN VAN Controller.
Fuente: Autores

En esta fase, si todo se ha configurado correctamente, usando la opción Openflow Topology se puede observar la topología de red creada y definida por software. Trabajando por el protocolo OpenFlow, el controlador muestra los router Mikrotik y los hosts que pertenecen a cada red LAN, como indica la Figura 27.

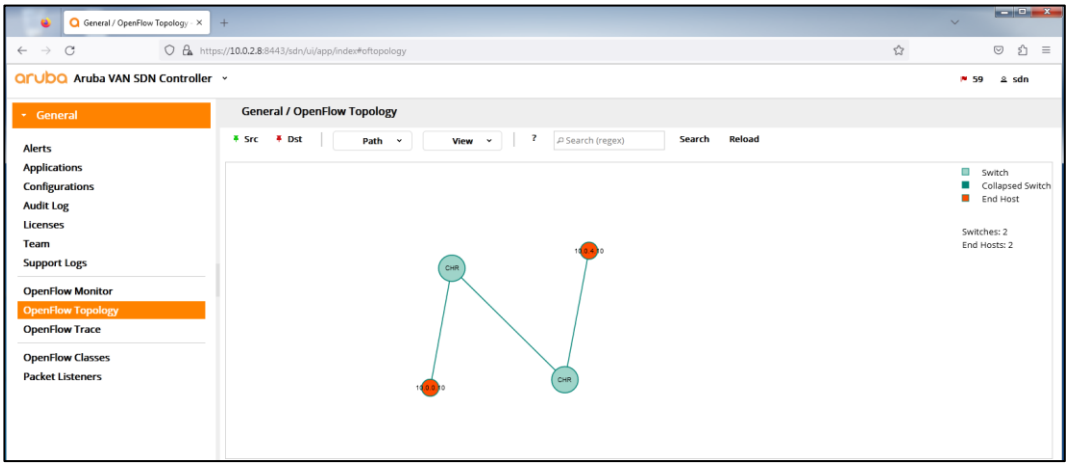


Figura 27. Topología de red SDN VAN Controller.
Fuente: Autores

La figura 28 muestra mediante el proceso de actualización de la interface web del servidor SDN VAN Controller representado por la dirección IP 10.0.2.8 (URL https://10.0.2.8:8443), la presencia de una máquina cliente de Windows con la dirección IP 10.0.2.10, otra con la dirección IP 10.0.4.10 y la IP 10.0.0.10 de la máquina con el Mininet, pertenecientes cada una a segmentos de red diferentes, pero entre todos tienen comunicación debido al proceso de enrutamiento que existe.

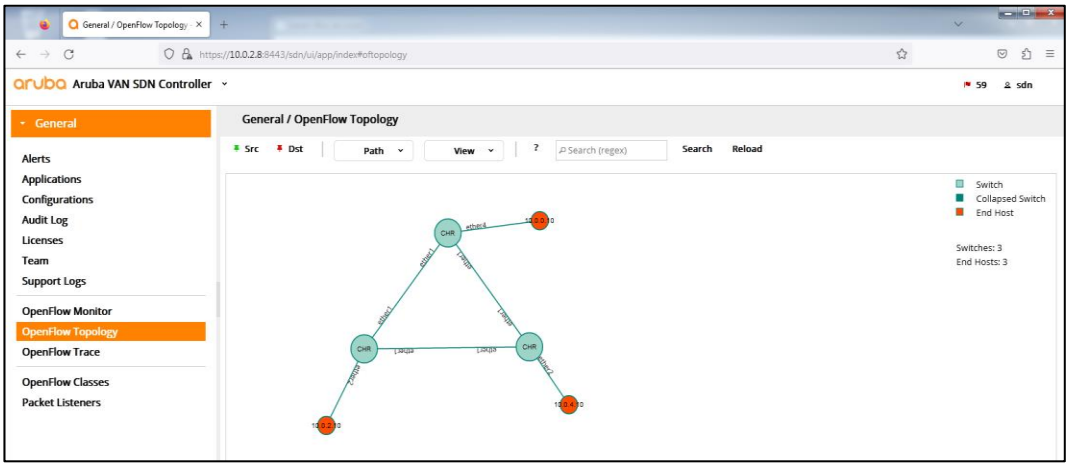


Figura 28. Topología de red SDN VAN Controller - actualización.
Fuente: Autores

Dentro de la opción OpenFlow monitor se pueden observar todos los equipos router que son gestionados por el controlador SDN VAN Controller, con los detalles de cada conmutador (Figura 29).

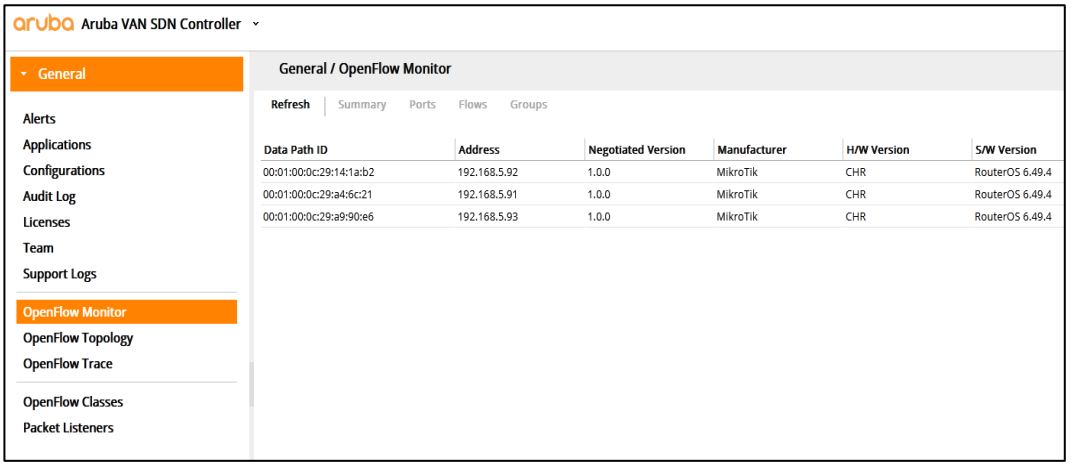


Figura 29. Visualización de equipos conmutadores de la red SDN VAN Controller.
Fuente: Autores

Un conmutador OpenFlow consta de una o varias tablas de flujo y una tabla de grupo, la que permite realizar búsquedas y reenvíos de paquetes; un canal OpenFlow a un controlador externo donde el conmutador se comunica con el controlador; el sistema Mininet y la integración con los router Mikrotik. De esta forma se puede dirigir hacia cualquier router y visualizar los flujos dentro de los mismos (Figura 30).

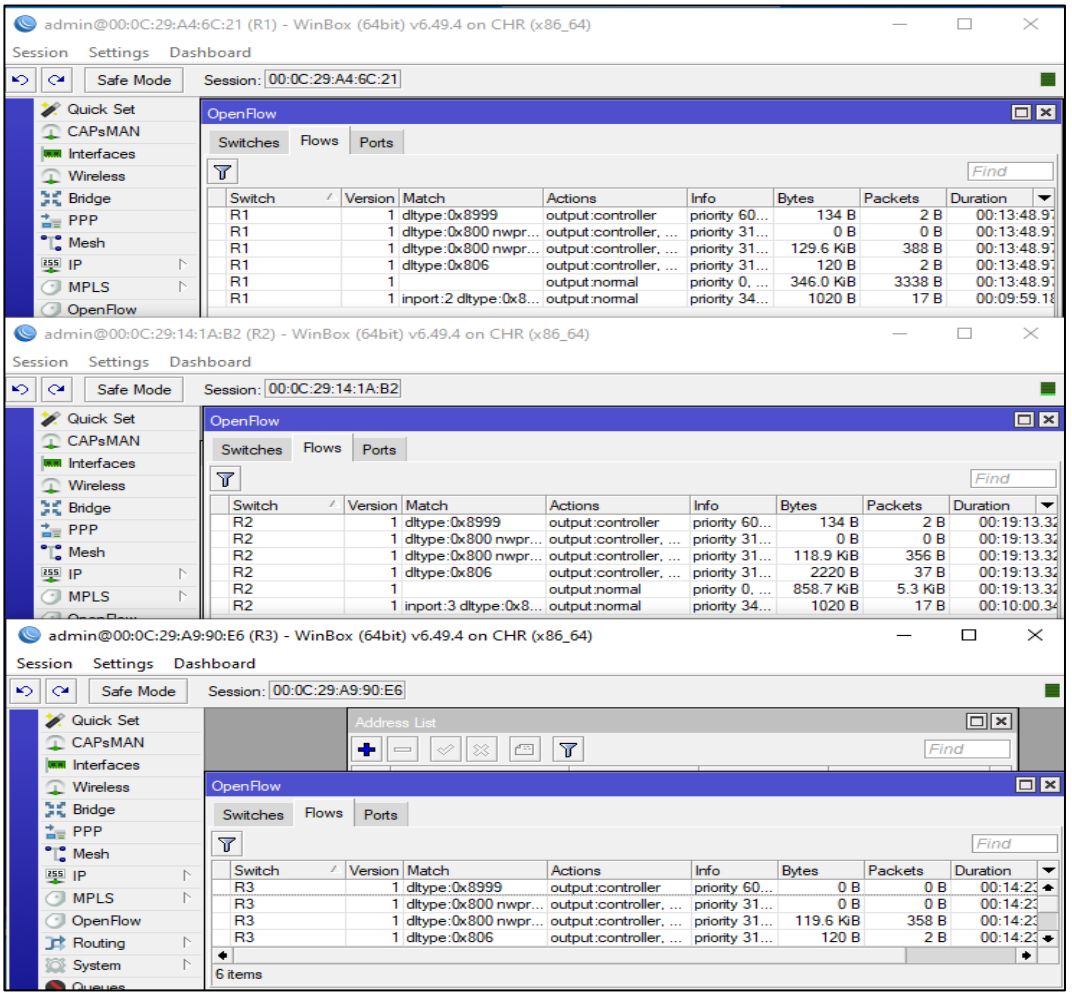


Figura 30. Flujos OpenFlow de la red SDN VAN Controller.
Fuente: Autores

La Figura 31 muestra la topología final integrada con los 3 router Mikrotik y el sistema de programación de redes SDN Mininet. Se procedió a crear la topología de red SDN mediante el comando sudo mn --controller=remote,ip=10.0.2.8 --topo=single, 5, para ampliar el número de los hosts del controlador SDN VAN

Controller y los nodos, en este punto es fundamental realizar pruebas de alcance ICMP.

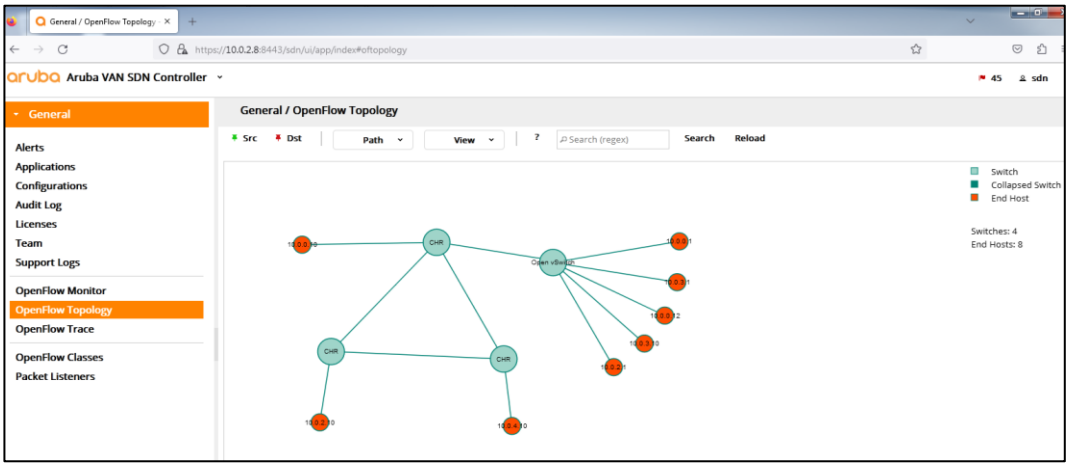


Figura 31. Topología OpenFlow de la red SDN VAN Controller.
Fuente: Autores

Para realizar las pruebas de comunicación entre todos los equipos y la red SDN se usaron las opciones Src y Dst. Para este caso, se seleccionaron el origen que es el host 10.0.2.10 y el destino que corresponde al host 10.0.3.1 y de manera automática se traza la ruta, dando como resultado satisfactorio, ver Figura 32.

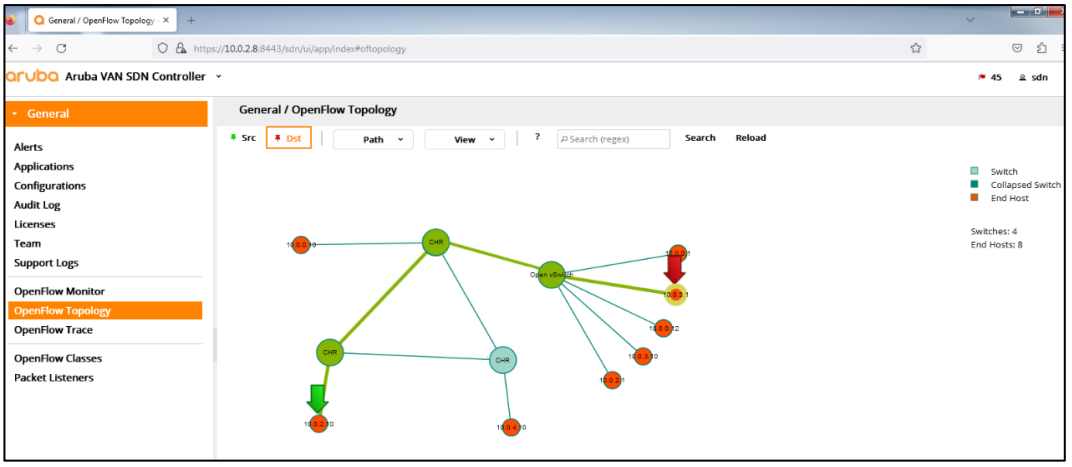


Figura 32. Pruebas de Funcionamiento de la red SDN VAN Controller.
Fuente: Autores

3.2.5. FASE DE OPERACIÓN

En esta fase, se muestra el funcionamiento de la red desplegada sobre el laboratorio de pruebas SDN, implementado en base a los requerimientos plenamente establecidos en las fases anteriores. Una vez definida las interfaces, se puede realizar una programación de nuevas redes programables o realizar la respectiva monitorización del funcionamiento de la red SDN, se deben revisar constantemente las tablas de ruteo, actualizar nuevas rutas, si fuera el caso, o la respectiva actualización del controlador VAN.

Tabla 7. Programación de topología SDN segmentada

Programación de topología Segmentada SDN

```
#!/usr/bin/python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf, OVSLink
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                   build=False,
                   link=OVSLink,
                   ipBase='10.0.0.0/23')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                         controller=RemoteController,
                         ip='192.168.5.94',
                         protocol='tcp',
                         port=6633)

    info( '*** Add switches\n')
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch, protocols=["OpenFlow10"])
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch, protocols=["OpenFlow10"])
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch, protocols=["OpenFlow10"])
    # s1 = net.addSwitch('s1', cls=OVSKernelSwitch, protocols=["OpenFlow10,OpenFlow13"])
    # s2 = net.addSwitch('s2', cls=OVSKernelSwitch, protocols=["OpenFlow10,OpenFlow13"])

    info( '*** Add hosts\n')
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.15', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.20', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='10.0.0.30', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='10.0.0.40', defaultRoute=None)
    h5 = net.addHost('h5', cls=Host, ip='10.0.0.50', defaultRoute=None)
    h6 = net.addHost('h6', cls=Host, ip='10.0.0.60', defaultRoute=None)
    h7 = net.addHost('h7', cls=Host, ip='10.0.0.70', defaultRoute=None)
    h8 = net.addHost('h8', cls=Host, ip='10.0.0.80', defaultRoute=None)
    h9 = net.addHost('h9', cls=Host, ip='10.0.0.90', defaultRoute=None)
    h10 = net.addHost('h10', cls=Host, ip='10.0.0.100', defaultRoute=None)

    info( '*** Add links\n')
    net.addLink(s1, s2)
    net.addLink(s1, s3)
    net.addLink(s2, s3)
    net.addLink(h1, s1)
    net.addLink(h2, s1)
    net.addLink(h3, s1)
    net.addLink(h4, s1)
    net.addLink(h5, s1)
    net.addLink(h6, s1)
    net.addLink(h7, s1)
    net.addLink(h8, s1)
    net.addLink(h9, s1)
    net.addLink(h10, s1)
    info( '*** Starting network\n')
    net.build()
```

Fuente: Autores

La tabla 7 muestra el código fuente de la programación de una red SDN, con la creación de Open VSwitch y 10 host, cabe indicar que esta topología está segmentada debido a que se puede acceder desde cualquier parte de la red. Se pueden observar los flujos de paquetes enviados al controlador para la gestión correspondiente, se configura la opción IP remota 10.0.2.8 del controlador con el puerto de gestión 6633.

Se procede a ejecutar las instrucciones para el script de la topología SDN descrita en la Tabla 6. La Figura 33 muestra el resultado correspondiente a la ejecución del script dentro del Mininet. En este punto ya se puede gestionar mediante el controlador VAN. Se han añadido el controlador, los switches, los hosts y los links para los flujos.

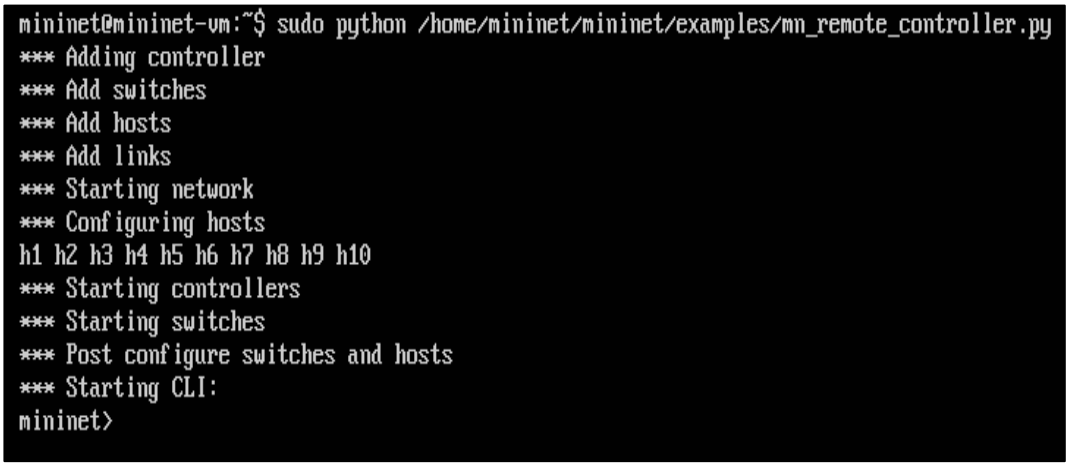


Figura 33. Resultado de la creación de la topología SDN en Mininet.
Fuente: Autores

Para comprobar el correcto funcionamiento de la red SDN, se procede a probar que todos los hosts tengan comunicación entre ellos, para eso se ingresa el comando pingall sobre la consola de Mininet y como regulador devuelve la respuesta inmediata como muestra la Figura 34.

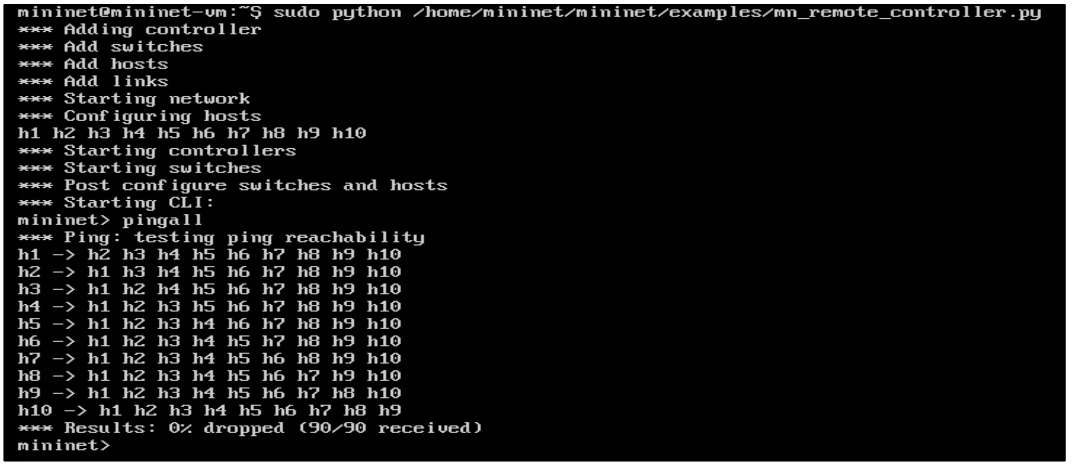


Figura 34. Respuesta de comunicación ICMP de la topología SDN en Mininet.
Fuente: Autores

Se revisan las diferentes subredes, realizando las comprobaciones mediante el comando net sobre la consola de Mininet, obteniendo el resultado que muestra la Figura 35: todos los hosts conectados al Switch 1.

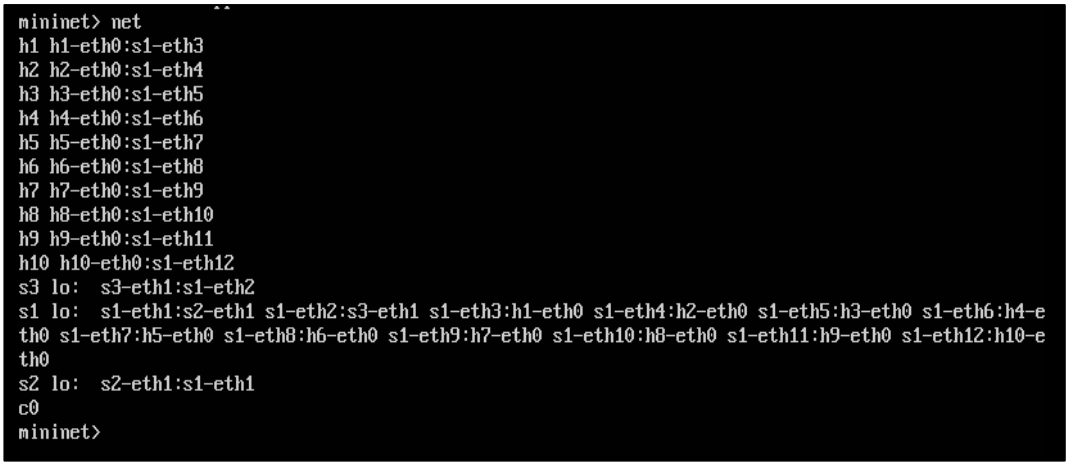


Figura 35. Resultado de la red y subredes topología SDN en Mininet
Fuente: Autores

3.2.6. FASE DE OPTIMIZACIÓN

Esta última fase tiene como objetivo detectar y resolver posibles errores que puedan producirse en la red, atendiendo los diferentes escenarios o situaciones reales de explotación.

Para lo cual se debe realizar una administración proactiva que valore y permita anticiparse a una serie de situaciones, con el objetivo de realizar acciones específicas para prevenirlas y corregir los errores que podrían llevar a una falla en el funcionamiento de la red propuesta, teniendo en cuenta, además, el crecimiento exponencial del empleo de las redes de datos mediante el uso de redes sociales y los contenidos multimedia, la explosión de la computación en la nube y el impacto del creciente uso de tecnología móvil.

De aquí que la valoración de la propuesta para minimizar los costos mientras que los ingresos empresariales se mantienen fijos y el despliegue de una solución basada en software, supone ventajas sobre las redes tradicionales y sus posibles migraciones a otras de mayor operatividad, por lo que en esta investigación se propone la topología SND como muestra la Figura 36.

Se presenta como una solución basada en software, una topología de red programable, como solución de bajo costo y de alta disponibilidad, convergiendo a aplicaciones y desacoplando la capa de aplicación del plano de datos que pueda ser gestionada y administrada de manera centralizada.

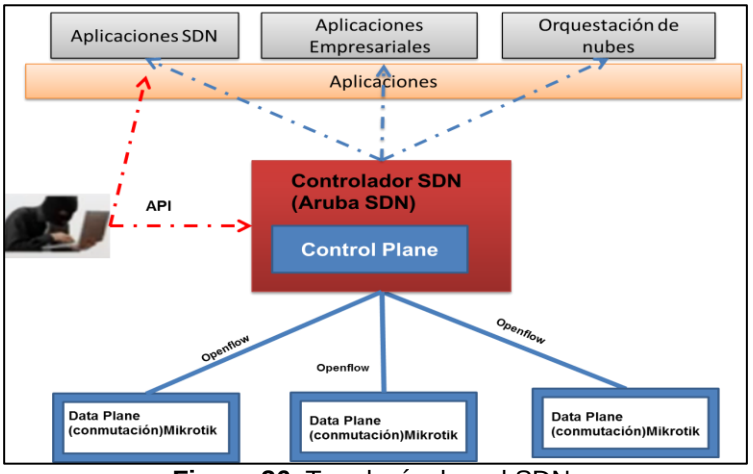


Figura 36. Topología de red SDN.
Fuente: Autores

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

Para la obtención de los resultados, se creó un entorno virtual para las pruebas que permitió la instalación de cada uno de los software y herramientas ya descritos en la metodología empleada, con sus respectivas configuraciones, que permitieran desplegar una red con similar estructura a la existente en la Universidad Técnica de Cotopaxi extensión La Maná. El principal resultado que se pretende demostrar que reutilizando el mismo equipamiento ya existente, es posible implementar una red de tipo SDN con los beneficios que esta conlleva (mayor control y flexibilidad, reducción de costos y tiempo para su implementación), resaltando fundamentalmente su rápida adaptación a las exigencias por las nuevas tecnologías emergentes. Los principales resultados obtenidos de la investigación lo constituyen:

- La identificación del controlador a utilizar en redes definidas por software (SDN).

Como ya fue descrito, existen varios controladores (Tabla 8) para redes SDN, en esta investigación fue seleccionado el HP Van SDN Controller, por sus características y facilidad de implementación.

Tabla 8. Relación de controladores SDN.

Controladores	Documentación	Soporte	Plataforma
HP Van Controller	Buena	OpenFlow v1.0-v1.4	Linux
OpenDayLight	Media	OpenFlow v1.0-v1.4	Linux, Mac Os, Windows
Ryu	Media	OpenFlow v1.0-v1.3	Linux
Beacon	Buena	OpenFlow v1.0	Linux, Mac Os, Windows
NOX	Media	OpenFlow v1.0	Linux
POX	Pobre	OpenFlow v1.0	Linux, Mac Os, Windows
Floodlight	Buena	OpenFlow v1.0-v1.4	Linux, Mac Os, Windows

Fuente: Autores

- **Identificación del protocolo a implementar.**

Si bien es conocido que existen varios protocolos (OpenFlow, OpFlex, NETCONF, MPLSTP, LISP, entre otros) que permiten la comunicación entre las herramientas actores (Controlador, Switch, Router) para la implementación de redes SDN como se muestra en la Tabla 9, en esta investigación se selecciona OpenFlow y se comprueba que es el más utilizado por compañías, proveedores de servicios y universidades como la de Stanford.

Tabla 9. Relación de protocolos empleados en redes SDN.

Protocolo	Funciones
OpenFlow	Comunicación entre los planos de control y plano de datos en arquitectura SDN.
OpFlex	Busca mantener la inteligencia de control en la infraestructura de la red
NETCONF	Permite la gestión de configuración de dispositivos de red.
MPLSTP	Es utilizado como una tecnología de capa de red en redes de transporte.
LISP	Resuelve los inconvenientes de escalabilidad de las tablas de enrutamiento de la Zona Libre de Internet.

Fuente: Autores

- **La Identificación de un Emulador.**

En la actualidad existen diversos emuladores (Mininet, GNS3, OMnet++, entre otros) que podrían aplicarse a investigaciones o simplemente servir para laboratorios de prueba, en este caso fue seleccionado Mininet, atendiendo al soporte, documentación existente, sus funcionalidades y compatibilidad con el escenario desplegado. La utilización de un emulador en la investigación tiene como principal objetivo el de facilitar pruebas en tiempo real para demostrar la factibilidad de la propuesta ante un número mayor de estaciones de trabajo, carga en la red y el funcionamiento de la arquitectura con el controlador.

Una vez implementado el escenario virtual se hicieron pruebas para comprobar el funcionamiento de la arquitectura desplegada sobre el laboratorio controlado.

- **Pruebas de envío de paquetes entre las estaciones.**

Se efectuaron el envío y recepción de paquetes a nivel de protocolo de mensajes de control de Internet ICMP entre 10 estaciones de red desplegadas mediante el Controlador SDN, basada en una topología segmentada en Mininet.

Se obtuvo el rendimiento del controlador SDN VAN CONTROLLER, el mismo que se encarga del proceso de enrutamiento y gestión de la red programada mediante el sistema Mininet, obteniendo como resultado 180 paquetes enviados, con 200 milisegundo de duración del envío, con una prioridad de flujo 5% y el tratamiento realizado por el controlador, un proceso de enrutado y gestión verdadera, como muestra la Tabla 10.

Tabla 10. Comunicación de red SDN.

Estado	Paquete	Duración	Prioridad de flujo	Nombre de la tabla	Selector	Tratamiento	Nombre de la Aplicación
Agregado	0	200	40000	0	ETH_ TYPE: lldp	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	0	200	40000	0	ETH_ TYPE: bdp	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	180	200	40000	0	ETH_ TYPE: arp	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	180	200	5000	0	ETH_ TYPE: ipv4	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	0	200	40000	0	ETH_ TYPE: lldp	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro

Agregado	0	220	40000	0	ETH_ TYPE: bdp	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	160	210	40000	0	ETH_ TYPE: arp	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	180	200	5000	0	ETH_ TYPE: ipv4	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	160	210	40000	0	ETH_ TYPE: arp	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro
Agregado	180	200	5000	0	ETH_ TYPE: ipv4	imm[SALIDA:CONTROLADOR], borrador:verdadero	*centro

Fuente: Autores

Mediante los resultados de la Tabla 10 se analizó la eficiencia del servicio de transmisión y recepción de paquetes de la red SND, obteniendo resultados muy favorables en relación a pruebas efectuadas en la red de datos tradicional sin despliegue de tecnología SDN y controladores.

- **Puertos para Dispositivo y nivel de latencia.**

En la Tabla 11 se registran los datos obtenidos del nivel de latencia, a través del ping establecido a todas las PC (10), mediante el sistema Mininet, desplegadas y gestionadas por el sistema VAN CONTROLLER desde las PC1 virtual hacia todas las demás PCs.

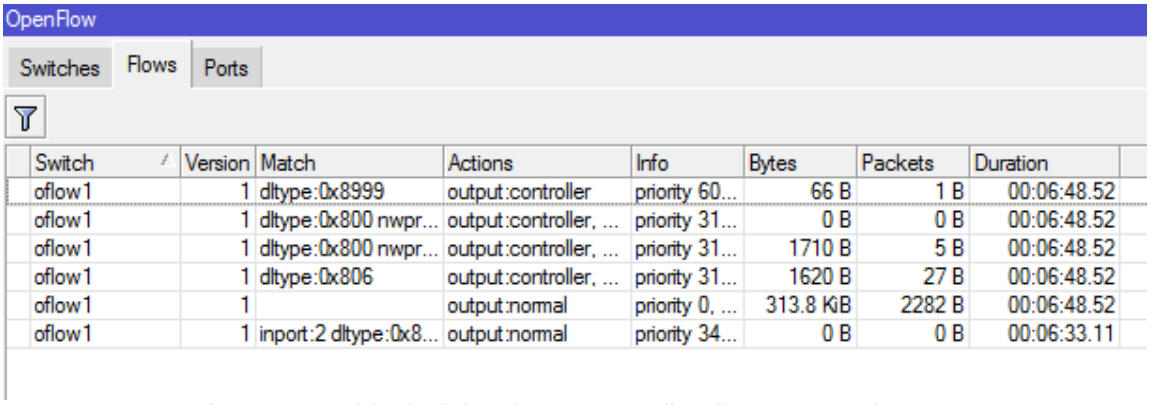
Tabla 11. Dispositivo y nivel de latencia.

ID de Puerto	Paquetes Recibidos	Paquetes Enviados	Bytes Recibidos	Bytes Enviados	Paquetes RX Descartados	Paquetes TX Caídos	Duración (SEG)
1	44	945	3456	131903	0	0	0
2	45	951	3498	132755	0	0	0
3	880	888	125328	127552	-1	-1	0

4	900	890	127174	129120	-1	-1	0
5	44	945	3456	131903	0	0	0
6	45	951	3498	132755	0	0	0
7	880	888	125328	127552	-1	-1	0
8	900	890	127174	129120	-1	-1	0
9	880	888	125328	127552	-1	-1	0
10	900	890	127174	129120	-1	-1	0

Fuente: Autores

Los resultados de la tabla de flujos en R1, R2 y R3 muestran que cada interruptor ha agregado una entrada de la tabla de flujo con una acción directa hacia el controlador SDN, se observa la cantidad de bytes enviados, paquetes procesados y la duración del paquete, ver figura 37.



Switch	Version	Match	Actions	Info	Bytes	Packets	Duration
oflow1	1	dltype:0x8999	output:controller	priority 60...	66 B	1 B	00:06:48.52
oflow1	1	dltype:0x800 nwpr...	output:controller, ...	priority 31...	0 B	0 B	00:06:48.52
oflow1	1	dltype:0x800 nwpr...	output:controller, ...	priority 31...	1710 B	5 B	00:06:48.52
oflow1	1	dltype:0x806	output:controller, ...	priority 31...	1620 B	27 B	00:06:48.52
oflow1	1		output:normal	priority 0, ...	313.8 KiB	2282 B	00:06:48.52
oflow1	1	inport:2 dltype:0x8...	output:normal	priority 34...	0 B	0 B	00:06:33.11

Figura 37. Tabla de flujos de router 1 Mikrotik con OpenFlow y SDN
Fuente: Autores

4.2. DISCUSIÓN

Diversos autores muestran resultados similares a los que se han obtenido en la presente investigación. Oviedo Bayas et al. (2021), desarrollan un proyecto enfocado a la implementación de una red SDN que permite brindar servicio de VoIP seguros. Para medir la latencia realizan pruebas a dos direcciones IP, obteniendo un resultado óptimo, dando a conocer que no hubo pérdidas de paquetes en el transcurso de la transmisión, con un retardo de 5ms en el paquete #56 y 0ms en los paquetes restantes.

Para la evaluación del rendimiento y el performance, estos autores utilizaron la visualización de las interfaces para el monitoreo de la red del Mikrotik, obteniendo una transmisión y recepción de velocidad de 891 kbps, con 53 paquetes enviados y 53 recibidos.

En la presente investigación se coincide con Facchini et al. (2021), quienes analizan el comportamiento de las redes SDN para compararlas con las tradicionales mediante técnicas de simulación, bajo distintas perspectivas y escenarios topológicos posibles. Obtuvieron el rendimiento del controlador SDN VAN CONTROLLER, el mismo que se encarga del proceso de enrutamiento y gestión de la red programada mediante el sistema Mininet, obteniendo resultados satisfactorios con respecto a la cantidad de paquetes enviados en relación a la duración del envío, con una prioridad de flujo 5%.

Castaño Castañeda y Valencia Henao (2016), crearon una red definida por software, con 4 switches y 12 host, además de un controlador. Al analizar la velocidad, comprobaron que el tiempo de envío de los paquetes en las redes SDN, es más acelerado que una red tradicional, debido a que los switches consultan sus tablas de flujo antes de hacer el forwarding del tráfico. Estos resultados coinciden con los obtenidos en la presente investigación, ya que se logra una red gestionada de manera centralizada. Así mismo, una prueba límite Mininet muestra un ancho de banda máximo de 30 Gbits/segundo condicionado al número de host virtualizados.

Coincidiendo con lo expuesto en la presente investigación, Moreira Cabrera (2018), concluye que Mininet es una herramienta muy potente orientada a demostrar de manera sencilla y educativa las principales características de OpenFlow y el modo en el que interactúan sus diferentes elementos de red, sobre todo el controlador.

A través de estas tecnologías, los autores de la presente investigación, desarrollaron en un entorno virtual, una red SDN para comprender cómo administrar y supervisar su funcionamiento con el HP VAN SDN CONTROLLER como controlador. Para corroborar la solución propuesta en la presente investigación y los resultados obtenidos se analizan los resultados presentados por Google presentados por MPA Publishing International Ltd. (2022) y Diarioti.com (2023).

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

1. El desarrollo de la presente investigación permitió mediante bases conceptuales conocer las ventajas y características que posee las redes SDN, ya que este nuevo paradigma de red posibilita programar la red permitiendo configurar los equipos y gestionarlos de manera centralizada, aunque siendo una tecnología bastante joven posee un gran potencial lo cual permitió presentar una solución adecuada para la red de la Universidad Técnica de Cotopaxi Extensión La Maná.
2. La simulación dentro de un ambiente controlado permitió determinar el funcionamiento de las herramientas utilizadas, mismas que son basadas en software libre, permitiendo evidenciar mejoras significativas en términos de rendimiento, escalabilidad, funcionalidad, administración y control de la red. Se elige a Mininet Ubuntu 20.04 como emulador de la red SDN, el controlador Aruba Van SDN este es el encargado de enrutar los paquetes de datos, el protocolo OpenFlow mismo que permite a los conmutadores de red a donde deben enviar los paquetes transmitidos, otras herramientas tecnológicas de virtualización y comunicación como son VMware Workstation y el router Mikrotik.
3. Analizando los resultados obtenidos de la simulación se concluye que existen mejoras significativas en el paradigma de red planteado como es la red definida por software (SDN), ya que se evidencia un mejor esquema de red, reducción en la cantidad de equipos utilizados, además se centraliza todos los dispositivos de red generando poca probabilidad de que existan inconsistencias en el flujo de datos dentro de la red.
4. Se diseña la propuesta de red SDN, se realizan las pruebas y se describen los elementos teóricos que demuestran su contribución a la mejora de la administración y la calidad de los servicios de la red de la Universidad de Cotopaxi Extensión La Maná.

5.2. RECOMENDACIONES

1. Continuar la investigación y el análisis del desempeño de redes SDN para hacer uso de todas sus funcionalidades y ventajas sobre las redes tradicionales, además considerar el equipamiento existente que sea compatible con la solución propuesta para evitar la compra innecesaria de equipamiento.
2. Capacitar al personal en los conceptos y tecnologías asociadas con SDN. Es importante asegurarse de que el personal tenga el conocimiento y las habilidades necesarias para diseñar, implementar y gestionar una red SDN de manera efectiva.
3. Implementar medidas de seguridad adecuadas para proteger la red SDN y los datos que transitan por ella. Esto incluye la implementación de autenticación, autorización, encriptación y otras medidas de seguridad necesarias para proteger la integridad y confidencialidad de la red y los datos.
4. Realizar pruebas y validación exhaustivas antes de la implementación completa de SDN. Esto incluye la realización de pruebas de concepto, pruebas de interoperabilidad y pruebas de seguridad para asegurarse de que la red SDN funcione correctamente y cumpla con los requisitos establecidos.

BIBLIOGRAFÍA

Aguilar Peña, D. F. (2021). *Factibilidad de una red metro Ethernet basada en la metodología PPDIOO aplicada a PYMES* Universidad Técnica de Machala]. Machala, Ecuador.

Ahuja, N., Singal, G., Mukhopadhyay, D. y Kumar, N. (2021). Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*, *187*, 103-108. https://doi.org/https://doi.org/10.1016/j.jnca.2021.103108

Al Mahdawi, R. S. y Salih, H. M. (2021). Optimization of open flow controller placement in software defined networks. *International Journal of Electrical and Computer Engineering*, *11*(4), 3145-3153. https://doi.org/10.11591/ijece.v11i4.pp3145-3153

Alvarado Cadena, R. (2013). *Documentación, implementación y elaboración de guías de laboratorio sobre protocolos de enrutamiento en la Red: RIP, IS-IS, OSPF Y BGP; basados en un software de simulación*. Repositorio Institucional UPB.

Badotra, S. y Panda, S. N. (2020). Software-Defined Networking: A Novel Approach to Networks. Handbook of Computer Networks and Cyber Security,

Barona, L., Valdivieso, L. y Guamán, D. (2014). *Una Nueva alternativa a mininet: Emulación de una red definida por software usando VNX* IX Congreso de Ciencia y Tecnología ESPE, España.

Barrera Pérez, M. Á., Serrato Losada, N. Y., Rojas Sánchez, E. y Gaona, G. M. (2018). State of the art in software defined networking (SDN). *Visión Electrónica*, *13*(1), 178-194.

Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., . . . Parulkar, G. (2014). ONOS: Towards an open, distributed SDN OS. ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking,

Bernal, I. y Mejía, D. (2016). Las Redes Definidas por Software y los Desarrollos Sobre Esta Temática en la Escuela Politécnica Nacional. *Revista Politécnica*, *37*(1), 1-11.

Braun, W. y Menth, M. (2018). Software-defined networking using OpenFlow: protocols, applications and architectural design choices. *Future Internet*, *6*, 302-336.

Citrix.es. (2019). *SDN: Software Defined Networks y el papel de los servicios de red en la entrega de aplicaciones*. Citrix.es. Retrieved 12-12-2022 from https://www.citrix.com/content/dam/citrix/enus/documents/products-solutions/sdn-102-software-defined-networks-and-the-role-of-application%2ddelivery-network-services-es.pdf

Contreras Pardo, C. A. (2014). Implementación de un openflow controller para el manejo de openflow switches. In *Reposito institucional*: Pontificia Universidad Javeriana.

Córdoba López, S. (2018). *Estudio de redes SDN mediante Mininet y MiniEdit* Universitat Politécnica de Valencia].

Chafloque Mejia, J. D. (2018). *Propuesta de diseño de una red de datos de área local bajo la arquitectura de redes definidas por software para la Red Telemática de la Universidad Nacional Mayor de San Marcos* Universidad Nacional Mayor de San Marcos]. Perú. http://cybertesis.unmsm.edu.pe/handle/20.500.12672/10017

Deloitte.com. (2019). *Predicciones de Tecnologíia, Medios de Comunicaciónn y Telecomunicaciones*. Deloitte.com. Retrieved 12-12-2022 from http://www2.deloitte.com/content/dam/Deloitte/es/Documents/tecnologia-media-telecomunicaciones/DeloitteESTMTPrediccioes-2019.pdf

Diarioti.com. (2023). *Google explica las ventajas de haber incorporado SDN y NFV en su propia nube*. Retrieved 11-2-2023 from https://diarioti.com/google-explica-las-ventajas-de-haber-incorporado-sdn-y-nfv-en-su-propia-nube/82876

Dixon, J. (2016). *Software Defined Networking: What it is and what you should know*. Infosecwriters. http://www.infosecwriters.com/Papers/JDixon_SDN.pdf

Doria, A., Hadi, J., Hass, R., Khosravi, H., Wang, W., Dong, L., . . . Halpern, J. (2019). *Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810*. Internet Research Task Force.

Duarte, G. y Lobo, R. (2015). Emulación de escenarios virtuales, en una SDWLAN (Software Defined Wireless Local Area Network), de un campus universitario. *Ingenieria al Dia*, *1*(2), 69-85.

Dueñas Santos, C. L., Marín Muro, Y. A. y Cruz Enriquez, H. (2017). *Analysis of the performance of an SDN network over a traditional network in LAN environments* Convención Científica Internacional Ciencia, Tecnología y Sociedad. Perspectivas y retos, Universidad Central “Marta Abreu” de Las Villas, Santa Clara, Cuba.

Enns, R., Bjorklund, M., Schoenwaelder, J. y Bierman, A. (2018). *Network Configuration Protocol (NETCONF). RFC 6241*. Internet Research Task Force.

Erickson, D. (2013). The beacon openflow controller. Second ACM SIGCOMM workshop on Hot topics in software defined networking, New York.

Flores, A. (2022). *VMware vs. VirtualBox: ¡descubre cuál es el mejor virtualizador de sistemas operativos*. Retrieved 1-8-2022 from https://www.crehana.com/blog/desarrollo-web/vmware-vs-virtualbox/

García, R. (2022). *¿Qué es VMware y para qué sirve?* Retrieved 5-6-2022 from https://blog.mdcloud.es/que-es-vmware/

Garrich, M., Romero-Gazquez, J. L., Moreno-Muro, F. J., Hernandez-Bastida, M., Delgado, M. V. B., Bravalheri, A., . . . Marino, P. P. (2020). IT and Multi-layer Online Resource Allocation and Offline Planning in Metropolitan Networks. *Journal of Lightwave Technology*, *38*(12), 3190-3199. https://doi.org/https://doi.org/10.1109/JLT.2020.2990066

Gonzalez, C., Flauzac, O. y Nolot, F. (2018). Evolución y Contribución para el Internet de las Cosas por las emergentes Redes Definidas por Software. *Memorias de Congresos UTP*, *1*(1), 28-33. https://revistas.utp.ac.pa/index.php/memoutp/article/view/1842

Herrera Vaca, A. J. (2020). *Diseño y optimización de una red GPON a través de una red SDN para la Facultad Técnica para el Desarrollo.* Universidad Católica de

Santiago de Guayaquil]. Guayaquil, Ecuador. http://repositorio.ucsg.edu.ec/bitstream/3317/15577/1/T-UCSG-PRE-TEC-ITEL-383.pdf

Hewllet Packard Enterprise. (2022). *OpenDaylight "Powered by" recognises technical excellence*. TechNative. Retrieved 16-12-2022 from https://technative.io/opendaylight-powered-programme/

Huang, D., Chowdhary, A. y Pisharody, S. (2018). Software-Defined Networking and Security: From Theory to Practice. In.

IONOS. (2020). *Hyper-V: el programa de virtualización de Microsoft*. Retrieved 11-5-2022 from https://www.ionos.es/digitalguide/servidores/know-how/que-es-hyper-v/

Jammal, M., Singh, T., Shami, A., Asal, R. y Li, Y. (2017). Software defined networking: State of the art and research challenges. *Comput. Netw*, *72*, 74-98.

Javanmardi, S., Shojafar, M., Mohammadi, R., Nazari, A., Persico, V. y Pescapè, A. (2021). FUPE: A security driven task scheduling approach for SDN-based IoT-Fog networks. *Journal of Information Security and Applications*, *60*(102853), 1-17. https://doi.org/https://doi.org/10.1016/j.jisa.2021.102853

Jha, P. y Tech, B. (2017). *End-to-End Quality-of-Service in Software Defined Networking*. McGraw Hill.

Jiménez Contreras, I. (2018). *Sistemas Informáticos y Redes locales*. Garceta.

Jiménez, J. (2019). *Una vulnerabilidad crítica de Cisco sin solucionar permite a un atacante tener el control total*. https://www.redeszone.net/2019/01/21/vulnerabilidad-critica-cisco-controlar-sistema/

Jiménez Morales, P. J. (2018). *Desarrollo de prácticas de laboratorio de SDN en Mininet* Unuversidad Central "Marta Abreu" de Las Villas]. Santa Clara, Cuba. https://dspace.uclv.edu.cu/bitstream/handle/123456789/10059/Pedro%20J%20Jim%C3%A9nez.pdf?isAllowed=n&sequence=1

Juniper Networks. (2023). *Descripción general del BGP*. Juniper Networks, Inc. Retrieved 10-6-2023 from https://www.juniper.net/documentation/mx/es/software/junos/bgp/topics/topic-map/bgp-overview.html

Karakus, M. y Guler, E. (2020). *RoutingChain: A Proof-of-Concept Model for a Blockchain-Enabled QoS-Based Inter-AS Routing in SDN* IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), New York.

Kreutz, D., Ramos, F. M., Verissimo, P., Rothenberg, C. E., Azodolmolky, S. y Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proc. IEEE*, *103*(1), 14-76.

Kumar, M. B. D. y Deepa, B. (2017). Computer Networking. *International Journal of Trend in Research and Development*, *2*(5), 126-130.

Kurth, L. (2013). *XenServer.org and the Xen Project*. Xen Project.

Kurth, L. (2015). *Proxmox... termina la introducción*. GUTL.

KVM.org. (2019). *KVM*. Retrieved 10-10-2022 from https://www.linux-kvm.org/page/Main_Page

Liang, X. y Qiu, X. (2017). *A software defined security architecture for SDN-based 5G network* 5th International Conference on Network Infrastructure and Digital Content, IEEE IC-NIDC 2016,

Mahabir, H. (2023). *SDN vs NFV: por qué son los componentes clave para modernizar las redes*. Adapt IT Telecoms. Retrieved 10-2-2023 from https://telecoms.adaptit.tech/es/blog/sdn-vs-nfv/

Marín Muro, Y. A. y Alvarez Paliza, F. (2016). *Plataforma de pruebas para evaluar el desempeño de las redes definidas por software basadas en el protocolo Openflow*. ReseartGate. Retrieved 11-3-2023 from https://www.researchgate.net/publication/312187600_PLATAFORMA_DE_PRUEBAS_PARA_EVALUAR_EL_DESEMPENO_DE_LAS_REDES_DEFINIDAS_POR_SOFTWARE_BASADAS_EN_EL_PROTOCOLO_OPENFLOW

Millán, R. (2014). SDN: el futuro de las redes inteligentes. *Conectrónica*(179), 24-27.

Mocha Guacho, G. y Celleri Pacheco, J. (2020). Análisis Comparativo de Protocolos de Comunicación para Redes definidas por Software. *Hamut´ay*, *7*(3), 39-50. https://doi.org/http://dx.doi.org/10.21503/hamu.v7i3.2190

N.O.X. Repo. (2019a). *The NOX Controller. Contribute to noxrepo/nox development by creating an account on GitHub*. N.O.X. Repo.

N.O.X. Repo. (2019b). *The POX network software platform. Contribute to noxrepo/pox development by creating an account on GitHub*. N.O.X. Repo.

Núñez, R. A. V. (2015). *Red definida por software (SDN) en base a una infraestructura de software de libre distribución* Universidad Técnica de Ambato].

O. N. Fundation. (2012). Software-defined networking: The new norm for networks. *ONF White Paper*, *2*, 2-6.

Open Network Operating System. (2019). *Contribute to opennetworkinglab/onos development by creating an account on GitHub*. Open Networking Foundation.

Open Networking Foundation. (2016). *SDN Architecture. Issue 1.1* (O. N. Foundation, Ed.). Open Networking Foundation.

Open Networking Foundation. (2017). *Migration Use Cases and Methods*. Migration Working Group.

OpenDaylight.org. (2019). *Home. OpenDaylight*. OpenDaylight.org. Retrieved 4-10-2022 from https://www.opendaylight.org/

Osrg/Ryu.org. (2019). *Ryu component-based software defined networking framework*. Osrg/Ryu.org.

Oviedo Bayas, B. W., Zhuma Mera, E. R., Bowen Calero, G. K. y Patiño Maisanche, B. S. (2021). Implementación de una red definida por software que permita brindar servicio de VoIP seguros. *Revista Universidad y Sociedad*, *13*(2), 389-396.

Paliwal, M., Shrimankar, D. y Tembhurne, O. (2018). Controllers in SDN: A review report. *IEEE Access*, *6*, 36256-36270. https://doi.org/https://doi.org/10.1109/ACCESS.2018.2846236

Pereira, G. y Gamess, E. (2017). Lineamientos para el Despliegue de Redes SDN/OpenFlow. *Revista Venezolana de Computación*, *4*(2), 21-33.

https://docplayer.es/72631676-Lineamientos-para-el-despliegue-de-redes-sdn-openflow.html

Pérez, G. C. y Marín, M. F., . (2021). Redes definidas por software: Solución para servicios portadores del Ecuador. *Investigation Research Review*, *6*, 41-63. https://doi.org/https://doi.org/10.31095/investigatio.2015.6.2

Pérez Tardío, E. R., Cruz Enríquez, H., Pérez Peláez, E. y Marín Muro, Y. A. (2019). *Evaluación de desempeño de las redes SDN mediante la aplicación de mecanismos de calidad de servicio definidos para redes IP* Universidad Central "Marta Abreu" de Las Villas]. Santa Clara, Las Villas. https://www.researchgate.net/profile/Yanko-Antonio-Muro/publication/336441596Evaluacion_de_desempeno_de_las_redes_SDN_mediante_la_aplicacion_de_mecanismos_de_calidad_de_servicio_definidos_para_redes_IP/links/5daf45de92851c577eb99e1a/Evaluacion-de-desempeno-de-las-redes-SDN-mediante-la-aplicacion-de-mecanismos-de-calidad-de-servicio-definidos-para-redes-IP.pdf

Pfaff, B. y Davie, B. (2018). *The Open vSwitch Database Management Protocol. RFC 7047*. Internet Research Task Force.

Prakruthi, S. y Patil, U. K. (2016). *Building SDN framework using OpenDaylight Controller*. McGraw Hill, New York.

Pretz, K. (2016). *Software already defines our lives but the impact of SDN will go beyond networking alone*. The IEEE News Source. http://theinstitute.ieee.org/technology-focus/technology-topic/software-already-defines-our-lives

Priano, D. A. (2021). *Análisis de protocolos de enrutamiento en redes definidas por software (Software Defined Networks)*. Universidad Nacional de La Plata.

PromonLogicalis. (2013). *Como o novo universo trazido pelas redes definidas por software ira impactar os negocios*. Retrieved 1-2-2022 from http://www.br.promonlogicalis.com/globalassets/latin-america/advisor-sdnwebsafe.pdf

Quijada, B. (2021). Tres décadas de investigación española en comunicación: hacia la mayoría de edad. *Comunicar*, *41*(XXI), 15-24. https://doi.org/http://dx.doi.org/10.3916/C41-2013-01

Ramírez Giraldo, M. y López Echever, A. M. (2018). Redes de datos definidas po rsoftware-SDN, arquitectura, componentes y funcionamiento. *Journal de Ciencia e Ingeniería*, *10*(1), 55-61.

Ramos Suavita, D. J. (2021). *Análisis de vulnerabilidades a nivel de seguridad en redes SDN para los planos de control y plano de datos* Universidad Militar Nueva Granada]. Bogotá, Colombia.

Ríos, R. A. (2016). Conceptualización de SDN y NFV. *Maskay*, *6*(1), 29-34. http://scielo.senescyt.gob.ec/scielo.php?pid=S1390-67122016000100029&script=sci_arttext

Roncero, R. (2014). *Software defined networking*. Barcelona. Universidad Politécnica de Cataluña.

Ruipérez Cuesta, J. (2021). *Seguridad en Redes definidas por software (SDN)* Universitat Politècnica de València]. España. http://hdl.handle.net/10251/165154

Salazar Chacón, G. D. (2021). *Hybrid Networking SDN y SD-WAN: Interoperabilidad de arquitecturas de redes tradicionales y redes definidas por software en la era de la digitalización* Universidad Nacional de La Plata]. Argentina.

Sayans Cobo, P. (2018). *Despliegue de laboratorios virtualizados de SDN utilizando Open vSwitch*. Linux.org.

Singh, S. y Jha, R. K. (2017). A survey on software defined networking: architecture for next generation network. *J. Netw. Syst. Manag.*, *25*(2), 321-374.

Spera, C. (2013). Software Defined Network: el futuro de las arquitecturas de red. *Logicalis Now*, 42-45. http://www.la.logicalis.com/globalassets/latin-america/logicalisnow/

ANEXOS

ANEXO 1. ENTREVISTA

Universidad Técnica de Cotopaxi Extensión La Maná

Maestría en Tecnología de la Información Mención Redes y Sistemas Distribuidos

Información del Entrevista

Fecha: _____ Hora: _____

Nombre entrevistado: _____

Cargo: _____

Email: _____

Teléfono: _____

Datos del entrevistador

Nombre: _____

Email: _____

1. ¿La dependencia de TICs de la UTC cuenta con políticas establecidas de TI?

2. ¿Cuenta con un inventario tecnológico de los equipos que maneja el departamento?

3. ¿Tiene el control total de gestión de cambios de su red de datos?

4. ¿Ha estado en riesgo de perder información sensible a causa de fallos físicos en sus equipos?

4. ¿Puede monitorear su red de datos proactivamente?

5. ¿La red de datos de la institución está preparada para el crecimiento de aquí en 5 años?
6. ¿El Departamento de TI se encarga de la gestión de hardware de red cuando se lo requiere o lo envían a contratistas?
7. ¿La Universidad dispone de algún proyecto de innovación para la mejora tecnológica incluyendo un presupuesto destinado para su financiación?
8. ¿Cuáles son los servicios y aplicaciones que utilizan en la Universidad? Web, correo, entre otros.

ANEXO 2. AUTORIZACIÓN REALIZAR LA INVESTIGACIÓN.

UNIVERSIDAD TÉCNICA DE COTOPAXI

DIRECCIÓN GENERAL LA MANÁ

LMD-05-2021
La Maná, 03 de septiembre del 2021

Srs.
Ing. Jonathan Alexander Moran Macias
Ing. Karina Mendoza Zambrano
ESTUDIANTES MAESTRANTES DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ.

De mis consideraciones,

Por medio del presente me permito expresarle un cordial saludo y a la vez en atención al oficio s/n de 03 de septiembre del 2021, donde solicitan realizar el estudio comparativo con el tema: **PROPUESTA DE UNA SOLUCIÓN VIRTUALIZADA UTILIZANDO SOFTWARE LIBRE COMO INFRAESTRUCTURA EN LA RED DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ**, por lo expuesto autorizo se puedan realizar esta investigación sobre la red informática que contamos actualmente ,ya que a futuro permitirá mejorar la arquitectura de la red en nuestra institución.

Particular que comunico para fines pertinentes

Atentamente,

"POR LA VINCULACIÓN DE LA UNIVERSIDAD CON EL PUEBLO"

UNIVERSIDAD TÉCNICA DE COTOPAXI DIRECCIÓN GENERAL LA MANÁ

Ing. Gloria Paznaño Cano MBA.
DIRECTORA EXTENSIÓN LA MANÁ

GPC/jvt

ANEXO 3: PROPUESTA

ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ

PROPUESTA DE UNA SOLUCIÓN VIRTUALIZADA UTILIZANDO SOFTWARE LIBRE COMO INFRAESTRUCTURA EN LA RED DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ.

Septiembre 2023

PRESENTACIÓN CON UN DISEÑO

Plan de acción para optimizar los recursos digitales desde la migración de la red de la Universidad Técnica de Cotopaxi Extensión la Maná a una red de avanzada de tipo SDN.

1. DESCRIPCIÓN DE LA INSTITUCIÓN

La Universidad Técnica de Cotopaxi (UTC) está ubicada en el barrio El Ejido, en la parroquia Eloy Alfaro, perteneciente al cantón Latacunga de la provincia de Cotopaxi.

Hace 28 años inició el sueño de tener una institución académica de primer nivel en la provincia, varios años de lucha, trabajo y sacrificio, debieron pasar para que se constituya la extensión de la Universidad Técnica del Norte en 1992.

El sueño se vio conquistado el 24 de enero de 1995 cuando nace la Universidad Técnica de Cotopaxi como una institución con autonomía.

A lo largo de estos 23 años la institución ha levantado una lucha incansable por la igualdad social, por la formación de profesionales con un sentido humanista, por la gratuidad de la educación y el libre acceso de todos los jóvenes sin importar su estrato social a formarse como profesionales.

La universidad tiene su planta matriz ubicada en San Felipe, en esta funcionan las facultades de Ciencias Administrativas, Ciencias Humanas, y Ciencias de la Ingeniería y Aplicadas. En el campus Salache labora el Centro de Experimentación Académica Salache (Ceasa) en el cual se desarrolla la Facultad de Ciencias Agropecuarias y Recursos Naturales.

La UTC cuenta con su extensión en el cantón La Maná, la cual fue acreditada como una de las mejores del país en septiembre 2015.En la actualidad existe un

aproximado de 1.500 estudiantes matriculados de primer ciclo en adelante y 1.080 alumnos registrados en Nivelación. Laboran 350 docentes, 182 empleados entre funcionarios regidos por la Ley Orgánica de Servicio Civil y Carrera Administrativa (Losca) y servidores bajo el Código de Trabajo.

MISIÓN

La UTC forma profesionales humanistas y de calidad, capaces de generar conocimiento científico a través de la investigación y vinculación, para que contribuyan a la transformación social, tecnológica y económica del país.

VISIÓN

La UTC será una universidad innovadora, científica y eficiente, comprometida con la calidad y pertinencia para alcanzar una sociedad equitativa, inclusiva y colaborativa.

2. SIGNIFICADO DE LA RED DE DATOS PARA LA COMUNIDAD UNIVERSITARIA

La posibilidad de la conectividad y el servicio de internet, sin duda permiten el crecimiento de instituciones educativas, mostrando al mundo sus principales potencialidades y resultados, de igual forma contribuye a captar nuevos estudiantes, profesores e investigadores de las distintas ramas del saber.

Universidad Técnica De Cotopaxi Extensión La Maná, actualmente cuenta con la tecnología para el sustento de los procesos universitarios según se describe a continuación:

Nº	Medio	Descripción
1	280 Computadoras	De prestaciones medias, con procesadores I3, I7,…, con memorias RAM entre 4 y 8 GB, con capacidades de almacenamiento entre 500 y 1024 GB.

2	5 Servidores	Prestación de servicios de Web Hosting, Videoconferencias, gestión de la red, desde el Campus Universitario
3	2 Router	Interconectividad entre los escenarios
4	24 Dispositivos Wi-fi	Extensión de la red a dispositivos móviles, Tablet, Laptop, etc.
5	20 Switch	L2 o L3 para la gestión de la interconexión de departamentos y/o plantas.

Los medios antes descritos son utilizados en función de 6 laboratorios y su proceso docente, el resto es empleada en la investigación y gestión administrativa.

Actualmente se desarrollan actividades de tipo académica y se prestan servicios que favorecen el desarrollo de las actividades de las áreas o departamentos.

3. BENEFICIOS DE LA MIGRACIÓN A UNA RED DE MAYORES PRESTACIONES

En la actualidad los requerimientos de las herramientas y servicio de internet, exigen de un personal más preparado y actualizado con las tendencias del momento, de aquí que los departamentos encargados de la gestión de la Tecnologías de la Información son los responsables de verificar que las distintas herramientas digitales, ya sean apps o software, funcionen correctamente y a favor de las actividades de la universidad.

En tal sentido y valorando el crecimiento sostenido de las redes, sus tecnologías, la necesidad de una mayor seguridad y las demandas por parte de los usuarios, se propone la implementación de las herramientas adecuadas según necesidades y evaluando a su vez el costo de su implementación.

Es por ello que se valora la migración a una red de tipo SDN (Redes Definidas por Software), la cual es un enfoque arquitectónico de la red, que permite ser controlada de manera inteligente y central, teniendo como sus principales funcionalidades la flexibilidad y escalabilidad, aspecto éste contrario a las redes tradicionales y que

implican mayores inversiones en sus mantenimientos y no existe un software adecuado para una gestión centralizada.

4. MEJORAS EN LA SEGURIDAD Y APLICACIÓN DE POLÍTICAS

Las redes SDN OpenFlow permiten implementar mecanismos de seguridad y aplicación de políticas mediante la incorporación de módulos de software al controlador de la red. A continuación, se describen algunos ejemplos de casos de usos de estos mecanismos.

Seguridad: SDN posee cualidades importantes que facilitan la implementación de mecanismos de seguridad:

1. visión global del estado de la red,
2. inteligencia de control centralizada
3. APIs abiertas para programar el comportamiento de la red.

Con estas facilidades SDN se convierte en el punto central para desarrollar aplicaciones de seguridad inteligentes incluyendo filtrado de paquetes, control de acceso, detección de intrusos y gestión de SLAs.

Aplicación de Políticas: Una gestión de políticas adecuada es de suma importancia en escenarios de redes corporativas y SDN puede ser utilizada para hacer cumplir políticas de red mediante la programación, el monitoreo y la entonación del desempeño de la red. El caso de uso de aplicación de políticas consiste en hacer cumplir políticas de red a usuarios, aplicaciones o dispositivos, puede definir políticas de uso de los servicios de la red como la tasa pico de ancho de banda de un enlace permitida a una aplicación o la hora del día en la cual se pueden acceder a determinadas VLANs o servicios.

5. PLAN DE ACCIÓN PARA LA MIGRACIÓN DE LA RED A SDN

Objetivo

Lograr la implementación de una red con mayores funcionalidades que permita una gestión centralizada.

Etapas	Objetivo	Acciones	Duración	Responsable
4. Evaluación de la tecnología existente	Determinar la tecnología existente que pueda utilizarse y que no implique nuevas inversiones.	1. Realizar un levantamiento de toda la infraestructura de red existente	1 semana	Analista de Comunicaciones y Arquitectura
		2. Revisión de la compatibilidad y reutilización para la propuesta	2 semanas	
5. Disposición de los componentes necesarios para la migración	Determinar todos los componentes de la red que puedan ser utilizados, aplicando los resultados de la investigación	3. Determinar los dispositivos a utilizar para la migración (Router, servidores o computadoras)	2 semanas	
		4. Revisión de la propuesta de investigación para la utilización del software propuesto	1 semanas	
		5. Preparar el software necesario para la migración (Controlador SDN, actualización de software en dispositivos o funcionalidad OpenFlow)	2 semanas	
6. Implementación de la migración a la red SDN	Implementar todos los componentes necesarios para el funcionamiento la red SDN	6. Instalación de los controladores	1 semana	
		7. Implementación del protocolo OpenFlow en los dispositivos de la red	1 semana	
7. Evaluación del desempeño de la propuesta	Evaluar el correcto funcionamiento de la propuesta implementada	8. Evaluar el correcto funcionamiento de la red mediante herramientas OpenFlow	1 semana	
		9. Determinar y reparar posibles fallas mediante las herramientas seleccionadas	2 semanas	

6. HERRAMIENTAS PARA REDES SDN

6.1. MONITOREO DE REDES SDN

Una arquitectura de red SDN puede utilizar herramientas de monitoreo de red propietarias tradicionales, como Netflow de Cisco, sFlow de InMon, o JFlow de Juniper Networks, o utilizar herramientas de monitoreo especializadas con poco overhead y alta precisión.

A continuación, se describen algunas arquitecturas de monitoreo disponibles:

PayLess: es un framework de monitoreo basado en consultas para redes SDN que provee un API RESTful flexible para recopilar estadísticas a diferentes niveles de agregación, como flujos, paquetes y puertos. Ejecuta la recopilación de información con alta precisión en tiempo real sin incurrir en un alto overhead de red. Utiliza un algoritmo de planificación adaptativo que permite alcanzar el mismo nivel de precisión del estándar OpenFlow sin tener que consultar continuamente a los switches.

OpenTM: es una arquitectura de monitoreo que lleva la traza de los flujos activos en una red OpenFlow. Adicionalmente obtiene la información de enrutamiento de la aplicación de enrutamiento del controlador y sondea periódicamente los contadores de cantidad de bytes y número de paquetes de los flujos activos en los switches a lo largo del camino de datos. Para reducir la sobrecarga en la red se pueden sondear de maneara aleatoria un subconjunto de los switches seleccionados cuidadosamente para no afectar la precisión de las estadísticas recopiladas por la herramienta.

FlowSense: es una arquitectura de monitoreo que permite estimar el desempeño de una red OpenFlow a un bajo costo. Utiliza un método pasivo que captura y analiza el intercambio de los mensajes de control entre los switches y el controlador de una red

6.2. HERRAMIENTAS PARA LA GESTIÓN MEDIANTE EL PROTOCOLO OPENFLOW

Algunas de las herramientas utilizadas para la gestión mediante el protocolo OpenFlow son las siguientes: YANG, NETCONF y OF-CONFIG.

Herramienta	Descripción
YANG	Proporciona los medios para definir el contenido transportado a través de NETCONF, tanto para datos como para operaciones. Juntos, ayudan a los usuarios a crear aplicaciones de administración de redes que satisfagan las necesidades de los operadores de redes
NETCONF	El Protocolo de configuración de red, mejor conocido como NETCONF, brinda acceso a las capacidades nativas de un dispositivo dentro de una red, define métodos para manipular su base de datos de configuración, recupera datos operativos e invoca operaciones específicas
OF-CONFIG	Aborda los problemas centrales de Operaciones, Administración y Gestión con el estándar OpenFlow

6.3. HERRAMIENTAS DE DEPURACIÓN OPENFLOW

Las herramientas de depuración son de suma importancia al momento de hacer implementaciones SDN ya que nos permiten efectuar validaciones del comportamiento de la red en desarrollo con respecto a los resultados esperados en el estudio y ayudan a identificar bugs o errores en la programación de las aplicaciones, a continuación, se enumeran algunas de las herramientas de depuración OpenFlow disponibles:

Herramienta	Descripción
NICE (No bugs In Controller Execution)	Herramienta de pruebas automatizada utilizada para ayudar a descubrir bugs en programas OpenFlow a través del chequeo del modelo para explorar el espacio del estado del sistema completo: controlador, switches y hosts a través de la ejecución simbólica de manejadores de eventos
oftrace	Herramienta de trazas y análisis OpenFlow que toma como entrada un archivo con formato libpcap generado por tcpdump, wireshark u otro programa de análisis de red y genera como salidas estadísticas útiles sobre la sesión OpenFlow
Anteater	Herramienta de depuración OpenFlow que chequea invariantes de red en el plano de datos, así como aspectos de conectividad o consistencia. La herramienta es agnóstica de los protocolos y puede capturar errores

	adicionales como fallas en el firmware del switch o inconsistencias con la comunicación del plano de control
VeriFlow	Herramienta de verificación OpenFlow que reside entre el controlador y los switches siendo capaz de detener reglas erróneas que pudieran provocar un comportamiento anómalo antes de alcanzar la red
OfRewind	Herramienta de depuración que permite registrar eventos de red asociados a los planos de control y datos para reproducirlos en una etapa posterior y poder identificar y resolver los eventos que causaron alguna anomalía en la red
ndb	Herramienta que implementa puntos de chequeo y trazas hacia atrás de paquetes en ambientes SDN mostrando la secuencia de acciones de reenvío por las que atraviesa un paquete
STS	Simulador de resolución de problemas de redes SDN escrito en Python y dependiente de POX. La herramienta simula los dispositivos de la red permitiendo generar casos de pruebas y examinar el estado de la red de manera interactiva para encontrar las entradas responsables de generar bugs

- **CONCLUSIONES**:

1. La implementación de una red definida por software (SDN, por sus siglas en inglés) ofrece numerosos beneficios, como la flexibilidad, escalabilidad y automatización en la gestión de redes, lo que permite a las organizaciones adaptarse rápidamente a las cambiantes necesidades del negocio.
2. La adopción de SDN implica una transformación en la forma en que se diseñan, implementan y gestionan las redes, lo que requiere una comprensión profunda de los conceptos y tecnologías asociadas, así como la capacitación y actualización constante del personal.
3. La planificación cuidadosa y la definición clara de los objetivos son esenciales antes de implementar una red SDN. Es importante evaluar la infraestructura existente, identificar las áreas problemáticas y determinar los casos de uso más apropiados para implementar SDN.
4. La elección de la plataforma SDN adecuada es fundamental para el éxito de la implementación. Es importante evaluar las diferentes opciones disponibles en el mercado en términos de capacidad, escalabilidad, interoperabilidad y soporte de estándares abiertos.
5. La seguridad es un aspecto crítico en la implementación de SDN. Es necesario asegurarse de que se implementen medidas de seguridad adecuadas, como

autenticación, autorización y encriptación, para proteger la red SDN y los datos que transitan por ella.

- **RECOMENDACIONES:**

1. Realizar una evaluación exhaustiva de los requisitos y objetivos de la red antes de implementar SDN. Esto incluye identificar los casos de uso más apropiados, evaluar la infraestructura existente y definir claramente los objetivos de la implementación.
2. Capacitar al personal en los conceptos y tecnologías asociadas con SDN. Es importante asegurarse de que el personal tenga el conocimiento y las habilidades necesarias para diseñar, implementar y gestionar una red SDN de manera efectiva.
3. Realizar pruebas y validación exhaustivas antes de la implementación completa de SDN. Esto incluye la realización de pruebas de concepto, pruebas de interoperabilidad y pruebas de seguridad para asegurarse de que la red SDN funcione correctamente y cumpla con los requisitos establecidos.
4. Implementar medidas de seguridad adecuadas para proteger la red SDN y los datos que transitan por ella. Esto incluye la implementación de autenticación, autorización, encriptación y otras medidas de seguridad necesarias para proteger la integridad y confidencialidad de la red y los datos.
5. Mantenerse actualizado con las últimas tendencias y avances en el campo de SDN. La tecnología SDN está en constante evolución, por lo que es importante estar al tanto de las últimas novedades, estándares y mejores prácticas para garantizar una implementación exitosa y eficiente de SDN en la red.