UNIVERSIDAD TÉCNICA DEL NORTE

INSTITUTO DE POSTGRADO

MAESTRÍA EN TELECOMUNICACIONES

"METODOLOGÍA DE TRANSICIÓN DE ARQUITECTURAS CONVENCIONALES A REDES DEFINIDAS POR SOFTWARE EN PLATAFORMAS DE CÓDIGO ABIERTO CON BASE EN EL ESTÁNDAR ETSI GS NFV-PER 001"

Trabajo de Investigación previo a la obtención del Título de

Magíster en Telecomunicaciones

Director:

Ing. Andrés Fernando Reyes Castro, MSc

Autor:

Ing. Marcela Elizabeth López Huera

IBARRA - ECUADOR

2021

Instituto de Posgrado

UNIVERSIDAD TÉCNICA DEL NORTE

Resolución No. 001-073 CEAACES-2013-13

INSTITUTO DE POSGRADO

MAESTRIA EN TELECOMUNICACIONES

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Grado, presentado por la maestrante Marcela Elizabeth López Huera, para optar por el grado de MAGÍSTER EN TELECOMUNICACIONES, doy fe de que dicho trabajo reúne los requisitos y méritos suficientes para ser cometido a presentación (pública y privada) y evaluación por parte del jurado examinador que de designe.

En la cuidad de Ibarra a los 19 días del mes de agosto de 2021.

ANDRES FERNANDO REYES CASTRO

Firmado digitalmente por ANDRES FERNANDO REYES CASTRO
Nombre de reconocimiento (DN): cn=ANDRES FERNANDO REYES CASTRO, serialNumber=1306211808 25, ou=ENTIDAD DE CERTIFICACION DE INFORMACION, o=SECURITY DATA S.A. 2, c=EC
Fecha: 2021.08.18 12:42:03 -05'00'

Ing. Andrés Reyes, MSc.

TUTOR

Instituto de Posgrado
IBARRA - ECUADOR

UNIVERSIDAD TÉCNICA DEL NORTE

Resolución No. 001-073 CEAACES-2013-13
INSTITUTO DE POSGRADO

MAESTRIA EN TELECOMUNICACIONES

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. Identificación de la obra

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS PERSONALES		
Cédula de ciudadanía	100359767-9	
Apellidos y nombres	López Huera Marcela Elizabeth	
Dirección:	Ibarra, Calle Balzar y Santo Domingo.	
Email:	melopez@utn.edu.ec	
Teléfono:	0990680772 / (06) 2 558 - 392	
DATOS DE LA OBRA		
Título:	METODOLOGÍA DE TRANSICIÓN DE ARQUITECTURAS CONVENCIONALES A REDES DEFINIDAS POR SOFTWARE EN PLATAFORMAS DE CÓDIGO ABIERTO CON BASE EN EL ESTÁNDAR ETSI GS NFV-PER 001	
Autor:	López Huera Marcela Elizabeth	
Fecha:	19 de octubre de 2021	
Sólo para trabajos de grado		
Programa:	□ Pregrado	■ Posgrado
Título por el que opta:	Magíster en Telecomunicaciones	
Tutor:	Reyes Castro Andrés, MSc	

2. Constancia

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrollo, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que se asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 19 días del mes de octubre de 2021

El AUTOR

Marcela Elizabeth López Huera

C.C. 1003597679

DEDICATORIA

Con profundo amor, se lo dedico a mis padres, por ser el cimiento principal en la construcción de mi vida profesional, promotores de mis sueños y por supuesto de mis triunfos, definitivamente no me alcanzará la vida para expresarles todo mi agradecimiento.

A mis hermanos, quienes alegran mi vida con sus ocurrencias.

A mi compañero de aventuras, por los inmensos lotes de felicidad y su apoyo incondicional.

Marcela

RECONOCIMIENTO

A mi tutor, MSc. Andrés Reyes por su paciencia, entereza y sobretodo ser el guía durante el desarrollo de este trabajo.

Al MSc. Fabián Cuzme por su valiosa aportación con sus conocimientos que sirvieron para enriquecer el presente trabajo.

A todos quienes fueron parte de este proceso.

¡Gracias!

Marcela

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Instituto de Posgrado
IBARRA - ECUADOR

UNIVERSIDAD TÉCNICA DEL NORTE

Resolución No. 001-073 CEAACES-2013-13
INSTITUTO DE POSGRADO

"METODOLOGÍA DE TRANSICIÓN DE ARQUITECTURAS CONVENCIONALES A REDES DEFINIDAS POR SOFTWARE EN PLATAFORMAS DE CÓDIGO ABIERTO CON BASE EN EL ESTÁNDAR ETSI GS NFV-PER 001"

Autor: Marcela Elizabeth López Huera.

Tutor: Ing. Andrés Reyes Castro, Msc.

Año: 2021

RESUMEN

La presente investigación tiene por objetivo proporcionar una metodología de transición de arquitecturas convencionales a redes definidas por software a fin de ofrecer un proceso explícito de migración en entornos de redes privadas empresariales. Para ello, se empieza detallando las características de las redes tradicionales y el funcionamiento de las redes definidas por software, protocolos y plataformas que se utilizan para su desarrollo; posteriormente se realizó el levantamiento de información siguiendo una metodología exploratoria. En esta fase, se encontró que, algunas empresas carecen de metodologías y procedimientos debido a diversos factores que les impiden avanzar en la implantación de nuevas tecnologías, por lo que se realiza un análisis de requerimientos funcionales y no funcionales, mismo que sirvió de sustento para la elaboración de la propuesta metodológica.

Seguido se presenta la metodología de transición fundamentada en estándares internacionales como Cisco Safe, NFV-PER-001 y COBIT 5 para finalmente proceder a su validación, la cual abarcó la simulación de tres escenarios controlados con características similares a los diseños de red reales.

Palabras clave: sdn, nfv, mininet, diseño de redes, gns3

Instituto de Posgrado
IBARRA - ECUADOR

UNIVERSIDAD TÉCNICA DEL NORTE

Resolución No. 001-073 CEAACES-2013-13
INSTITUTO DE POSGRADO

"METODOLOGÍA DE TRANSICIÓN DE ARQUITECTURAS CONVENCIONALES A REDES DEFINIDAS POR SOFTWARE EN PLATAFORMAS DE CÓDIGO ABIERTO CON BASE EN EL ESTÁNDAR ETSI GS NFV-PER 001"

Autor: Marcela Elizabeth López Huera.

Tutor: Ing. Andrés Reyes Castro, Msc.

Año: 2021

ABSTRACT

The purpose of this research is to provide a methodology for the transition from conventional architectures to software-defined networks in order to offer an explicit migration process in private enterprise network environments. To this end, it begins by detailing the characteristics of traditional networks and the operation of software-defined networks, protocols and platforms used for their development; subsequently, information was gathered following an exploratory methodology. In this phase, it was found that some companies lack methodologies and procedures due to various factors that prevent them from advancing in the implementation of new technologies, so an analysis of functional and non-functional requirements was performed, which served as a basis for the development of the methodological proposal.

Next, the transition methodology is presented, based on international standards such as Cisco Safe, NFV-PER-001 and COBIT 5, to finally proceed to its validation, which included the simulation of three controlled scenarios with similar characteristics to real network designs.

Keywords: sdn, nfv, mininet, network design, gns3

Capítulo I

El Problema

1.1. Problema de Investigación

Las redes de datos constituyen infraestructuras complejas conformadas por múltiples equipos, por lo que, inicialmente los proveedores de servicios de telecomunicaciones se centraron en el desarrollo de hardware dedicado siguiendo un modelo "un dispositivo, un sistema operativo y una tarea" (Cordero, 2017). Sin embargo, las elevadas exigencias de los usuarios de acceder a la información de una forma ágil y eficiente, promovieron la evolución de nuevas tecnologías, haciendo que la infraestructura de redes convencional resulte poco eficiente, además de incrementar considerablemente las tareas de gestión para el administrador de red (Intel, 2015).

De esta manera, en un entorno de red privada empresarial (EPN), cuando surge la necesidad de agregar una nueva función de red, los departamentos de TI se ven obligados a integrar en su infraestructura algunos dispositivos de propósito particular, lo cual da lugar a inconvenientes de incompatibilidad, ineficiencia energética, aumento de costos de gestión, costos de operación (OPEX) y costos de capital (CAPEX) (Hernandez Valencia, Izzo, & Polonsky, 2015).

Según estadísticas recopiladas por Cisco, en empresas con esquemas de redes convencionales, alrededor del 70% del presupuesto de red se lo utiliza en equipos y mantenimiento; gastando mínimamente en servicios de configuración y software en general; mientras que, en datacenters, los cuales ya manejan las tecnologías de virtualización y redes definidas por software, las cifras se invierten, empleando la mayor parte del presupuesto en la adquisición de varios paquetes de software que a largo plazo retornan la inversión inicial (Network & Virtualization, 2018).

En efecto, líderes de la industria de telecomunicaciones como Cisco, Alcatel, Huawei, EstiNet, y demás, ya han puesto en marcha la comercialización de equipos robustos capaces de soportar tecnologías de virtualización y control de redes definidas por software, sin embargo, sigue siendo hardware específico del proveedor, que mantiene restringida compatibilidad con equipos de otros fabricantes, consecuentemente, si se adquiere uno de éstos, analizar otras marcas compatibles, se volverá una tarea más compleja sin dejar de lado los altos recursos económicos que se requieren (Mijumbi et al., 2016).

Por último, se encuentran los administradores de redes empresariales, que manifiestan sentir incertidumbre de alojar información en dispositivos virtuales, y algunos inclusive, consideran que la migración hacia la virtualización y la gestión centralizada solo está dirigida a las grandes organizaciones de telecomunicaciones, en su mayoría por falta de conocimiento técnico, metodologías, y estándares que indiquen cómo llevar a cabo ésta transición (Telesemana, 2017).

1.2. Objetivos de la Investigación

1.2.1. Objetivo General

Desarrollar una metodología de transición de arquitecturas convencionales a redes definidas por software utilizando plataformas de código abierto con base en el estándar de buenas prácticas ETSI GS NFV-PER 001, a fin de ofrecer un proceso detallado de migración en entornos de redes privadas empresariales.

1.2.2. Objetivos Específicos

- Realizar el estado del arte sobre los aspectos de virtualización de funciones de red (NFV), redes definidas por software (SDN), con énfasis en su arquitectura, plataformas de código abierto, estándar ETSI GS NFV-PER 001 y tecnologías que se pueden utilizar para su implantación.

- Analizar las necesidades actuales, topologías, servicios y funciones de entornos de redes empresariales convencionales para obtener los requerimientos del diseño de red.
- Elaborar la metodología de transición, contemplando los requerimientos de diseño obtenidos y con fundamento en el estándar ETSI GS NFV-PER 001.
- Validar la metodología desarrollada, a través de su implementación en un entorno controlado de código abierto, simulando diferentes escenarios.

1.3. Justificación

Un proceso detallado de cómo llevar a cabo la migración de redes existentes hacia la virtualización de funciones red (NFV) y redes definidas por software (SDN), se vuelve fundamental para entornos empresariales que buscan la evolución y optimización de sus procesos, explotando las mejores técnicas de virtualización a fin de abstraer las características de hardware y software dedicados, evitando la dependencia del proveedor de servicios de telecomunicaciones y sin discriminación de marcas, logrando minimizar costos de operación y mantenimiento (Nguyen, 2019).

China Mobile Communications Corporation, en un estudio realizado en el 2016 arrojó que, al desplegar su red de acceso y centralizarla en una infraestructura virtual, su consumo energético se redujo en un 41% puesto que se minimizaron gastos de suministro eléctrico, al utilizar menos equipos en esa porción de la red y afirma que, si se cambiase toda la red de producción a la nube, el consumo energético podría degradarse hasta un 87% (Mijumbi et al., 2016); de lo que se concluye que, la presente investigación está encaminada al concepto de eficiencia energética, y ahorro económico según la expansión de red.

No obstante, la preocupación con respecto a la seguridad sigue siendo una limitante al cambio de modelo de red; es cierto que NFV y SDN suponen grandes desafíos en éste

tema; motivo por el cual, es necesario hacer énfasis en que a través de NFV es posible virtualizar equipos de seguridad como firewalls o detectores de intrusos que harían el mismo trabajo que un dispositivo físico, y por otro lado, con el control de SDN se podría identificar y enrutar el tráfico presuntamente malicioso para su posterior análisis, dejando a salvo la integridad de los datos (Lee, 2016).

A pesar de ello, la falta de metodologías y procedimientos técnicos estandarizados, que evidencien las ventajas que representa migrar a infraestructuras virtualizadas sigue siendo una de las principales barreras para la implementación de NFV y SDN, de ahí que, surge la necesidad de realizar una metodología dirigida a empresas no proveedoras de servicios de telecomunicaciones, consideradas de mediana escala, como sitios de producción, locales y oficinas que tengan interconectadas sus sucursales, de manera que pueda servir de guía para el despliegue de arquitecturas virtuales sin desaprovechar su infraestructura vigente (Casazza et al., 2019).

Finalmente, cabe señalar que este trabajo se encuentra dentro de la línea de investigación de innovación tecnológica y productos de telecomunicación de la Universidad Técnica del Norte; y vinculado al literal d.10. Incrementar el acceso a servicios públicos de telecomunicaciones y tecnologías de información del parámetro de Lineamientos territoriales de acceso equitativo a infraestructura y conocimiento del Plan Nacional de Desarrollo 20172021 (Plan Nacional de Desarrollo, 2017), puesto que, se desarrollará el procedimiento que permitirá aportar a la mejora tecnológica de los esquemas de redes empresariales actuales del país.

Capítulo II

Marco Referencial

2.1. Antecedentes

Las redes definidas por software (SDN son un paradigma que permiten desagregar el plano del control del plano de datos, dando lugar a una red que se gestiona de forma centralizada y sobretodo programable (Ramirez & Lopez, 2018); mientras que, la virtualización de funciones de red (NFV) tiene por objetivo transformar la manera en que se diseña la red, consolidando las funciones de red en aplicaciones de software que permitan su movilización, ejecución en tiempo real y que puedan ubicarse en diferentes puntos geográficos (ETSI, 2019).

Inicialmente, estas tecnologías se orientaron a centros de datos y proveedores de internet exclusivamente, no obstante, durante los últimos años la aplicación de éstas tecnologías ha ido evolucionando, dejándola abierta, no sólo a la industria dedicada a producir tecnología, sino a pequeñas y medianas empresas que aún mantienen esquemas de redes convencionales, pero, que quieren optimizar sus procesos innovando su infraestructura actual a través de un nuevo modelo de negocio SDN/NFV.

2.2. Referentes Teóricos

2.2.1. Redes Definidas por Software (SDN)

Software Defined Networking o Redes Definidas por Software, comprenden una arquitectura en donde se separan el plano de datos y el plano de control, haciendo que la red pueda ser gestionada de manera centralizada y “programable”, mediante la utilización de APIs capaces de controlar el comportamiento de la red, independientemente de la tecnología de red subyacente (Triana, 2017).

Arquitectura de las Redes Definidas por Software.

La arquitectura SDN fue especificada por The Open Networking Foundation (ONF) considerando las capas de infraestructura, control y aplicación, como se muestra en la figura 1 (Jammal, Singh, Shami, & Li, 2018) :

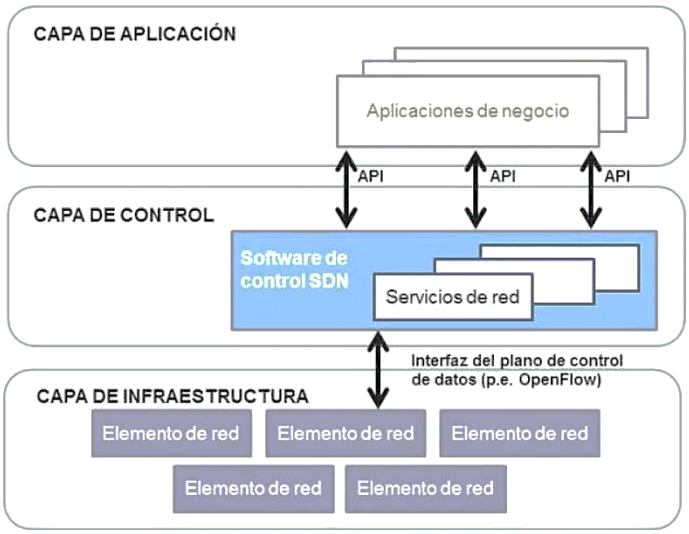


Figura 1. ***Arquitectura de las Redes Definidas por Software. Software-Defined Networking: State of the Art and Research Challenges.***
Nota: Tomado de Jammal, Singh, Shami, & Li (2018).

Este esquema detalla un controlador SDN, y APIs de entrada y salida:

- Controlador SDN: elemento clave que permite el reenvío de paquetes a la capa inferior y superior mediante las APIs southbound y northbound respectivamente, constituye el "cerebro" de la red, pues permite la visualización de todos los elementos que la conforman.
- Una API southbound: comprende la interfaz a través de la cual se enviará información a los switches o routers de la capa inferior permitiendo realizar cambios en tiempo real por lo que se considera una API de entrada, la más implementada es OpenFlow.

- Una API northbound: son las interfaces cuyo objetivo es interconectar el controlador SDN con los diferentes tipos de aplicaciones de la capa superior. Constituye una API de salida porque muestra la interfaz hacia el administrador de red, OpenDaylight es un ejemplo.

La figura 2 indica la arquitectura OpenDaylight, controlador OpenFlow, el cual a su vez es un protocolo que le permite al servidor decirle a los switches de red por donde encaminar los paquetes. En cada red tradicional, cada conmutador tiene software propietario que le especifica las reglas de enrutamiento.

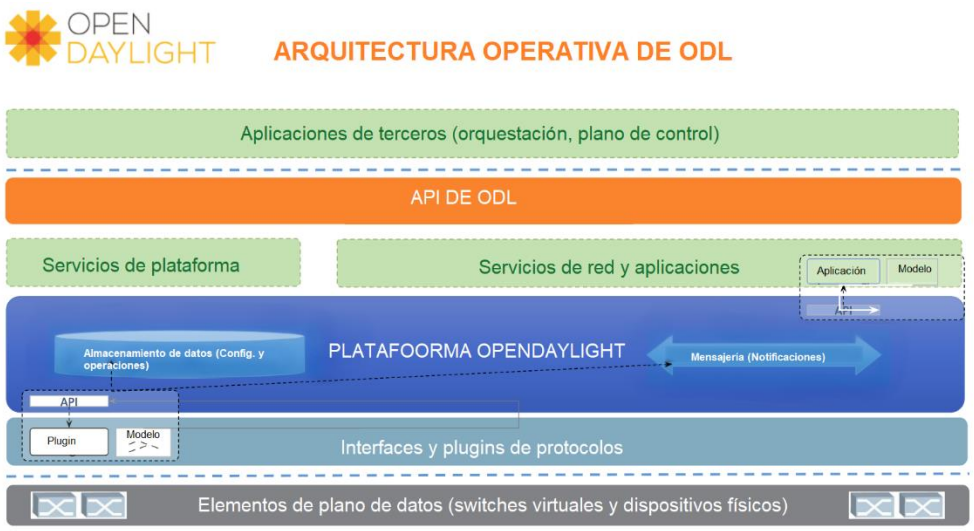


Figura 2. ***Arquitectura operativa de OpenDaylight***
Nota: Tomado de ONF (2019)

2.2.2. Virtualización de Funciones de Red (NFV)

Network Functions Virtualization o Virtualización de Funciones de Red surgió como alternativa a la dependencia de hardware propietario, en donde el objetivo es migrar a una infraestructura en donde las funciones de red puedan ser virtualizadas (VNFs) dentro de un servidor de alta capacidad en lugar de funcionar sobre equipos específicos. NFV convierte las

funcionalidades de dispositivos de red en aplicaciones tipo software que permitan su ejecución en tiempo real. (Ramirez & Lopez, 2018).

En la figura 3 se muestra una comparativa de virtualización entre el modelo tradicional vs el modelo “bare” y el modelo “alojado”.

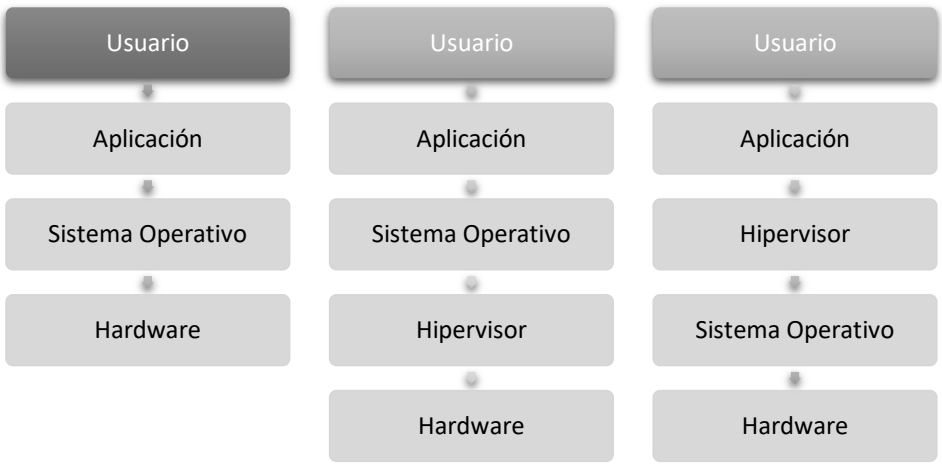


Figura 3. ***Modelo Tradicional vs Bare Model vs Hosted Model. A Study On Virtualization Techniques And Challenges In Cloud Computing.***
Nota: Tomado de (Durairaj & Kannan, 2014)

Arquitectura de la Virtualización de Funciones de Red.

El Instituto Europeo de Estándares de Telecomunicaciones (ETSI) propone un modelo de abstracción de dispositivos, consolidados en un solo hardware robusto con un framework de tres elementos principales, así se evidencia en la figura 4:

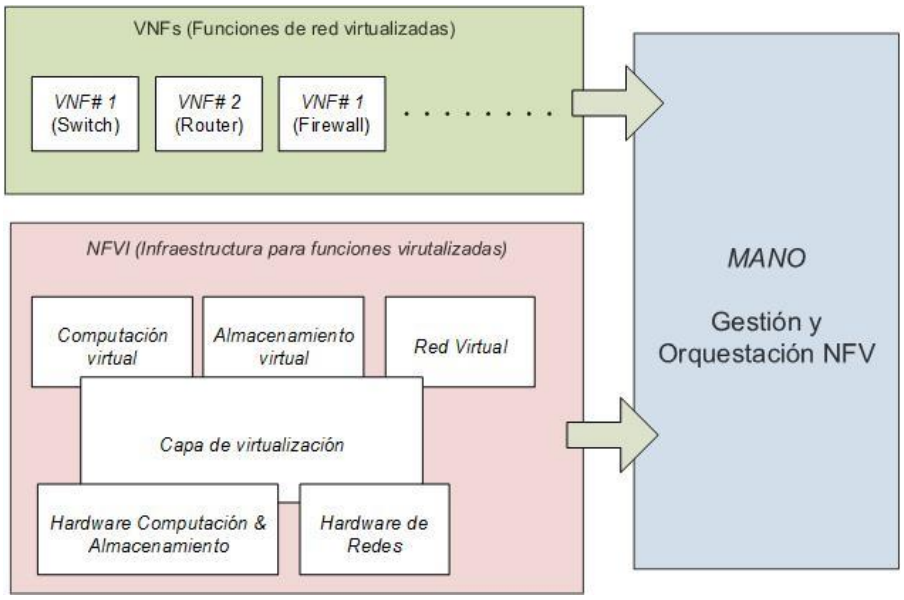


Figura 4. ***Arquitectura de virtualización de funciones de red. An Overview of NFV Elements***
Nota: Tomado de SDX Central (2018)

- Network Functions Virtualization Infrastructure (NFVI): hace referencia al servidor físico o hardware en el cual se alojarán los recursos y dispositivos virtualizados (Rajendra & Syed Farrukh, 2016).
- Virtualized Network Function (VNF): constituyen las funciones de red virtualizadas sobre la NFVI (Rajendra & Syed Farrukh, 2016).
- Management and Orchestration (MANO): es la capa de orquestación que está entre la NFVI y las VNFs, se encarga de la creación, eliminación y reserva de recursos necesarios para la administración de las máquinas virtuales (Rajendra & Syed Farrukh, 2016).

2.2.3. Características Generales de Arquitecturas Convencionales de Red

Pese a la evolución de la tecnología durante las últimas décadas aún existen compañías que manejan un diseño de red descentralizado, utilizando un servidor para cada tarea, a este esquema se denomina característica de red distribuida.

Luego, se encuentran aquellas que utilizan servicios alojados en servidores proporcionados por un proveedor externo; y finalmente están las organizaciones que ya han

dado el primer paso para un diseño de red inteligente: la virtualización, pero, por diferentes motivos no avanzan hacia una red automatizada. Los detalles de las características enunciadas se precisan a continuación:

Característica de Red Distribuida.

La característica de red distribuida (véase figura 5), se dio a conocer por primera vez por J. Postel en el RFC 791 (IETF, 1981) , y en este caso, se refiere a que las arquitecturas convencionales mantienen un diseño de red descentralizado, debido a que el reenvío de paquetes aún se encuentra embebido en cada dispositivo de red, por lo que, cada router o switch debe manejar protocolos como OSPF, IS-IS, STP que colaboren en la toma de decisiones de reenvío (Osaba, 2016).

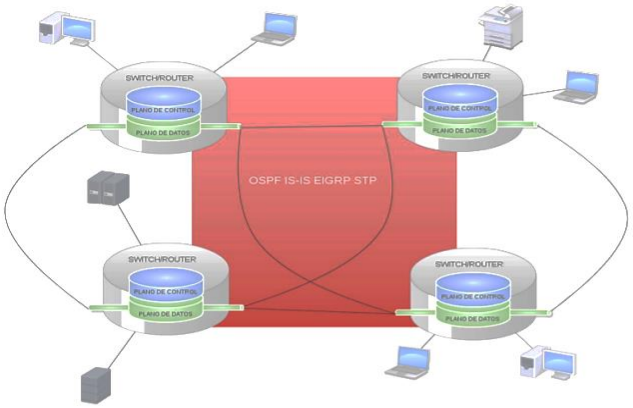


Figura 5. ***Arquitectura de red tradicional. Virtualización en redes definidas por software.***
Nota: Tomado de Osaba (2016).

Dispositivos de Propósito Particular.

Los dispositivos de red intermedios como switches o routers son los que permiten encaminar los paquetes en diseños convencionales de red, esto es posible debido a que en el mismo hardware se localiza el plano de datos y el plano de control, permitiendo el ruteo independiente de cada paquete entrante, diríjase a la figura 6 (Osaba, 2016).

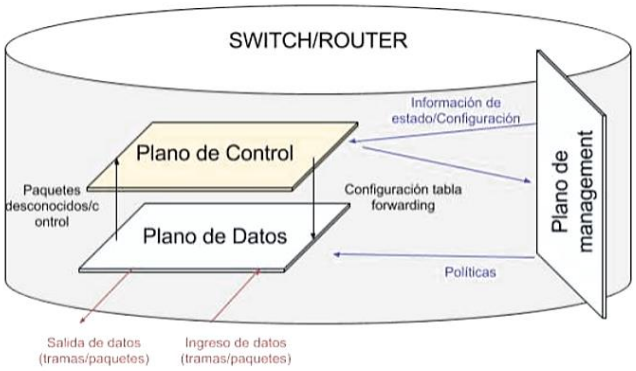


Figura 6. ***Naturaleza particular de routers y switches. Virtualización en redes definidas por software.***
Nota: Tomado de Osaba (2016).

Servicios en la Nube.

Al continuar analizando la infraestructura de empresas, se llega a aquellas que, además de tener sus propios equipos de red y servidores, alquilan otro tipo de servicios, como por ejemplo correo electrónico o tipo web, a un proveedor externo, lo que se conoce como "hosting", que dependiendo del tamaño de la empresa podría representar elevados costos versus los beneficios.

Dependiendo de las necesidades, las empresas suelen contratar diferentes tipos de servicios en la nube, cada uno de éstos ofrece prestaciones según los requerimientos deseados; una de las ventajas de esto, es que la empresa se deslinda del mantenimiento y soporte del servicio, no obstante, podría resultar contraproducente por cuando el administrador de red se encuentra inhabilitado de personalizar el hosting teniendo que recurrir directamente a su proveedor externo (Torres, 2020).

Virtualización.

Posteriormente se encuentran las compañías que con el fin de ahorrar CAPEX (capital expenditure o inversiones de capital) ya incursionaron en la virtualización, ejecutando múltiples sistemas operativos en un solo hardware servidor, permitiendo segmentar un

sistema grande en muchas partes más pequeñas, haciendo que los usuarios utilicen el servidor de manera más eficiente (Red Hat, 2019).

2.2.4. Inconvenientes de las Redes Convencionales

A continuación, se evidencian las debilidades que están atravesando las arquitecturas de redes actuales en entornos empresariales, vinculadas al diseño que denota complejidad en escalabilidad y gestión.

Escalabilidad.

Uno de los principales problemas de las empresas es que, debido a que sus objetivos de negocio no parecerían relacionarse con la tecnología, no creyeron factible invertir en ella puesto que no es su "prioridad", sin embargo, al crecer su negocio levantaron una infraestructura tecnológica por necesidad; pero, en muchos de los casos diseñaron su red para sus requerimientos actuales solamente, sin planificar su extensión, por lo que al crecer la demanda de usuarios, servicios y aplicaciones tienen que realizar rediseños constantemente (CISCO, 2018).

Gestión Compleja.

En las redes tradicionales, algunos protocolos como OSPF y BGP se manejan con nodos que comparten la información con sus vecinos y de manera limitada para evitar la cogestión de red. Esto implica que no se tiene una visión global de la red, si se requiere controlar o modificar una ruta en particular para un determinado flujo, el administrador debe modificar individualmente directa o remotamente los parámetros y prioridades para lograr el comportamiento deseado de red, haciendo más difícil su gestión a medida la red se expande (Hakiri, Gokhale, Berthou, Schmidt, & Gayraud, 2015).

Compatibilidad con Servicios en la Nube.

Los servicios en la nube han tomado protagonismo en los últimos años, dando la posibilidad de adquirir soluciones IaaS (Infraestructura como Servicio), permitiendo el aprovisionamiento de almacenamiento, bases de datos y servidores; aquí se encuentran AWS (Amazon Web Services) o Microsoft Azure, entre otros. No obstante, el tiempo de inactividad que se ocasiona al migrar aplicaciones "on premise" hacia la nube, es uno de los principales inconvenientes (Osaba, 2016).

2.2.5. Principales Fabricantes de Tecnologías SDN/NFV

Varios de los líderes de la industria de telecomunicaciones están innovando en el despliegue de soluciones SDN/NFV, así por ejemplo Cisco y Microsoft ya han lanzado al mercado soluciones propietarias orientadas a éstas tecnologías. En la tabla 1 se describen los principales fabricantes del medio.

Tabla 1.
Principales fabricantes de tecnologías SDN/NFV

Item	Fabricante	Enfoque	Solución comercial
1	Cisco	SDN para centros de datos.	Cisco One: kit de hardware y software que facilita la gestión de la red.
2	Hp	SDN/NFV en dispositivos intermedios.	Virtual Application Networks (VAN): switches compatibles entre sí que abarcan las capas de infraestructura, control y aplicación.
3	Microsoft	SDN en la nube	Azure SDN: servidores en la nube. SDN VE Platform: controlador
4	IBM	Basados en código abierto.	SDN con base en la arquitectura DOVE (Distributed Overlay Virtual Ethernet) compatible con switches OpenFlow.
5	Huawei	SDN/NFV	SoftCOM: reestructuración del modelo de negocio.

Nota: Tomado ETSI ESTÁNDAR (2019)

Costos Elevados.

Generalmente las empresas se ven limitadas cuando se trata de adquirir equipamiento de networking, a causa de la gran inversión que esto representa y esto se debe a que los principales actores del mercado ofrecen soluciones propietarias y cerradas restringiendo las posibilidades de adquisición, incluso a los propios vendors (Osaba, 2016).

De esta manera se presentan algunos de los obstáculos de las redes tradicionales, que servirán de punto de partida para fundamentar la implementación de nuevos paradigmas de redes, que darán lugar a redes inteligentes que soporten la demanda de los servicios actuales.

2.2.6. Plataformas de Código Abierto

Para el despliegue de estas tecnologías existen plataformas de código abierto a través de las cuales se puede llevar a cabo la implantación de las redes definidas por software y la virtualización de funciones de red. Tal que, OpenFlow es el principal protocolo de comunicación para el despliegue SDN, mientras que OpenStack es otra poderosa herramienta para aprovisionar recursos en la nube.

Protocolo OpenFlow.

Es el estándar principal de código abierto a través del cual se puede implantar nodos virtuales y orquestarlos mediante el controlador SDN. Se encuentra definido en la capa de control y permite la interoperabilidad entre fabricantes, mediante de dispositivos de red físicos o virtuales basados en hypervisores, asegurando el despliegue y coexistencia con las redes actuales (Errera, 2014).

Este protocolo analiza la definición de flujo para identificar el tráfico, basándose en la implementación de políticas programables que indiquen a los paquetes cómo atravesar los

dispositivos de red mediante patrones, aplicaciones, entre otros (Spera, Development, Logicalis, & Cone, 2014).

OpenStack.

OpenStack surgió como desarrollo de un proyecto liderado por la Openstack Foundation y empresas como: IBM, Dell, Intel, VMWare, entre otras, que, basado en la computación en la nube y software libre brindan una infraestructura como servicio (IaaS) que proporciona al administrador de red una gestión de alto nivel, en redes planas o segmentación en vlans, con la opción de crear sus propias redes aprovechando las mejores características de las redes definidas por software (García Ibáñez, 2016).

OpenDaylight.

OpenDaylight es un controlador OpenFlow fundamentado en la tecnología de contenedores karaf, que tiene por objetivo centralizar el envío y recepción de paquetes a fin de que la red se programa de manera independiente de los switches individuales (Smith, 2018).

Este controlador puede operar en diversos entornos de red, utiliza la interfaz hacia el norte o northbound para la comunicación con las APIs que son parte de la red, mientras que la interface hacia el sur o southbound establece la comunicación con los dispositivos intermedios como: switches o routers; así, se pretende minimizar los tiempos de respuesta en caso de incidentes y por supuesto optimizar la administración de redes (Cordero, 2017).

Open vSwitch.

Es un switch virtual para la gestión de redes en hipervisores, con soporte en openflow, es capaz de soportar ipv6, protocolos de tunneling como IPsec, GRE, VXLAN, entre otros, permitiendo incluso configurar calidad de servicio QoS (Jammal et al., 2018).

2.2.7. Seguridad en SDN/NFV

Al combinar estas tecnologías se logra tener una arquitectura de red centralizada, lo que posibilita visualizar todo el esquema de red, en donde uno de los objetivos de esta arquitectura centralizada es detectar amenazas y actuar de forma proactiva en caso de ataques; sin embargo, esto conlleva nuevos retos en el ámbito de la seguridad, pues los intrusos podrían intentar atacar el dispositivo central o controlador SDN, dando de baja el funcionamiento normal de la red (Roncancio R, Ginna & Sáenz G, 2016).

De esta forma, una red SDN/NFV debe proveer niveles indispensables de seguridad, como por ejemplo considerar la implantación de dispositivos intermedios como firewalls o IDS/IPS, que en su mayoría se aplican dentro del mismo controlador SDN, y por supuesto contar con imágenes de respaldo en caso de NFV (Ramiro, 2017).

2.2.8. Metodologías y Estándares.

Estándar ETSI GS NFV-PER 001.

El estándar GS NFV-PER 001 v1.1.1 propuesto por el Instituto Europeo de Normas de Telecomunicaciones constituye una lista de recomendaciones y buenas prácticas para selección adecuada del hardware en donde se levantará el hipervisor “bare metal” o “alojado”, con el objetivo de lograr una carga de trabajo equilibrada de las funciones de red virtualizadas (VNFs) para el plano de datos, plano de control, entre otros (ETSI, 2014).

Metodología para Identificación de Cuellos de Botella. Este estándar propone seguir un proceso de dos pasos para identificar cuellos de botella en la red y posteriormente utilizar esta información para seleccionar la tecnología SDN/NFV, así se tiene:

- Primer paso: analizar el rendimiento de red en “bare metal”, es decir, en la red sin aplicar la virtualización NFV, teniendo en cuenta el procesamiento de paquetes como métrica principal (ETSI, 2014).

- Segundo paso: comprende realizar el análisis de red en el hardware que contiene la virtualización, tomando en cuenta memoria, I/O, entre otros; para finalmente identificar los nuevos cuellos de botella y tratar de reducir la brecha entre el "bare metal" y el entorno NFV al máximo (ETSI, 2014).

Método de Pruebas Funcionales en el Entorno Virtualizado. Consiste en el reenvío de paquetes, creando tráfico entre hosts o a su vez modo cliente-servidor. ETSI GS NFV-PER 001 establece el procedimiento siguiente:

Paso 1: Generar carga de tráfico, que incluya al menos los siguientes protocolos: http, dns, ftp, telnet, ssh, tráfico generado por redes sociales y tráfico de fondo.

Paso 2: Registrar los datos obtenidos.

- Identificar la cantidad de máquinas virtuales por instancia.
- Analizar los recursos NFVI utilizados (memoria, procesador, almacenamiento).

Paso 3: Benchmarking en la escala de rendimiento.

- Llevar a cabo las pruebas de funcionalidad aumentando la cantidad de usuarios o hosts de dispositivo final.

Paso 4: Resultados. El rendimiento de los usuarios existentes, al realizar el paso 3, no debe verse afectado significativamente, aun cuando los usuarios adicionales añadan carga de trabajo.

Elección del Hipervisor. Finalmente, el objetivo del estándar es decidir qué hipervisor utilizar para lo cual se sugieren seguir las tres etapas que se detallan a continuación:

Paso 1: Identificar las necesidades de la empresa, por ejemplo: flexibilidad, escalabilidad, ecosistema robusto, disponibilidad; la interpretación de disponibilidad puede variar, por

ejemplo, algunas podrían admitir que el hipervisor se duerma en las noches y otras requerir un 99.99% de disponibilidad de las aplicaciones.

Paso 2: Comprender las características que ofrece el mercado. La mayor diferencia entre soluciones existentes radica en la usabilidad, el nivel de experiencia del personal de TI y las necesidades planteadas en el paso 1.

Paso 3: Comparar costos. Las organizaciones se encuentran en el negocio de ganar dinero, por lo que la solución que se escoja debe tener un impacto positivo en la economía de la empresa.

Prueba de Convergencia Finales. Una de las métricas para evaluar un entorno virtualizado y definido por software es el tiempo de convergencia que sirve para determinar los tiempos de respuesta y la alta disponibilidad de los servicios de una red. Lo primero será iniciar el protocolo o reglas de enrutamiento en los dispositivos virtuales, esperar a que se realice la convergencia de red, para posteriormente mover un host con una función de red virtualizada (VNF) y registrar los tiempos de pérdida y restablecimiento de paquetes.

El estándar sugiere utilizar la técnica “bare” para comparar los resultados obtenidos antes y después de la virtualización (ETSI, 2014).

Metodología PPDIOO.

PPDIOO es un procedimiento definido por Cisco orientado al ciclo de vida continuo de los servicios requeridos en una red, véase figura 7. Este enfoque permite llevar un proceso de diseño organizado, identificar y validar los requisitos tecnológicos y aumentar la disponibilidad de la red (Balaji Sivasubramanian, Erum Frahim, 2010).

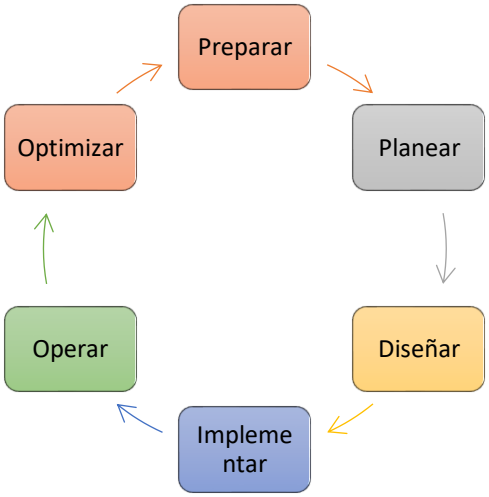


Figura 7. ***Metodología PPDIO. Analyzing the Cisco Enterprise Campus Architecture***
Nota: Tomado de CiscoPress (Balaji Sivasubramanian, Erum Frahim, 2010)

Arquitectura SAFE.

Modelo que proporciona las pautas de diseño para la construcción de redes confiables y seguras que tengan capacidad de resistencia frente a diversas formas de ataque. SAFE divide la red en áreas lógicas llamadas lugares en la red (Places In Network: PIN); cada PIN es tratado de manera diferenciada (Safe Cisco, 2015). Se propone en primer lugar recabar la siguiente información:

- Identificar los objetivos de la empresa.
- Dividir la red empresarial en piezas manejables.
- Desarrollar criterios para el éxito del negocio. Es posible establecer prioridades en los procesos internos.
- Identificar los riesgos, amenazas y políticas vigentes.

Y, al final construir la propuesta de diseño y seguridad con base en las tres fases del método SAFE que se muestran en la figura 8:

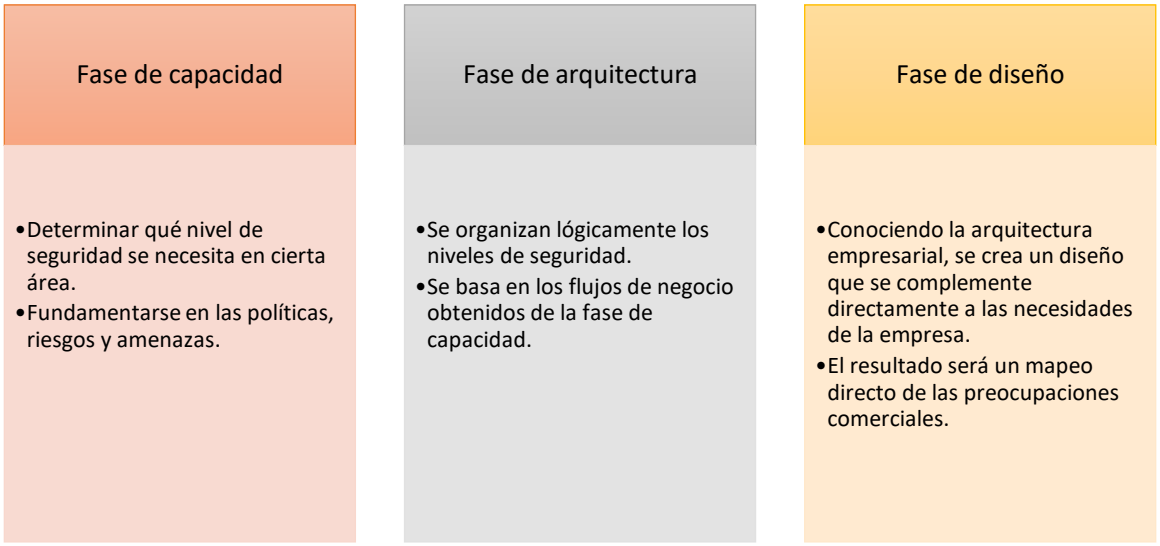


Figura 8. ***Arquitectura SAFE. SAFE Overview Guide***
Nota: Tomado de Safe Simplifies Security (Safe Cisco, 2015).

COBIT 5.

Vincular los objetivos comerciales con la infraestructura de TI es el aspecto principal de la orientación comercial de COBIT. Esto se hace proporcionando diferentes modelos de madurez y creando diferentes métricas para que una empresa mida el logro del marco. La versión actual del framework, COBIT 5, surgió a partir del año 2012 aumentando el contenido y funciones que su antecesor, COBIT 4.1. COBIT 5 cumple con las pautas de estándares y normativas internacionales como: ITIL, COSO, PRINCE2, Val IT, Risk IT e ISO (27001 o ISO 27002) (Horvath, 2021).

2.2.9. Mininet como Herramienta de Simulación

Mininet es un software emulador que proporciona un entorno virtual para el despliegue de redes, utilizando diferentes dispositivos de red en un núcleo basado en Linux. A través de una interfaz de líneas de comandos (CLI) se pueden ejecutar scripts e interactuar con la red, utiliza el puerto TCP 6634 (Cordero, 2017) .

Características de Mininet.

Por medio de sencillos comandos es posible levantar cualquier topología de red, utiliza Python como lenguaje de programación y lo más importante es que permite emular el renvío de paquetes mediante interfaces reales, lo que lo convierte en un software sencillo pero robusto para experimentación en el despliegue de redes definidas por software (Cordero, 2017).

Mininet crea una red virtual realista, ya que ejecuta el kernel real, conmutador y código de aplicación , en una sola máquina. Utiliza virtualización ligera para hacer que un solo sistema parezca una red completa, ejecutando el mismo kernel, sistema y código de usuario. Un host Mininet se comporta como una máquina real; por lo que es posible utilizar la línea de comandos como ping o ssh en cada host. Los programas que ejecuta pueden enviar paquetes a través de lo que parece una interfaz Ethernet real, con una velocidad de enlace y un retraso determinados. Cuando dos programas, como un iperfcliente y un servidor, se comunican a través de Mininet, el rendimiento medido debe coincidir con el de dos máquinas nativas.

Aunque mininet posee muchas características funcionales se debe tomar en cuenta que utiliza los recursos del servidor físico en el que se aloja por lo que éstos se compartirán con los hosts virtuales emulados.

En conclusión del presente capítulo, se puede afirmar que SDN y NFV son tecnologías compatibles, pero independientes, por lo que no es necesario implementar una para que funcione la otra, dicho de mejor manera, NFV brinda la infraestructura para que SDN se ejecute, pues actúa como el sistema operativo de las redes (Hernandez Valencia, Izzo, & Polonsky, 2015).

Capítulo III

Marco Metodológico

Esta investigación está orientada a sitios que cuentan con una red privada empresarial (EPN), de modo que, en este capítulo se empezará describiendo el entorno al que se está aplicando, el enfoque y tipo de investigación, el procedimiento y finalmente las consideraciones bioéticas que se van a utilizar para el desarrollo de la metodología propuesta basada en el estándar ETSI.

3.1. Descripción del Área de Estudio

Una red privada empresarial (EPN) constituye el clásico esquema que conecta sucursales remotas de una empresa con su sede central, como se indica en la figura 9, utilizando tecnologías y protocolos de routing como OSPF, MPLS/IP, entre otros. La oficina central es el núcleo de la red empresarial, allí se localizan todos los dispositivos de red físicos o virtuales y servicios a los cuales convergen las sucursales. Los servidores a los que acceden las sucursales cumplen las típicas funciones como: mensajería (pop, imap), transferencia de archivos (ftp), web (http), entre otros (Shoaib, 2019).

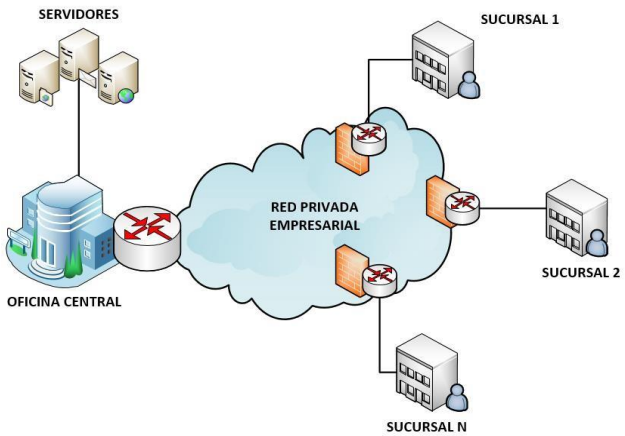


Figura 9. ***Arquitectura del entorno de investigación. Red Privada Empresarial***
Nota: Elaboración propia.

Dentro de esta arquitectura pueden existir dispositivos de seguridad intermedios como cortafuegos, proxys, detectores de intrusos, entre otros; todos con el objetivo de permitir, denegar, limitar, cifrar o descifrar el tráfico de entrada o salida, con base en políticas y criterios previamente configurados (Shoaib, 2019).

Consecuentemente, las EPN surgieron en el ámbito de tener una gestión central en la "oficina matriz" de la empresa, sin embargo, la realidad actual se desapega de este concepto, debido a la masiva expansión de servicios y equipos que se necesitan implantar a medida que la red se va escalando, lo que ha provocado que en las sucursales también se alojen dispositivos de red, descentralizando la gestión.

3.2. Diseño y Tipo de Investigación

Se utilizó una investigación exploratoria, a través de la cual se recolectaron las bases teóricas que sirvieron de sustento para el desarrollo de los siguientes procedimientos; seguidamente la investigación fue de tipo descriptiva, ya que se exploraron las características de arquitecturas de redes convencionales para establecer requerimientos y necesidades.

3.3. Procedimiento de Investigación

Este documento pretende ser una guía de transición orientada principalmente a la industria donde su giro de negocio no es la tecnología, por lo que, para la gestión de requerimientos se optó por la aplicación de metodologías de ingeniería de requerimientos tales como: metodología DoRCU o Documentación de Requerimientos Centrada en el Usuario propuesta por Griselda Báez & Brunner (n.d.), metodología RUP o Proceso Unificado Racional y el Levantamiento de Requerimientos Basado en el Conocimiento del Proceso (Pérez-Virgen, Salamando-Mejía, & Valencia-Ayala, 2013).

Para ello se proponen tres fases: elicitación de requerimientos, análisis de requerimientos, y especificación de requerimientos.

3.3.1. Elicitación de Requerimientos

Corresponde a la etapa de observación y recolección de información, la cual está dividida en fuentes de información primarias y fuentes de información secundarias. Las fuentes de información primaria hacen referencia al estudio de casos de éxito de NFV/SDN y la observación directa mediante entrevistas; mientras que las fuentes de información secundaria se obtuvieron mediante la revisión bibliográfica de textos, artículos y documentos oficiales actualizados.

Levantamiento de Información de Fuentes Primarias

Casos de Uso. Se tomaron en cuenta casos de éxito que ya han aplicado las tecnologías en cuestión hace ya varios años, para ello se construyó un directorio (véase tabla 2) de Organizaciones Empresariales reconocidas a nivel mundial que son partícipes en el impulso de las redes definidas por software y la virtualización de funciones de red, ésta información fue extraída de la sección de colaboradores de la página web oficial de la Open Networking Foundation (ONF), fundación sin fines de lucro que impulsa la transformación de la infraestructura de red y los modelos comerciales a través de la utilización de software de código abierto y los estándares definidos por software para revolucionar la industria de las telecomunicaciones (Open Networking Foundation, 2021).

Tabla 2.
Directorio de casos de éxito

Razón Comercial	Sede	Caso de éxito	Año de implementación
Ciena Corporation [a]	Maryland, Estados Unidos	Transformación basada en software en centros de datos	2020
Radisys Corporation[b]	Oregon, Estados Unidos	Connect Open Broadband, distribución de red óptica pasiva definida por software (PON)	2020

Adtran, Inc[c]	Alabama, Estados Unidos	Estrategia de Renovación y Evolución del Acceso (ARES)	2020
China Unicom[d]	Hong Kong, China	SD-WAN Service	2020
Turk Telekom Group[e]	Ankara, Turquía	vRAN: Red de acceso de radio basada en SDN (redes definidas por software) tecnología para LTE y 5G con 40 solicitudes de patente	2020
Amazon Inc[f]	Washington, Estados Unidos	Conectividad SD-WAN con AWS Transit Gateway Connect	2020
AT&T [g]	Dallas, Estados Unidos	En proceso de transición (75% hasta 2019)	2019
Stanford University [h]	California, Estados Unidos	Migración a un nivel híbrido en usuarios inalámbricos	2014
Google LLC	California, Estados Unidos	Migración de Google WAN	2014
Intel Corporation	California, Estados Unidos	Evolución del uso de la nube en Centros de datos: almacenamiento, computación y red	2014
NTT DoCoMo	Sanno Park Tower, Japón	Mobile Packet Core (EPC)	2011

Nota: Elaboración propia. Recopilado de Open Networking Foundation (2021). [a]"Ciena: Software-based transformation" (2020). [b]Radisys (2020). [c]Schulte (2020). [d] China Unicom (2020). [e] Turk Telecom (2020). [f] Vaidy A, Guru Kannan (2020). [g] Fuetsch (2019). [j]Mahankali & Rungta (2014).

Entrevistas. Esta investigación resguarda la identidad de los participantes que fueron parte del proceso por lo que, la información recopilada es de carácter confidencial y debido a ello se exhiben como anónimas. Se aplicó un cuestionario a algunos líderes de los departamentos de Tecnologías de la Información, Sistemas y Telecomunicaciones de la provincia de Imbabura, las interrogantes se efectuaron con base en los requerimientos revisados anteriormente y el marco de referencia COBIT 5 que proporciona las pautas para alinear los objetivos de control de cada organización con sus metas comerciales, el Anexo A muestra el cuestionario completo.

La figura 10 muestra los resultados de las entrevistas realizadas a los jefes del área tecnológica de organizaciones dedicadas a actividades como manufactura, textiles, insumos agrícolas, y ventas en general, los cuales arrojaron que, las organizaciones sí cuentan políticas de Tecnologías de la Información, sin embargo la mayoría manifiesta que no existe un manual de procesos exclusivo para el departamento de TI por cuanto el recurso humano en esta área es reducido y realizan todo tipo de funciones, existe un espacio destinado para los equipos tecnológicos, no obstante, a medida que se ha ido expandiendo la red de datos, este espacio se ha reducido incluso obstaculizando las vías de acceso al mismo, ninguna de las empresas tiene el mando en cuanto a configuraciones de nivel lógico y dependen en su totalidad de su Proveedor de Servicios de Internet lo cual presenta algunas veces inconformidades en cuanto los tiempos de respuesta. La mayoría considera también que el Departamento de TI está de cierto modo “relegado” ya que los recursos se destinan a otras áreas consideradas de mayor importancia y por ello algunos procesos aún se realizan de forma manual.

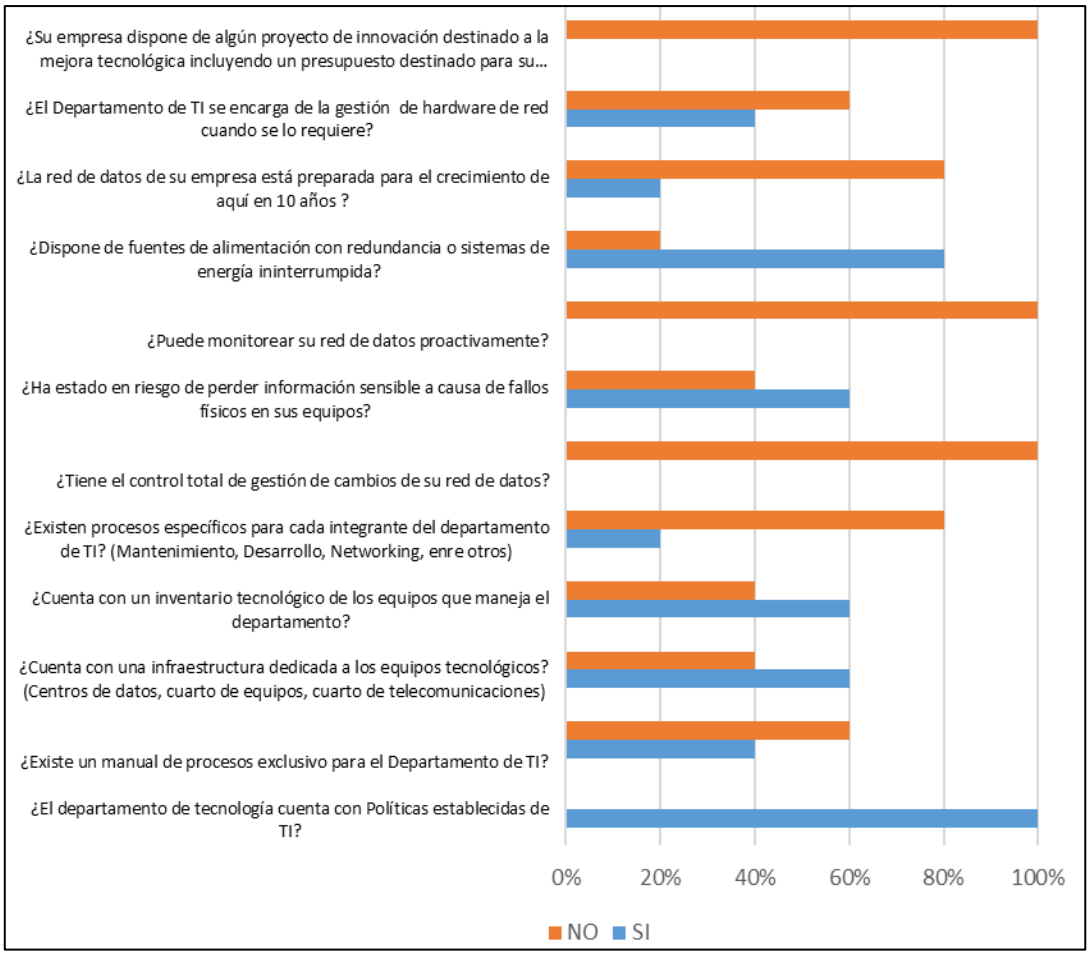


Figura 10. ***Resultados de entrevistas realizadas a Departamentos de TI de Imbabura.***
Nota: Elaboración propia.

Levantamiento de Información de Fuentes Secundarias

Detalle de bases de datos electrónicas para la búsqueda de información. Se tomaron en cuenta repositorios digitales de bibliotecas virtuales como: Scopus, IEEE Xplore, Ebsco Host, motores de búsqueda como google schoolar y repositorios de universidades.

Procedimiento de Búsqueda. Se llevó a cabo una búsqueda exploratoria con algunos filtros que permitieron delimitar de mejor manera la estrategia de investigación

- Fecha de publicación del documento: a partir de 2015
- Empresas consolidadas: núcleo de negocio no enfocado a la tecnología.

- Palabras de búsqueda: redes tradicionales, red de datos, diseño de redes
- Fuente: con acceso al texto completo
- Idioma: español e inglés.

Selección de Documentos. Se procedió a realizar una selección de los documentos revisados de los cuales alrededor del 50% fueron excluidos por no aportar significativamente a la investigación. Se tiene un total de 21 documentos que cumplen con los criterios de selección. Estas investigaciones se analizarán nuevamente el siguiente apartado. La tabla 3 indica los documentos revisados.

Tabla 3.
Ficha de resumen de revisión bibliográfica

Item	Estudio	Autor/es	Año	Idioma	Palabras claves
1	Arquitectura empresarial con Zachman Framework en una Compañía de Manufactura.	Hendy Tannady, Bernadus Gunawan Sudarsono, Johanes Andry, Yuliawan Krishartanto	2020	Inglés	Arquitectura, framework, manufactura
2	Impacto de la computación en la nube en cuanto a coste y seguridad. Caso de uso: Vodacom Tanzania Public Limited Company.	Mary Corrine Mtumbuka	2020	Inglés	computación en la nube, seguridad informática, servicios
3	Un marco de big data para la seguridad de la red de pequeñas y medianas empresas para la informática del futuro.	Ha-Kyun Kim , Won-Hyun So , Seung-Mo Je	2019	Inglés	Big data, framework, python
4	Diseño de una red de datos para el mejoramiento de la gestión de comunicación interna en UNIANDES Quevedo.	Ítalo Mecías Serrano, Luis Javier Molina, Andrea Raquel Zúñiga	2019	Español	Diseño, Metodologías, Cisco System, Optimización, VoIP.
5	Diseño e Implementación de la red de datos del laboratorio centro de desarrollo de software y productos IOT de la facultad de ingeniería de la Universidad Católica de Santiago de Guayaquil.	Victor Hugo Guerra	2019	Español	Infraestructura, redes de datos, redes virtuales, software
6	Diseño de la infraestructura de redes para la mejora de la comunicación de datos en la empresa SEAFROST fundamentado en la norma TIA/EIA-942A.	Félix Adolfo Franco García	2019	Español	Infraestructura de redes, Comunicación de datos, Norma TIA/EIA-942A

7	Rediseño de la Red LAN en la Empresa VLACAR S.A.C	Mirko Fidel Oroya	2019	Español	Rediseño, Transmisión, Topología
8	Análisis de la estructura de Información del Corporativo de Redes Empresariales.	Vladimir N. Azarov , Evgeny A. Saksonov , Yury L. Leokhin	2018	Inglés	Redes corporativas, diseño de redes corporativas, modelos matemáticos
9	Diseño e implementación de una red de datos segura para la Pontificia Universidad Católica del Ecuador, Santo Domingo.	Carlos V. Galarza-Macancela	2018	Español	LAN; seguridad; confidencialidad; integridad; disponibilidad
10	Sistemas de Información en la Empresa. Diseño de Red Empresarial Segura.	N Y Saigushev, U V Mikhailova,O A Vedeneeva, A A Tsaran	2018	Inglés	Empresas, redes corporativas, redes seguras
11	Evaluación de redes de centros de datos y futuros caminos.	Sohini Basu	2017	Inglés	centros de datos, redes de telecomunicaciones
12	Evaluación de la Infraestructura Tecnológica basado en Estándares de Control Interno Caso: Empresa National Tire Experts S.A.	Ynge Vanessa Cedeño Rodríguez	2017	Español	Evaluación, infraestructura tecnológica, estándares de control interno, COBIT
13	Diseño de redes informáticas para universidades en Países en desarrollo.	Rafid Salih Sarhan AlSarhan	2016	Inglés	Diseño de redes, diseños de redes en universidadeds
14	Diseño de una Red LAN para la Empresa INTEL CORP.	Libardo Niño Cruz, Camilo Andrés Vento Serrato	2016	Español	diseño, redes, redes LAN
15	Suministro de Calidad de Servicio en tiempo real para video Interactivo sobre Redes Definidas Por Software.	Harold Owens	2016	Inglés	MPLS, SDN, QoS

16	Diseño de redes informáticas empresariales: una forma simple.	Asif Hassan , Raktim Mondol	2015	Inglés	Redes de computadoras, buenas prácticas de diseño de redes
17	Efectos del conocimiento de la seguridad cibernética en la detección de ataques.	Noam Ben-Asher , Cleotilde Gonzalez	2015	Inglés	seguridad informática, redes, sistemas de intrusión, sistemas de detección
18	Diseño e Implementación del Esquema de Seguridad Perimetral y Monitoreo para la Red de Datos en una Empresa Industrial.	Christian Miranda	2015	Español	Empresas industriales, seguridad perimetral, gestión de producción
19	Diseño de una Red LAN para el transporte de voz, datos y video para el Municipio del Cantón Valencia Provincia de Los Ríos.	Byron Verdezoto	2015	Español	LAN, red de transporte, voz, video
20	Diseño de un esquema de intercambio de información de red empresarial basado en tecnología de seguridad.	Jun Ji, Fei-Fei Xing, Yu-Qing Zang	2015	Español	Seguridad de la información, seguridad compartida, cortafuegos
21	Proyecto de Red Informática Corporativa para Empresa Comercializadora de Electricidad.	Jorge García Molinero	2015	Español	Seguridad de redes, VLANS, trunk

Nota: Elaboración propia. Recopilado de Tannady, Andry, Sudarsono, & Krishartanto (2020). Mtumbuka (2020). Kim, So, & Je (2019). Mecías Serrano, Molina, & Zúñiga (2019). Guerra (2020). Franco García, (2019). Oroya Acosta (2019). Azarov, Saksonov, & Leokhin (2018). Galarza-Macancela (2018). Saigushev, Mikhailova, Vedeneeva, & Tsaran (2018). Basu (2017). Cedeño Rodríguez (2017). Salih & Alsarhan (2016). Libardo Niño & Vento Serrato (2016). Owens (2016). Hassan, Mondol, & Hasan (2015). (Ben-Asher & Gonzalez, 2015). Miranda Moreira (2015). Verdezoto (2015). Ji, Xing, & Zang, (2015). García Molinero (2015)

3.3.2. Análisis de Requerimientos

Una vez que se ha consolidado la información y se tiene la matriz de datos, se procedió a aplicar una plantilla con los parámetros de la tabla 4 correspondientes a los requerimientos técnicos. Para la selección de criterios de se tomaron en cuenta documentos oficiales como la guía oficial de Cisco capítulo 1 denominada Requisitos de Diseño de Red: Principios de Análisis y Diseño (Community Cisco, 2015) y el artículo "Análisis de requerimientos de Red" publicado por Singh (2019) perteneciente a la Universidad de Melbourne.

En dichos textos se manifiesta que existen requerimientos funcionales mínimos para un correcto desempeño de la red, éstos son: objetivo del negocio, continuidad del negocio, estrategias innovadoras, escalabilidad, diseño de red (tradicional o jerárquico), disponibilidad, seguridad, asequibilidad, segmentación de tráfico, requerimientos de hardware o software, diagramas de topología, mecanismos de convergencia, compatibilidad con tecnologías nuevas, aplicaciones finales de usuario, equipamiento de infraestructura y recurso humano o staff de expertos en TI (Singh, 2019).

Tabla 4.
Criterios de selección para especificación de requerimientos.

Item	Criterio	Definición conceptual
1	Objetivo de negocio	El propósito más amplio de una empresa, ¿a qué se dedica la organización?
2	Escalabilidad	Capacidad de una empresa para aceptar un mayor volumen sin afectar el margen de contribución.
3	Diseño de red	Modelo o diseño de red, puede ser tradicional o jerárquico.
4	Disponibilidad	Cantidad de tiempo de actividad en un sistema de red durante un intervalo de tiempo específico.
5	Segmentación de tráfico	Control de flujo de tráfico para mejorar el rendimiento y la seguridad de la red.
6	Mecanismos de convergencia	Determinan cómo los datos llegan a su destino y ayudan a que ese proceso sea lo más fluido posible, pueden ser: protocolo de información de enrutamiento (RIP), Protocolo

		de puerta de enlace interior (IGRP), Protocolo de enrutamiento de puerta de enlace interior mejorado (EIGRP), Protocolo de puerta de enlace de borde (BGP), entre otros.
7	Topología de red	La forma en que se organiza una red, incluida la descripción física o lógica de cómo se configuran los enlaces y los nodos para relacionarse entre sí.
8	Compatibilidad con nuevas tecnologías	Capacidad de la red para adaptarse a actualizaciones sin modificar significativamente su gestión de cambios.
9	Aplicaciones y servicios	Aplicaciones y servicios asociados a la capa aplicación y el modelo cliente-servidor pudiendo ser: centralizado, distribuido o virtual.
10	Infraestructura de red física	Ubicación de los componentes de la red y los diferentes conectores, se hace referencia a los cables de red físicos.
11	Infraestructura de red lógica	Especifica cómo fluyen los datos dentro de una red.
12	Seguridad	Conjunto de reglas y configuraciones diseñadas para proteger la integridad, confidencialidad y accesibilidad de las redes. Utilización de dispositivos intermedios.
13	Orquestación de red	Gestionar el comportamiento de la red, generalmente por software.
14	Proyección de crecimiento	Responde a la interrogante ¿Se ha planificado el crecimiento de red conforme crece el número de empleados?
15	Recurso humano de TI	Personal que se encarga de la gestión de recursos tecnológicos.

Nota: Elaboración propia. Recopilado de (Singh, 2019).

De esta manera, se procedió a la revisión bibliográfica detallada en cada uno de los documentos tomados en cuenta, bajo la lógica de que, entre más información proporcione el documento, mejores serán los resultados que se obtengan. Con el objetivo de presentar la información de manera organizada y para evitar ambigüedades, se presenta la matriz de la tabla 5 la cual contiene el resumen de los documentos bibliográficos tomados en cuenta especificando las métricas que cumple cada uno.

Tabla 5.
Criterios técnicos de selección

Título del artículo, tesis o publicación / Criterio	**Objetivo del negocio**	**Escalabilidad**	**Diseño de red**	**Disponibilidad**	**Segmentación de tráfico**	**Mecanismos de convergencia**	**Topología de red**	**Compatibilidad con tecnologías**	**Aplicaciones y servicios**	**Infraestructura física**	**Infraestructura Lógica**	**Seguridad**	**Orquestación de red**	**Proyección de crecimiento**	**Recurso Humano**
Arquitectura empresarial con Zachman Framework en una Compañía de Manufactura.	X	-	-	-	-	-	-	-	X	X	X	-	-	X	X
Impacto de la computación en la nube en cuanto a coste y seguridad. Caso de uso: Vodacom Tanzania Public Limited Company.	X	X	-	X	-	-	-	-	-	-	-	X	-	X	-
Un marco de big data para la seguridad de la red de pequeñas y medianas empresas para la informática del futuro.	-	-	-	-	-	X	-	-	-	X	X	X	-	X	X
Diseño de una red de datos para el mejoramiento de la gestión de comunicación interna en UNIANDES Quevedo.	X	X	X	-	-	X	X	X	X	X	X	-	X	X	X
Diseño e Implementación de la red de datos del laboratorio centro de desarrollo de software y productos IOT de la facultad de ingeniería de la Universidad Católica de Santiago de Guayaquil.	X	X	X	-	X	X	X	-	X	X	X	X	-	-	X

Título del artículo, tesis o publicación / Criterio	Objetivo del negocio	Escalabilidad	Diseño de red	Disponibilidad	Segmentación de tráfico	Mecanismos de convergencia	Topología de red	Compatibilidad con tecnologías	Aplicaciones y servicios	Infraestructura física	Infraestructura Lógica	Seguridad	Orquestación de red	Proyección de crecimiento	Recurso Humano
Diseño de la infraestructura de redes para la mejora de la comunicación de datos en la empresa SEAFROST fundamentado en la norma TIA/EIA-942A.	X	-	X	-	X	-	X	-	X	X	X	-	-	-	-
Rediseño de la Red LAN en la Empresa VLACAR S.A.C	X	-	X	-	-	-	X	-	-	X	X	-	-	-	X
Análisis de la estructura de Información del Corporativo de Redes Empresariales.	-	-	-	-	-	-	-	-	-	-	X	X	-	X	-
Diseño e implementación de una red de datos segura para la Pontificia Universidad Católica del Ecuador, Santo Domingo.	X	-	X	-	X	-	X	-	-	X	X	X	-	-	-
Sistemas de Información en la Empresa. Diseño de Red Empresarial Segura.	-	-	X	-	X	X	-	-	X	X	X	X	-	-	-
Evaluación de redes de centros de datos y futuros caminos.	X	X	X	X	X	X	X	X	-	X	X	X	-	X	-

Título del artículo, tesis o publicación / Criterio	Objetivo del negocio	Escalabilidad	Diseño de red	Disponibilidad	Segmentación de tráfico	Mecanismos de convergencia	Topología de red	Compatibilidad con tecnologías	Aplicaciones y servicios	Infraestructura física	Infraestructura Lógica	Seguridad	Orquestación de red	Proyección de crecimiento	Recurso Humano
Evaluación de la Infraestructura Tecnológica basado en Estándares de Control Interno Caso: Empresa National Tire Experts S.A.	X	X	X	X	X	-	X	-	X	X	X	X	-	X	X
Diseño de redes informáticas para universidades en Países en desarrollo.	-	X	X	-	-	-	X	-	-	X	X	X	X	-	-
Diseño de una Red LAN para la Empresa INTEL CORP.	X	-	X	-	X	X	X	X	X	X	X	X	X	-	-
Suministro de Calidad de Servicio en tiempo real para video Interactivo sobre Redes Definidas Por Software.	-	X	X	-	X	-	X	X	X	-	X	-	X	-	-
Diseño de redes informáticas empresariales: una forma simple.	-	X	X	-	-	-	X	-	-	X	X	-	-	X	-
Efectos del conocimiento de la seguridad cibernética en la detección de ataques.	-	-	-	-	X	-	-	-	X	-	X	X	-	-	-
Diseño e Implementación del Esquema de Seguridad Perimetral y Monitoreo para la Red de Datos en una Empresa Industrial.	X	-	-	-	X	-	-	-	X	X	X	X	-	-	-

Título del artículo, tesis o publicación / Criterio	Objetivo del negocio	Escalabilidad	Diseño de red	Disponibilidad	Segmentación de tráfico	Mecanismos de convergencia	Topología de red	Compatibilidad con tecnologías	Aplicaciones y servicios	Infraestructura física	Infraestructura Lógica	Seguridad	Orquestación de red	Proyección de crecimiento	Recurso Humano
Diseño de una Red LAN para el transporte de voz, datos y video para el Municipio del Cantón Valencia Provincia de Los Ríos.	X	-	X	-	X	X	X	-	X	X	X	X	-	X	-
Diseño de un esquema de intercambio de información de red empresarial basado en tecnología de seguridad.	-	-	X	-	X	X	-	-	X	X	X	X	X	-	-
Proyecto de Red Informática Corporativa para Empresa Comercializadora de Electricidad.	X	-	X	X	-	-	X	-	X	X	X	X	X	X	X

Nota: Elaboración propia. Recopilado de Tannady, Andry, Sudarsono, & Krishartanto (2020). Mtumbuka (2020). Kim, So, & Je (2019). Mecías Serrano, Molina, & Zúñiga (2019). Guerra (2020). Franco García, (2019). Oroya Acosta (2019). Azarov, Saksonov, & Leokhin (2018). Galarza-Macancela (2018). Saigushev, Mikhailova, Vedeneeva, & Tsaran (2018). Basu (2017). Cedeño Rodríguez (2017). Salih & Alsarhan (2016). Libardo Niño & Vento Serrato (2016). Owens (2016). Hassan, Mondol, & Hasan (2015). (Ben-Asher & Gonzalez, 2015). Miranda Moreira (2015). Verdezoto (2015). Ji, Xing, & Zang, (2015). García Molinero (2015)

3.3.3. Especificación de Requerimientos SDN/NFV

El análisis de requerimientos permitió identificar el estado actual de las redes convencionales, protocolos, servicios, aplicaciones y gestión, entre otros, así como también el camino que siguieron grandes de la industria para llegar con éxito a la tecnología de las redes definidas por software, por lo que se procede a consolidar estas características a través de los siguientes puntos:

Alcance del Diseño de Red.

Es importante analizar el alcance del diseño, evaluando el punto de partida la red, la investigación arrojó que las empresas "más jóvenes" sí consideran que su red de datos debe ser escalable por lo que se encontrarían en un estado "greenfield" partiendo de que, es más fácil construir algo nuevo que reparar lo que ya está, o, si por el contrario se trata de una red de producción en marcha y hasta dónde se quiere llegar, por supuesto para establecer esto, la empresa debe considerar, costos, tiempo, infraestructura actual y el nivel de experticia del su personal de TI.

Requerimientos de Negocio.

Otro aspecto importante es conocer las necesidades del negocio puesto que como se manifiesta en Cisco Analysis and Design Principles, estos aspectos pueden influir en el diseño de red ya sea de forma directa o indirecta, para ello se observa el panorama empresarial, sus objetivos, su visión y direcciones futuras (Community Cisco, 2015). Éstos son:

- Expansión del negocio (añadir más sucursales).
- Reducir costos operacionales.
- Mejorar la productividad de los empleados.
- Reducir los costos de mantenimiento de hardware.

Crecimiento Futuro de la Organización.

Se refiere a la flexibilidad que el diseño de red pueda proporcionar, como lo indica el estudio de Mecías Serrano et al, (2019) refiriéndose a un campus educativo que consta de cinco edificios pero en los próximos años se tiene previsto la incorporación de una facultad más o el caso de VLACAR, empresa que planea implementar más sucursales (Oroya Acosta, 2019); el diseño debe ser capaz de soportar esta expansión, en términos de hardware, plano de control y orquestación.

Requerimientos Funcionales.

Se describen como aquellos que una infraestructura de red deme suministrar en términos de flexibilidad y escalabilidad. Así se tiene:

- Una infraestructura que soporte voz, video, datos y comunicación inalámbrica, primordiales servicios que utilizan las compañías a la fecha.
- Aislamiento del tráfico generado por invitados, o usuarios internos a fin de evitar la saturación de ancho de banda.
- Capacidad de integración de sitios remotos o equipos sin necesidad de hacer un rediseño de red.
- Proporcionar mecanismos de protección como listas de acceso.

Requerimientos de Aplicaciones.

Para todas las empresas investigadas independientemente de su núcleo de negocio, la experiencia del usuario final es una las principales prioridades que cualquier Departamento de Tecnologías y diseño de red debe cubrir, así se proponen las siguientes categorías:

- Clientes: pueden ser individuos, o colectivos.
- Usuarios internos: personal que infiere en la productividad de la empresa, tienen una relación directa con el éxito empresarial.

- Business Partners: socios que trabajan en equipo para conseguir ciertos objetivos.

Luego se podrá contrastar con las aplicaciones que se necesiten, en relación al alcance planteado, para lo cual se requiere conocer, la prioridad de las aplicaciones del negocio, como por ejemplo VoIp.

Topología de Red.

Con base en la arquitectura actual de red, se debe proponer un diseño compatible con el entorno; no es posible dar un solo salto a una red SDN/NFV, primero se debe llegar a un escenario híbrido: una arquitectura de red convencional que soporte redes definidas por software. De esta forma, las topologías utilizadas por éstas organizaciones son:

- Topología jerárquica.
- Topología tipo árbol.

Requerimientos Técnicos.

Para satisfacer las necesidades planteadas y cumplir con los objetivos empresariales, se presentan los requerimientos técnicos en la tabla 6, siendo posible asignarles una prioridad dependiendo de cada empresa.

Tabla 6.
Requerimientos técnicos de diseño SDN/NFV

Item	Requerimiento técnico	Característica
1	Disponibilidad	Rápida convergencia o nodos redundantes
2	Calidad de servicio	Prioridad de tráfico (QoS)
3	Seguridad	Dispositivos intermedios (firewall)
4	Aislamiento de tráfico	Técnicas de virtualización de funciones de red.
5	Escalabilidad	Diseño de hardware escalable.

Nota: Tomado de Cisco Analysis and Design Principles (Community Cisco, n.d.)

Cuadro Resumen de Requerimientos de Diseño SDN/NFV.

Como se mostró en la figura 1, una arquitectura definida por software se divide en tres capas:

- Capa de infraestructura. (NFVI)

- Capa de control, y;

- Capa de aplicación

Por lo que los requerimientos de diseño de red acorde a esta arquitectura, se definen en la figura 11:

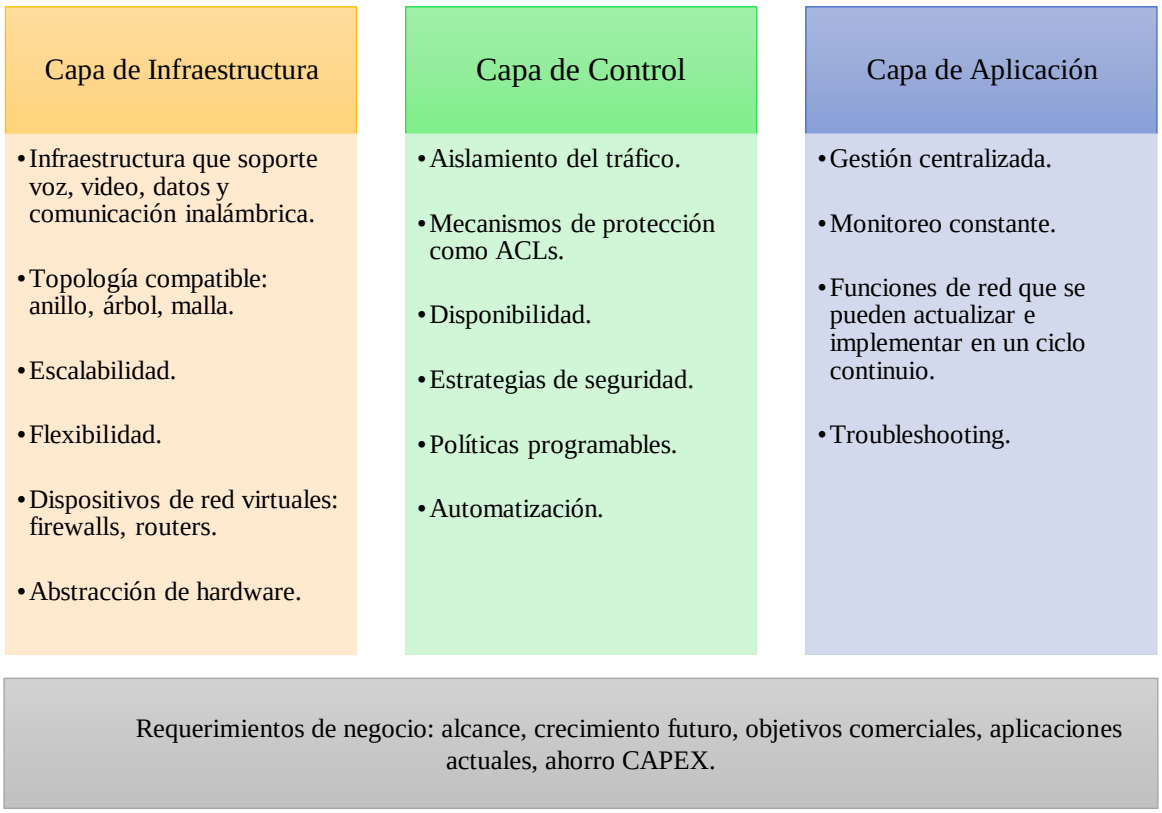


Figura 11. ***Requerimientos de diseño SDN/NFV.***
Nota: Elaboración propia.

En conclusión, para migrar a una red SDN/NFV se debe partir de las necesidades comerciales del negocio, priorizando un diseño de red que le agregue valor a la empresa y se alinee con sus objetivos, para esto se consideran las brechas o puntos débiles actuales tanto en infraestructura como en procesos; la elicitación, análisis y especificación de requerimientos permitirán sustentar la metodología a realizar puesto que son las las características base para el adecuado funcionamiento de red a fin de minimizar o evitar interrupciones de servicios durante la transición y por supuesto que la red se catalogue como un "facilitador de negocios".

El desarrollo de esta metodología, además de tomar en cuenta las buenas prácticas del estándar ETSI GS NFV-PER 001 y las propuestas por el grupo ONF (Open Networking Foundation) en el paper "Migration Use Cases and Methods", se fundamenta en las metodologías estudiadas en el capítulo de estado del arte: metodología PPDIOO, la arquitectura de seguridad CISCO SAFE propuesta por Cisco y la matriz de referencia COBIT 5, debido a que son las más utilizadas por expertos al momento de diseñar redes.

3.4.Metodología de Transición

Descripción General

El objetivo principal de esta metodología es sistematizar las técnicas necesarias para llevar a cabo una transición a redes inteligentes partiendo de una infraestructura existente; uno de los atributos de la intersección SDN/NFV es la retro - compatibilidad con redes legacy, por lo que en la figura 12 se proponen las siguientes fases:

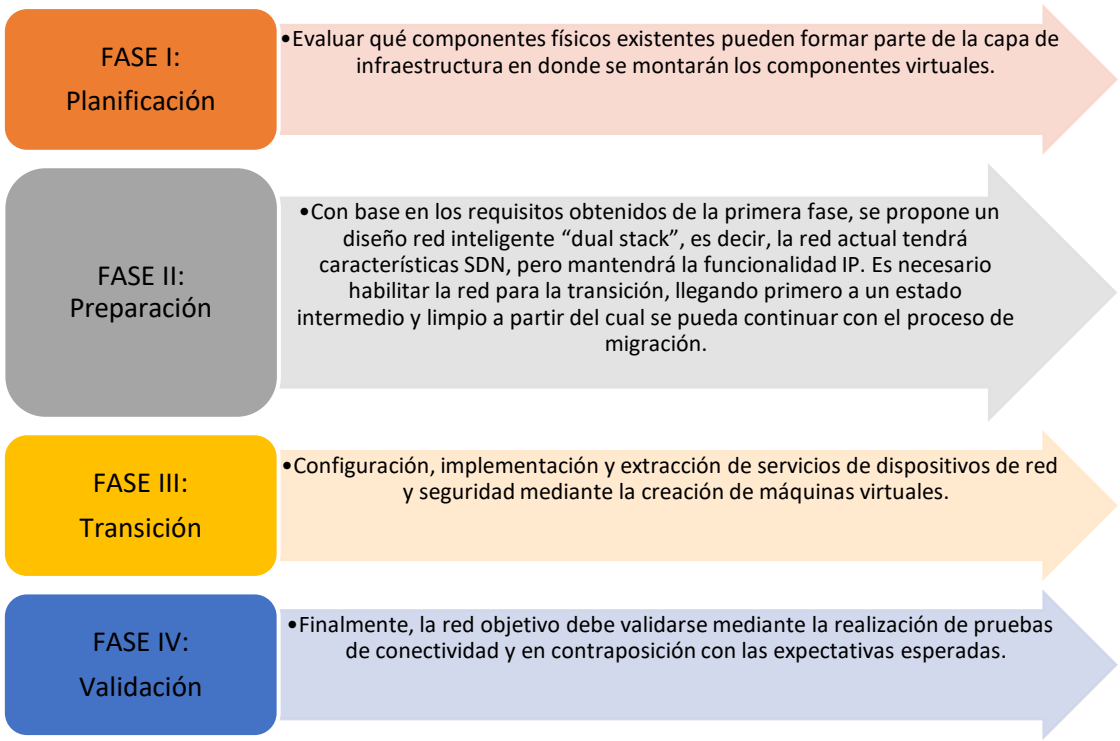


Figura 12. ***Metodología de transición SDN/NFV.***
Nota: Elaboración propia.

Fase I: Planificación

La planificación corresponde al aspecto más crítico de la transición, ya que se necesita descubrir y mapear los flujos de conectividad de red, se deben priorizar los servicios y aplicaciones, identificando cada equipo, se recomienda fundamentarse en bitácoras o registros que indiquen la configuración de red actual. Las organizaciones disciplinadas que tengan registros actualizados y precisos de esta información podrán importar su documentación y avanzar rápidamente a la siguiente etapa. Se proponen las estrategias siguientes:

- **Starting Network.** Corresponde al punto de partida, para lo cual es necesario contar con toda la documentación referente a topología, equipos, hardware, software, protocolos de enrutamiento, protocolos de switching, y herramientas existentes. La tabla 7 muestra los parámetros relevantes:

Tabla 7.
Descripción de la red inicial

Item	Parámetro	Descripción
1	Descripción general	Indicar el propósito de red, sus servicios principales y objetivos comerciales.
2	Modo operacional	Protocolos que se utilizan en la red, por ejemplo BGP.
3	Hardware	Inventario del equipo desplegado en la red interna para identificar qué equipo puede soportar actualizaciones-
4	Redundancia	Mencionar si existe un modelo de redundancia y en qué capas se enfoca.
5	Herramientas de gestión	Mostrar las herramientas de gestión que se utilizan para supervisar, monitorear y realizar troubleshooting.
6	Capacidad de red	Breve descripción del número de usuarios o tamaño general de red.
7	Disponibilidad	Cuál es la disponibilidad de servicios sobre la red actual, ¿Cuáles son los cortes tolerados del servicio?

Nota: Elaboración propia.

- **Análisis de Brechas**. Identificar el impacto que puede tener la transición en los servicios existentes con el objetivo de buscar alternativas que mitiguen los desafíos que pueden presentarse durante el proceso de migración. Se sugiere utilizar la metodología para identificación de cuellos de botella que propone el estándar GS NFV-PER 001 v1.1.1 que sugiere analizar el procesamiento de paquetes en la red "bare" para luego compararlo con el entorno virtualizado.

  Debido a que la red se encuentra en el ambiente de producción no será necesario generar tráfico, pues este ya está en marcha, y bastaría con realizar las pruebas y documentar los resultados.
- **Check Lists.** Realizar listas de verificación previas y posteriores a la transición, éstas se utilizarán para las comprobaciones de conectividad y continuidad del servicio; con esto es posible que existan debilidades previas, por lo que serán excluidas como problemas a causa de la migración.
- **Procedimientos de Back-out.** Es importante llevar un registro paso a paso de cada modificación que se realice tanto a nivel de hardware como de software, por más irrelevante que parezca, esto permitirá minimizar la interrupción de los servicios en caso de fallos, así como reversar las configuraciones si no se cumplieron las expectativas.
- **Capa de infraestructura.** En este punto ya es posible determinar qué elementos de red pueden formar parte de la capa de infraestructura de redes definidas por software.

Fase II: Preparación

Consiste en preparar la red para la transición, dejándola en un estado "bare", como indica el estándar ETSI GS NFV-PER 001, se puede ayudar con la utilización de:

- **Herramientas de Aprovisionamiento.** Que permitan monitorear el tráfico durante y después de la transición.
- **Capa de virtualización.** Con base en la información recabada en la fase de planificación ya se puede realizar la elección del hipervisor, alineando la infraestructura con la plataforma SDN adecuada. Esta investigación propone la utilización de software libre, en la tabla 8 se realiza una comparación con las alternativas sugeridas.

Tabla 8.
Comparación de los tipos de hipervisores de código abierto

Item	Hipervisor	Compañía	Modelo	Hardware asistido	Sistema Operativo
1	KVM	Red Hat	Alojado	No	No
2	XEN	Citrix system, inc	Bare Metal	Sí	No
3	OpenVZ Linux	OpenVZ	Alojado	No	Sí
4	Linux-Vserver	Linux-Vserver	Alojado	No	Sí
5	Proxmox VE	Proxmox	Bare Metal	No	Sí

Nota: Tomado de A Study On Virtualization Techniques And Challenges In Cloud Computing (Durairaj & Kannan, 2014)

- **Elección del Controlador SDN.** En una red definida por software el controlador SDN es el "cerebro" de la red, siendo un punto estratégico de la arquitectura. Este trabajo propone la utilización de controladores SDN de código abierto, en la tabla 9 se puede observar la variedad que existen.

Tabla 9.
Comparación de los tipos de controladores de código abierto

Item	Controlador SDN	Lenguaje de programación	GUI	Plataforma que soporta	Southbound APIs	Northbound APIs	Partner
1	ONOS	java	web	Linux, MAC OS, Windows	OF1.0, 1.3, NETCONF	REST API	ON.LAB,AT&T, Ciena, Cisco, Ericsson, Fujitsu, Huawei, Intel, Nec, Nsf.Ntt Comunnication, Sk Telecom
2	Open Daylight	java	web	Linux, MAC OS, Windows	OF1.0, 1.3, NETCONF/YANG, OVSDB,PCEP,BGP/ LS/SNMP	REST API	Linux Foundation With Memberships Covering Over 40 Companies, Such As Cisco, IBM, NEC
3	NOX	C++	python +QT4	Linux	OF1.0	REST API	Nicira
4	POX	Python	python +QT4	Linux, MAC OS, Windows	OF1.0	REST API	Nicira
5	RYU	Python	web	Linux	OF1.0, 1.2, 1.3, 1.4, NETCONF	REST API	Nippo Telegraph And Telephone Corporation
6	Beacon	java	web	Linux, MAC OS, Windows	OF1.0	REST API	Stanford University
7	Flood light	java	web/java	Linux, MAC OS, Windows	OF1.0, 1.3	REST API	Big Switch Networks
8	Iris	java	web	Linux, MAC OS, Windows	OF1.0, 1.3, OVSDB	REST API	ETRI

Nota: Tomado de SDN Controllers: A Comparative Study (Salman, Elhajj, Kayssi, & Chehab, 2016)

En publicación realizada por Salman, Elhajj, Kayssi, & Chehab titulada “SDN Controllers: A Comparative Study”, se realiza una comparación de diversos controladores basados en múltiples criterios y pruebas de rendimiento, concluyendo

que OpenDaylight es una buena opción ya que soporta una amplia gama de funciones e integración de componentes en su interfaz Dlux.

- **Servidores a migrar:** identificar el almacenamiento requerido, el CPU y el sistema operativo.
- **Control de Versión de OpenFlow.** Existen varias versiones del protocolo OpenFlow por lo que se debe verificar que las versiones de los equipos actuales sean compatibles con el controlador SDN que se vaya a implementar.
- **Actualizaciones.** Para homologar y garantizar la compatibilidad con OpenFlow es posible que ciertos equipos requieran actualización de su firmware.

  En este punto, los elementos de red se volverán híbridos por lo que puede ser necesario añadir paquetes de software OpenFlow. Se llega a un escenario de red "doble pila".
- **Planear una Estrategia de Seguridad.** Es importante contar con procedimientos que proporcionen seguridad la infraestructura SDN/NFV; en el libro titulado "Security Analysis of Software-Defined Networking and Network Function Virtualization" de Artmann & Khondoker se proponen los lineamientos que debe tener esta arquitectura, de los cuales se sugieren utilizar:
  a. **Seguridad en dispositivos intermedios:** utilizar dispositivos de seguridad intermedios y configurar calidad de servicio para una optimización del performance de red.
  b. **Seguridad en el controlador SDN:** al tener una gestión centralizada, se obtiene también un único punto de falla por lo que mínimo se deben tomar en cuenta las mejores prácticas que se describen en la figura 13:

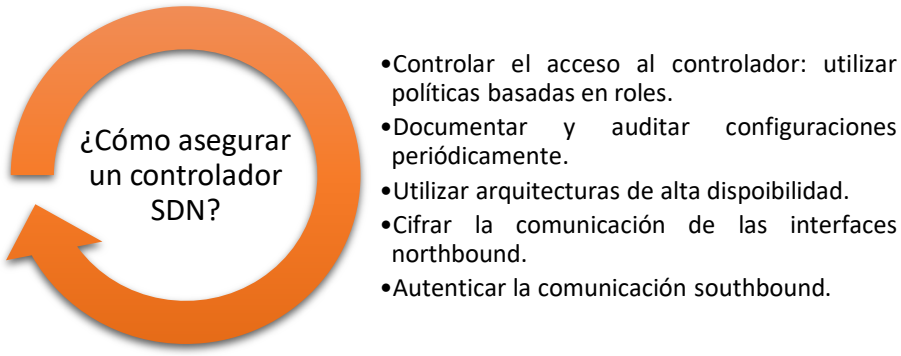


Figura 13. ***Mecanismos de seguridad en un controlador SDN. Security Analysis of Software-Defined Networking and Network Function Virtualization***
Nota: Tomado de (Artmann & Khondoker, 2018)

Fase III: Transición

En esta fase, las organizaciones empezarán con el proceso de cambios, contando con los procedimientos de back-out descritos anteriormente que respalden las modificaciones SDN y de infraestructura. Los plazos de transición pueden variar dependiendo de varios factores, como por ejemplo el tamaño y la complejidad de la infraestructura de red.

Esta metodología propone un enfoque incremental, se tiene claro que no se pueden migrar todas las aplicaciones a la vez, por lo que, se construye según las etapas:

- **Planteamiento de Políticas.** Se deberán crear políticas con definiciones de alto nivel, que, a través de un compilador de políticas, que pueden ser las mismas APIs del controlador serán traducidas a reglas de bajo nivel acorde a la tecnología subyacente como se muestra en la figura 14, éstas deben expresar qué es lo que se permite y que no, representando la diversidad de recursos y operaciones.

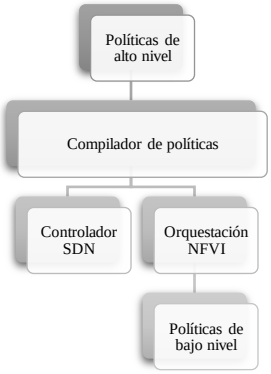


Figura 14. ***Políticas SDN/NFV. Guide to security in SDN and NFV.***
Nota: Tomado de (Shao Ying Zhu, Sandra Scott-Hayward, Ludovic Jacquin, 2017)

- **Configuración de Nuevos Servidores:** con la información de la fase de preparación se pondrán en marcha los nuevos servidores que estarán "en blanco" y se montarán sobre la capa de virtualización en el hipervisor seleccionado.
- **Disponibilidad de Servicios:** se sugiere crear un servicio ficticio, como por ejemplo una VPN a fin de constatar su disponibilidad y realizar pruebas de funcionamiento antes de poner el resto de las aplicaciones o servicios en el ambiente de producción.
- **Migrar las Aplicaciones:** mover las aplicaciones del ambiente de pruebas a producción.
- **Redireccionar los Hosts de Usuario Final:** aun cuando todo el proceso es transparente para los usuarios, es posible que se tenga que apuntar a la nueva dirección IP de un servidor o puerto.

Fase IV: Validación

El siguiente paso posterior a la transición de arquitecturas convencionales a redes definidas por software, es la gestión continua de las políticas implementadas; para ello se necesita estar en constante monitoreo y auditoría de cambios, se sabe que las aplicaciones

comerciales varían con el tiempo, por tanto, los administradores deberán también actualizar las políticas de red y verificar:

- **Conectividad.** La conectividad entre el controlador SDN y los dispositivos OpenFlow debe verificarse.
- **Troubleshooting.** Se pueden emplear mecanismos de resolución de problemas como ping, trazas, entre otros.

Métricas de las Redes Definidas por Software

La mayoría de las métricas con las que se evalúan las redes tradicionales también son aplicables para las SDN, como por ejemplo: razón de pérdida de paquetes, latencia o tiempos de activación de un servicio; sin embargo existen factores propios de las redes definidas por software relacionadas con OpenFlow: tiempo de activación del protocolo, conmutación, establecimiento de flujos, comunicación entre el controlador y los switches; y, escalabilidad, refiriéndose al proceso de agregar más switches y analizar cuantos puede soportar el controlador SDN sin degradar su rendimiento.

Resumen

Para llevar a cabo un procedimiento exitoso de transición de redes definidas por software es importante responder algunas interrogantes, como, ¿Para qué se desea migrar?, ¿Cuáles son las opciones de migración?, ¿Cómo han llevado acabo la transición organizaciones semejantes?, y probablemente la más importante es ¿Cuál es el propósito de migrar a esta tecnología? Responder estas cuestiones resulta clave para mitigar el riesgo de fallos durante el procedimiento y por supuesto, optimizar tiempos.

Para finalizar, se puede mencionar que la virtualización es una tecnología que ya se ha establecido hace mucho tiempo, sin embargo, explotando sus mejores técnicas y buenas

prácticas, es posible agregar una capa de containers, transformando la capa de aplicación en una plataforma bastante ágil capaz de virtualizar firewalls, routers y otros dispositivos de red.

Lo siguiente consiste en agregar agilidad a la capa de funciones de red a través de SDN para administrar los dispositivos de red y llegar a la automatización. Con esta metodología se pretende optimizar los servicios de TI en una infraestructura altamente receptiva y resistente.

3.5. Consideraciones Bioéticas

Toda la información recopilada a lo largo del desarrollo de esta investigación es de tipo confidencial, para uso exclusivo del investigador, y bajo los fines pertinentes, protegiendo la autonomía de los implicados y de los datos extraídos.

Asimismo, el contenido de la investigación se apega al concepto de ahorro energético pues se pretende minimizar el uso de hardware dedicado, que a largo plazo se convierten en basura electrónica, siendo una bomba ecológica para el planeta.

Capítulo IV

Resultados y Discusión

En este apartado, se procedió a validar la metodología propuesta en el capítulo IV, para ello se propusieron tres escenarios prototipo que fueron planteados con base en el análisis de requerimientos, que indican cómo se encuentran estructuradas actualmente las redes de diversas empresas, los ambientes que se simularon se asemejan a los entornos productivos reales de las redes tradicionales. En el primer escenario se planteó un esquema de red tradicional, en donde sus servicios siguen siendo locales y administrados manualmente, corresponde un escenario de red en crecimiento; el segundo corresponde a un ambiente de red más estructurado y con redundancia; y por último el tercer caso constituye un entorno virtualizado con una combinación de servicios en la nube.

En la sección 3.4 de la metodología, fase II tabla 9, se describe como procedimiento la elección del controlador SDN, no obstante, para las emulaciones que se presentan a continuación se omitirá este paso, pues se ha seleccionado el controlador OpenDaylight destacando su compatibilidad con grandes marcas como Cisco, Citrix, IBM, Juniper, RedHat, VMware, así como también las ventajas que presenta sobre otros controladores de su tipo, como es el de ofrecer políticas basadas en grupos (Group Based Policies) configurables a través de su API REST, que es un concepto heredado de Cisco, en el cual se tiene una configuración mucho más flexible que la metodología de “intents” que presentan sus equivalentes. La instalación de ODL se presenta en el Anexo C, al igual que mininet como software de emulación cuyo proceso de instalación está en el Anexo B.

4.1.Primer Escenario: Esquema Greenfield

Corresponde a un escenario de red netamente tradicional, sus equipos funcionan con protocolos de enrutamiento dinámicos y se tiene una convergencia exitosa, este tipo de red no

tiene inteligencia alguna, ya que el administrador se encarga de las configuraciones manuales en cada dispositivo.

Para la aplicación de la metodología en este primer caso, se denominó a la organización como “Empresa X”, dedicada a la comercialización de productos, que consta de una oficina matriz donde se localiza el datacenter con servidores de propósito particular y sucursales que también cuentan con equipos de red y servicios independientes, formando una topología tipo árbol. Para las configuraciones y pruebas que se llevaron a cabo, se hizo uso de Mininet como herramienta de simulación de la capa de infraestructura, OpenDaylight como controlador SDN, el cual se levantó sobre el dispositivo anfitrión implementando los módulos mediante la línea consola de karaf.

4.1.1. Aplicación Fase I

A continuación, se empieza a aplicar la metodología por fases. En la **primera fase** correspondiente a la planificación, se recopiló toda la información posible acerca de la red existente: servicios, protocolos, equipos, aplicaciones y demás. La metodología sugiere indicar cuál es el target del negocio, responder las interrogantes ¿Para qué quiero llevar a cabo la transición?, ¿Cuál es mi inventario de hardware actual?

Para el planteamiento de este escenario de red se hizo uso de documentación bibliográfica actualizada y la tabla resumen que se expone en el apartado 3.1. Este primer esquema responde a las características de los tres primeros entornos corporativos que se indican en la tabla 2 de la misma sección de requerimientos. Así, en la tabla 10 se muestra la descripción general de la Empresa X, mientras que la tabla 11 muestra los componentes de red, siendo bastante similares a las características reales de organizaciones.

Tabla 10.
Descripción de la red inicial de la Empresa X.

Item	Parámetro	Descripción
1	Target	Comercialización de productos, cuenta con servidores de aplicaciones para su sistema interno.
2	Modo operacional	Utiliza BGP y OSPF como protocolos de enrutamiento. Añade MPLS.
3	Redundancia	No existe redundancia a nivel de capa física.
4	Herramientas de gestión	Monitoreo y administración manual de red.
5	Disponibilidad	Es una empresa de ventas cuyo sistema principal debe estar activo durante las horas comerciales. Se tiene ventanas para troubleshooting en la noche.

Nota: Elaboración propia.

Tabla 11.
Componentes de red de la Empresa X

Item	Servicio / Componente	Descripción
1	Número de hosts de usuario	150
2	Topología de red	Tipo árbol
3	Ancho de banda	20Mbps
4	Firewall	Router con funcionalidades de firewall
5	Servidor de correo	Zimbra / Centos 7
6	Servidor web	Apache
7	Servidor de aplicaciones	Sistema de información interno.
8	Switches	Capa 2 marca Cisco
9	Routers	Capa 3 marca Cisco
10	Protocolos de comunicación	BGP, OSPF, MPLS

Nota: Elaboración propia.

En la figura 15 se planteó el diseño de red actual con sus componentes físicos y lógicos, está conformada por los dispositivos de enrutamiento que convergen gracias a OSPF y BGP, y además, se ha implementado Multiprotocol Label Switching para lograr una

convergencia más rápida debido a que MPLS se basa en el intercambio de etiquetas, en lugar de realizar búsquedas dentro de las tablas ip optimizando el rendimiento de red y priorizando paquetes (E. Rosen; Cisco Systems, Inc; A. Viswanathan; Force10 Networks, 2001). Se tiene una oficina matriz en donde se alojan los servidores y a su vez se conectan los usuarios. Esto se logra gracias a un switch core con funcionalidades de capa tres.

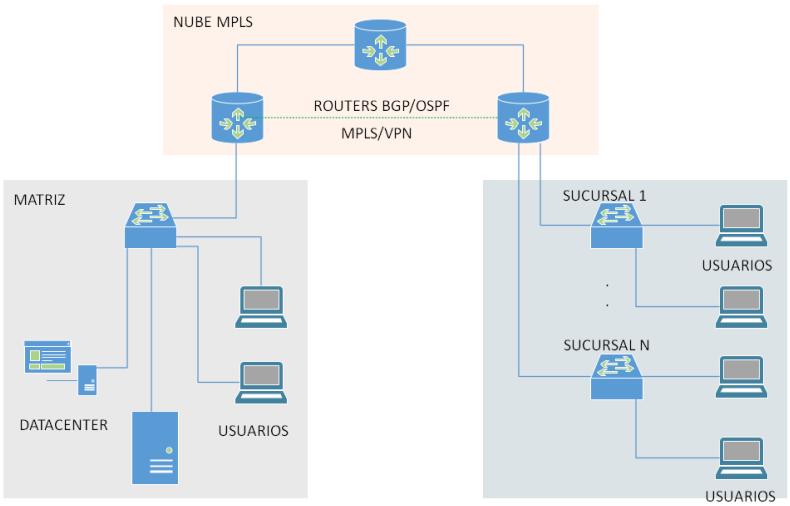


Figura 15. ***Topología primer caso: esquema de red Greenfield.***
Nota: Elaboración propia.

A fin de obtener métricas de comparación cuando se realice la transición se procedió a simular el esquema de red en GNS3, software que permite levantar diversos escenarios de redes ip emulando el comportamiento real del hardware de red (Gns3, 2020). La metodología sugiere tomar valores referenciales de métricas antes de iniciar el proceso de transición, para establecer comparativas más adelante. En el Anexo F, parte F.1) se encuentran las pruebas de conectividad realizadas en la herramienta GNS3.

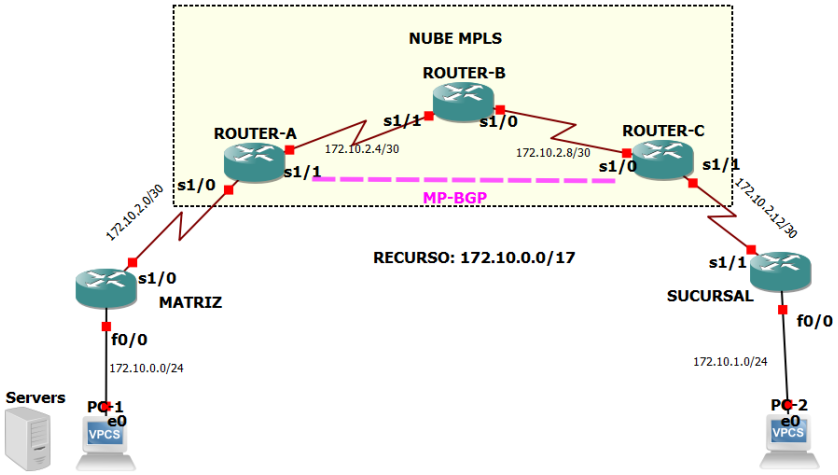


Figura 16. ***Topología tradicional primer caso en GNS3***
Nota: Elaboración propia.

La topología propuesta se puede observar en la figura 16, para la cual se utilizaron routers cisco en su serie 7200, utilizando OSPF como protocolo de enrutamiento en la nube MPLS y BGP en los routers de borde. En la tabla 12 se indica el direccionamiento ip que se utilizó:

Tabla 12.
Direccionamiento red tradicional esquema Greenfield en GNS3

Dispositivo	Interfaz	Dirección IP	Máscara de subred	Gateway predeterminado
RA	S1/0	172.10.2.2	255.255.255.252	N/A
	S1/1	172.10.2.5	255.255.255.252	N/A
	Lo0	10.1.1.1	255.255.255.255	N/A
RB	S1/0	172.10.2.9	255.255.255.252	N/A
	S1/1	172.10.2.6	255.255.255.252	N/A
	Lo0	10.2.2.2	255.255.255.255	N/A
RC	S1/0	172.10.2.10	255.255.255.252	N/A
	S1/1	172.10.2.13	255.255.255.252	N/A
	Lo0	10.3.3.3	255.255.255.255	N/A
MATRIZ	S1/0	172.10.2.1	255.255.255.252	N/A
	Fa0/0	172.10.0.1	255.255.255.0	N/A
SUCURSAL	S1/1	172.10.2.14	255.255.255.252	N/A
	Fa0/0	172.10.1.1	255.255.255.0	N/A
Pc1	Eth0	172.10.0.10	255.255.255.0	172.10.0.1
Pc2	Eth0	172.10.1.10	255.255.255.0	172.10.1.1

Nota: Elaboración propia.

Con esta información, se armó un checklist con los puntos clave que se revisarán en las fases posteriores. Así se muestra en la tabla 13 un resumen de lo que se tiene en la red actualmente, que sería lo mínimo que debe cumplir al migrar a la tecnología NFV/SDN.

Tabla 13.
Checklist de red escenario greenfield

Item	Parámetro		Observaciones
1	SEGURIDAD		
	Tiene un firewall para proteger su red interna del acceso no autorizado.	☐	
	Las contraseñas de los dispositivos de red se manejan por roles.	☑	Solamente se tiene el rol de administrador.
	Cada cambio de configuración es documentado y almacenado en una bitácora.	☑	
	Se utilizan solamente canales seguros para acceder a la red.	☑	Las direcciones ip públicas asignadas a los servidores se habilitan o deshabilitan según la necesidad.
	Se deshabilitan reglas de firewall innecesarias.	☑	Proceso muy complejo que se lleva a cabo minuciosamente dado que las políticas se implementan línea por línea mediante el terminal de comandos de cada dispositivo.
2	DISPOSITIVOS DE RED		
	Descarga de firmware, actualizaciones y parches de fuentes validadas.	☑	
	Se tiene una lista de todo el hardware de red, dispositivo, ubicación, número de serie.	☑	
	Se utiliza protocolos de administración como SNMP.	☐	
3	GESTIÓN DE SOFTWARE		
	Uso exclusivo de software compatible, libre o con licencia.	☑	
	Software pirata se elimina de los equipos	☑	
	Uso de redes privadas virtuales para acceso remoto.	☐	

Nota: Elaboración propia.

La metodología indica que en este punto se debe determinar qué elementos serán parte de la capa de infraestructura de la red definida por software. Este primer escenario resulta ser el menos complejo para una transición puesto que no necesita interoperación con dispositivos OpenFlow, que manejen una u otra versión, ya que, más bien se trata de una integración de un controlador OpenFlow al mismo escenario ip. Así se tiene en la figura 17, como en una red tradicional se integra un controlador OpenFlow, volviéndola "inteligente". Inicialmente cada dispositivo de red opera de manera individual, mientras que al añadir el controlador OpenFlow, la gestión se centraliza, teniendo una visión global de la red.

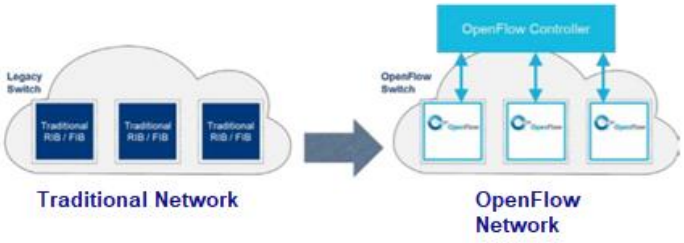


Figura 17. ***Transición de un esquema tradicional a OpenFlow Network***
Nota: Tomado de Open Networking Foundation (2014)

4.1.2. Aplicación Fase II

La **segunda fase** corresponde a la preparación, se trata de dejar la red en un estado "bare", listo para la transición; en este escenario la red ya se encuentra en un estado bare, por lo que únicamente se debe hacer uso de herramientas de aprovisionamiento que permitan monitorear el tráfico durante y después del procedimiento.

Como se mencionó en el inicio de este capítulo, la capa de virtualización estará conformada por un hipervisor alojado (En la tabla 8 se muestran los disponibles OpenSource), para efectos de esta simulación se trabajará con VirtualBox 6.0.22, siendo la más reciente a la fecha, herramienta de hardware asistido que se implementó sobre una máquina con Windows 10 Pro, de 64bits, procesador Intel (R) Core (TM) i7 con 16.0 GB de memoria RAM.

De igual manera en la tabla 9, se tiene la comparación entre controladores OpenFlow de código abierto, para los ejemplos presentados en este trabajo se utilizará OpenDaylight, que continuando con la metodología propuesta se implementará con un nivel de seguridad básico.

4.1.3. Aplicación Fase III

La **tercera fase** es la transición, que se sugiere realizarla en partes e ir testeando los resultados paulatinamente, la capa de infraestructura se emulará sobre la herramienta Mininet.

Diseño de la red definida por software con Mininet

Se procederá a levantar una topología similar a la realizada en GNS3, para lo cual se utilizará la herramienta Miniedit, la cual, al desplegar un panel, permite integrar los componentes de forma gráfica.

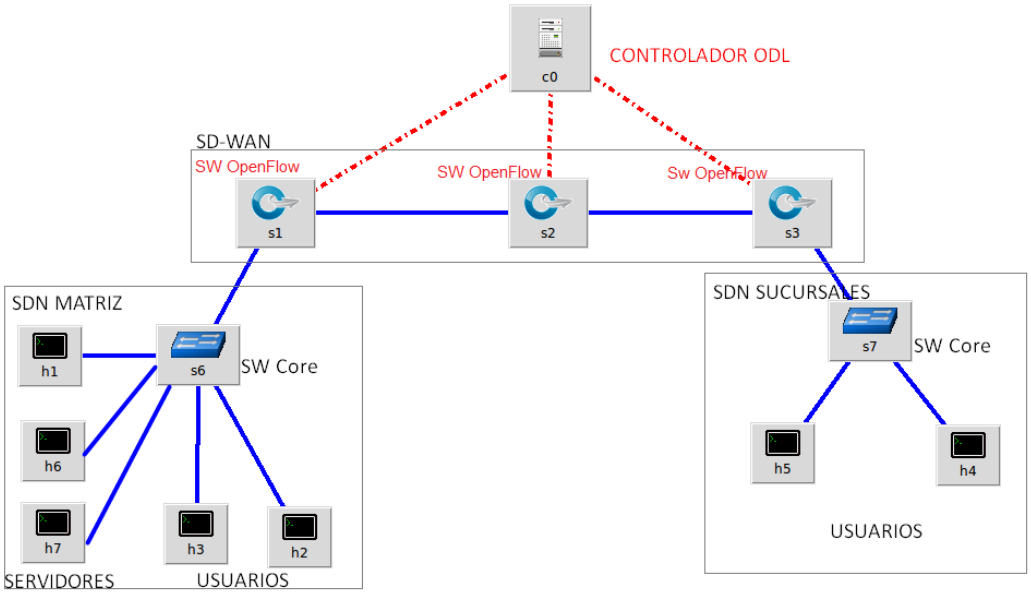


Figura 18. ***Simulación de red SDN en mininet: esquema Greenfield.***
Nota: Elaboración propia

El direccionamiento ip para la topología de la figura 16 es el mismo que se utilizó en la simulación en GNS3, con base en la red 172.10.0.0/17, en la tabla 14 se indican los dispositivos virtuales con la dirección ip asignada:

Tabla 14.
Direccionamiento red sdn esquema Greenfield en Mininet

Dispositivo virtual	Direccionamiento IP
Controlador OpenDaylight	172.10.10.100
Router A	172.10.2.5
Router B	172.10.2.9
Router C	172.10.2.13
MATRIZ	172.10.0.1
SUCURSAL	172.10.1.1
h1	172.10.0.10
h2	172.10.0.20
h3	172.10.0.30
h4	172.10.1.10
h5	172.10.1.20

Nota: Elaboración propia.

Para crear la topología en miniedit, en primer lugar, se deben realizar algunas configuraciones previas a través de “preferencias”, llenando el campo IP Base con la dirección de red a utilizar, marcar la versión de openflow y seleccionar “start CLI” para obtener un terminal en cada host. La figura 19 muestra los parámetros seleccionados:

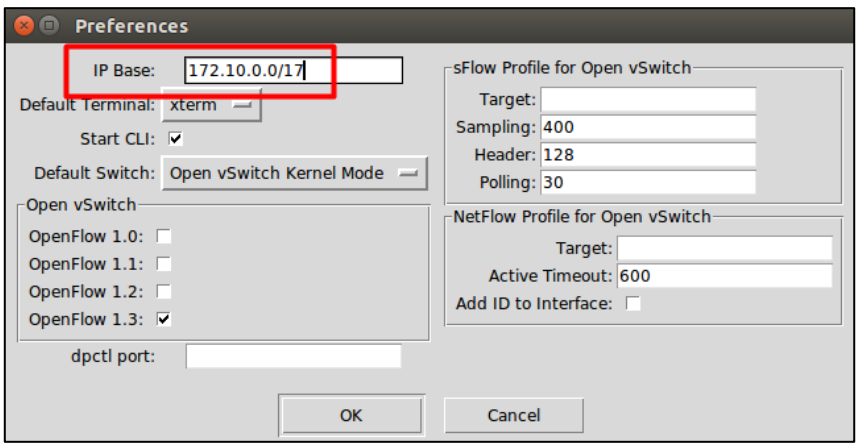


Figura 19. ***Recurso de red utilizado en mininet***
Nota: Elaboración propia

Lo siguiente será configurar las direcciones ip de los dispositivos, y para ejecutar el escenario creado, se presiona “run”, pudiendo exportar la topología a un formato en Python, el script exportado en formato .py se encuentra en el Anexo F, parte F.3), en el cual ya se indica que se está apuntando a la dirección ip del Controlador ODL.

Ahora basta con ejecutar el comando pingall en mininet para que se muestre la topología en el controlador, la figura 20 indica la conexión exitosa entre los dispositivos:

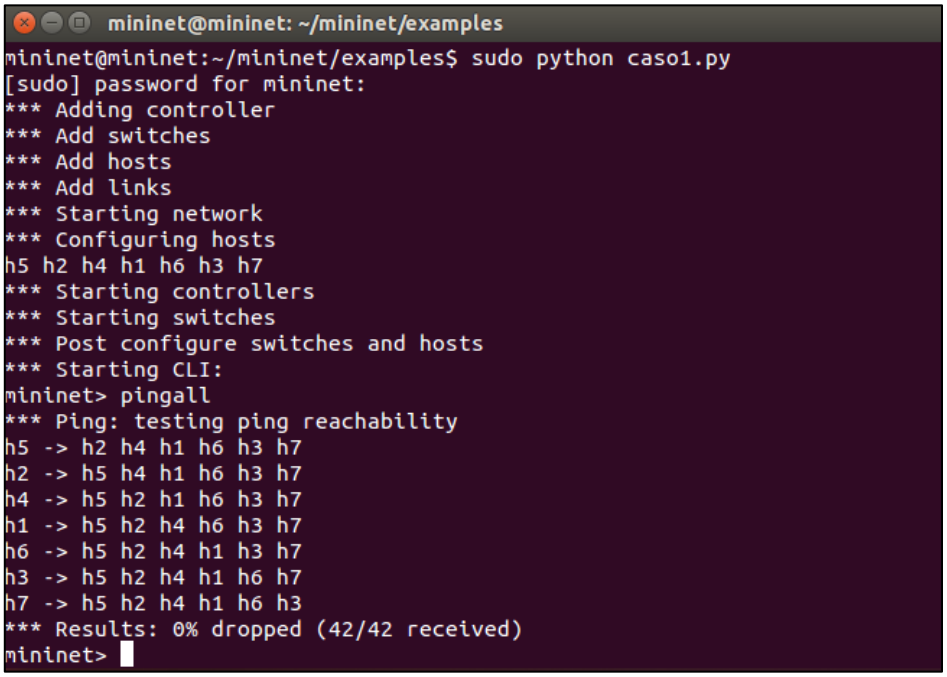


Figura 20. ***Conexión exitosa entre dispositivos***
Nota: Elaboración propia

Se procedió a acceder a la interfaz web de ODL a través del puerto 8181 con la dirección http://192.168.1.12:8181/index.html , para visualizar la topología de red en la interfaz DLUX de OpenDaylight se debe generar tráfico entre los hosts y luego recargar la GUI.

De esta manera, ya es posible visualizar la topología por completo, como se indica en la figura 21, resultando de gran ayuda para los administradores de red por cuanto permite descubrir automáticamente nodos y terminales de usuario.

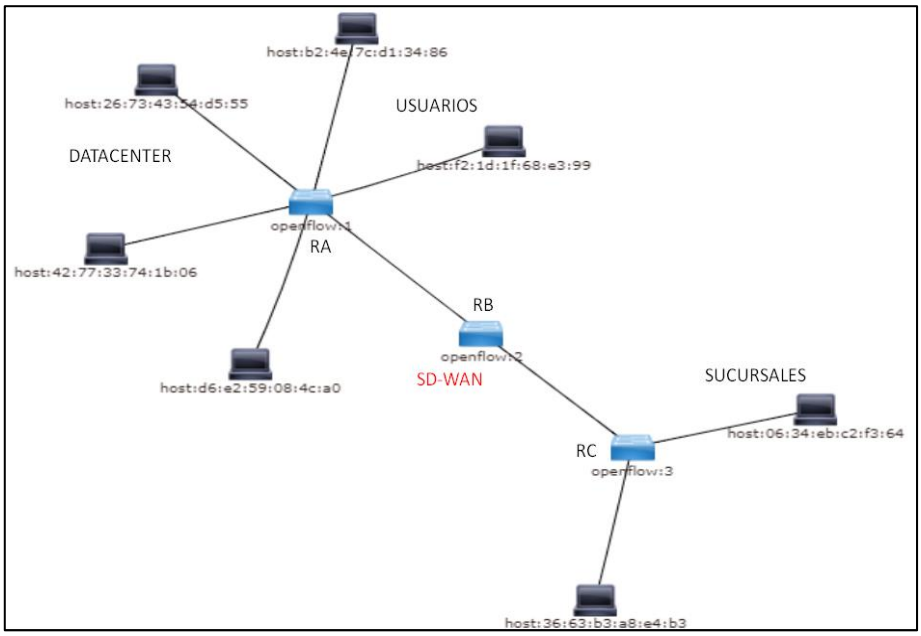


Figura 21. ***Topología de red vista desde ODL: esquema Greenfield***
Nota: Elaboración propia

Administración desde el Controlador OpenDaylight

En la figura 22 se muestra la GUI de ODL cuyos módulos aparecerán según las líneas de comando que se ejecutaron durante la instalación. Si no se han instalado los módulos, la interfaz aparecerá vacía.

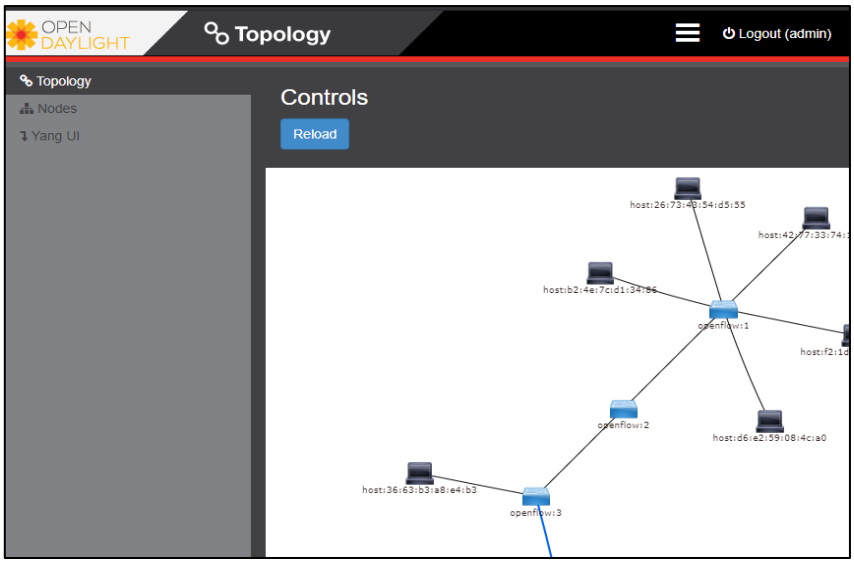


Figura 22. ***Panel de navegación de ODL.***
Nota: Elaboración propia

Es posible acceder a Yang Tools desde la GUI del controlador (véase figura 23), Yang es un lenguaje de modelado de datos que se utiliza para configurar datos manipulados por el protocolo de configuración de red NETCONF (Internet Engineering Task Force, 2010), aquí se tienen varios objetos y el primer paso fue obtener la topología de red operativa.

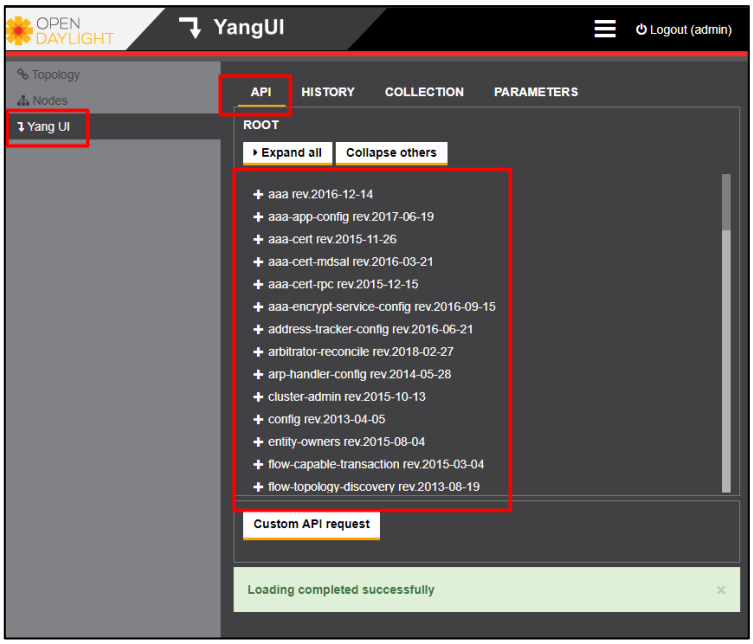


Figura 23. ***Yang UI de OpenDaylight***
Nota: Elaboración propia

Una vez que se haga click en la topología de red, Yang mostrará el link del API CONFREST que ocupa para salvaguardar esta información como se evidencia en la figura 24:

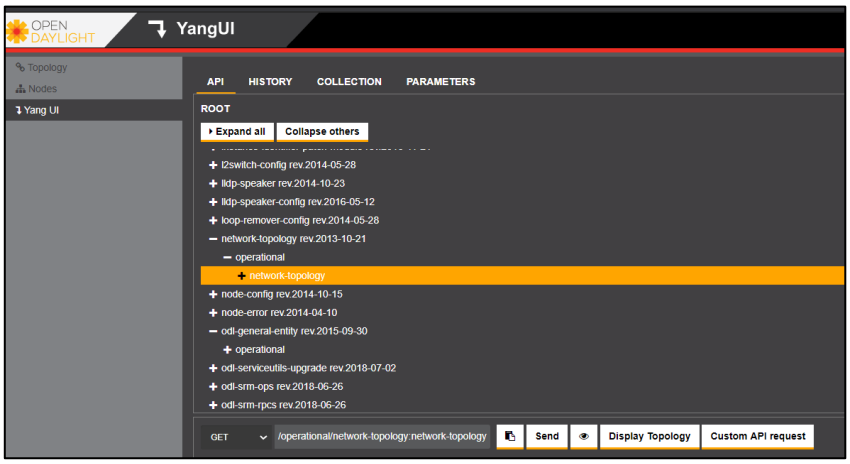


Figura 24. ***Generar topología operativa Yang UI.***
Nota: Elaboración propia

Se procede a copiar el link y se coloca en “send”, obteniendo lo que se visualiza en la figura 25, que corresponde a la topología de red operativa, donde se indican las direcciones ip de cada host, direcciones MAC.

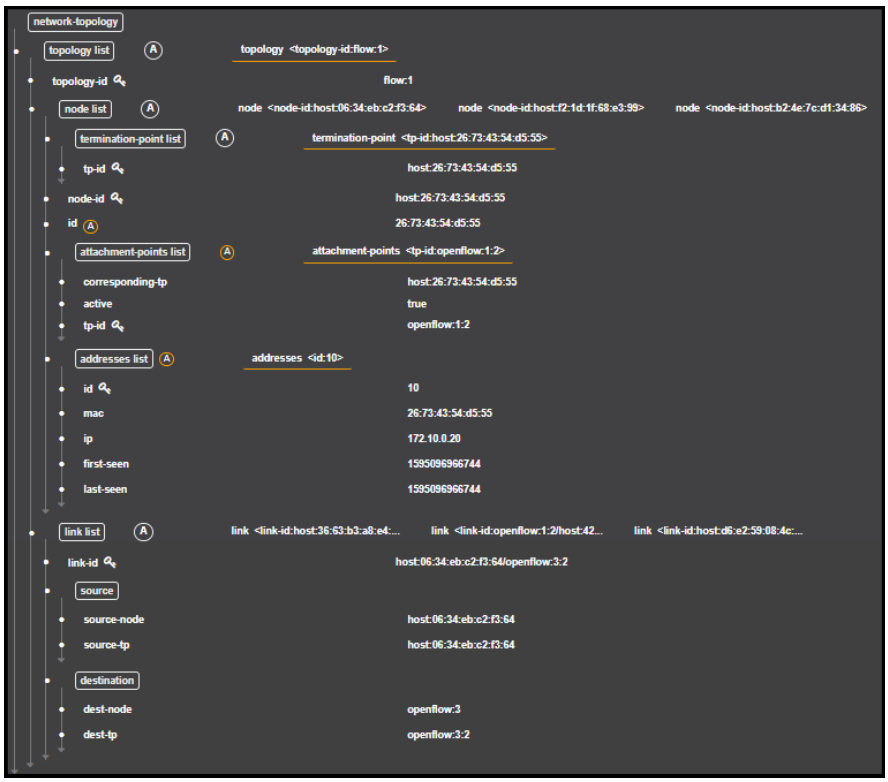


Figura 25. ***Topología operativa Yang UI.***
Nota: Elaboración propia

Seguridad en SDN/NFV

La metodología infiere en que es necesario plantear un escenario de red seguro, para ello se tiene la interfaz northbound del controlador OpenDaylight, la cual, mediante aplicaciones REST permite gestionar la red; en otras palabras, se puede crear un fichero.py en Python que realice diversas funciones como obtener información sobre los equipos, y permitir o denegar tráfico proveniente de ciertos hosts. ODL también permite la configuración de redes privadas virtuales haciendo uso del API REST para la creación de "puentes virtuales".

4.1.4. Aplicación Fase IV

Finalmente se tiene la **fase de validación** que consiste en monitorear y auditar permanentemente los cambios efectuados, a través de herramientas que prueben conectividad y permitan realizar troubleshooting.

Varias métricas definidas para las redes tradicionales son válidas también para las redes definidas por software. En esta investigación se definieron dos tipos de pruebas: pruebas de configuración o de conectividad y pruebas de desempeño.

Pruebas de Configuración

La finalidad de las pruebas de configuración es verificar el funcionamiento adecuado de la red definida por software, para ello se realizaron pruebas de conectividad con mensajes icmp entre las redes internas definidas por software, pruebas de interconexión entre la SD-WAN, y se probó la comunicación entre el controlador OpenFlow y los switches virtules. Se verificaron conexiones exitosas, las figuras 26, 27, y 28 indican los resultados. En el Anexo F parte F.1) se muestra con detalle todas las pruebas de configuración realizadas.

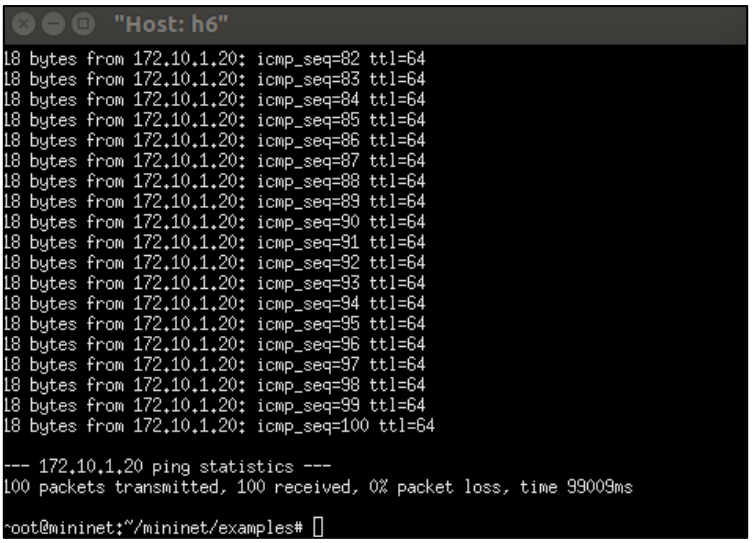


Figura 26. ***ICMP entre hosts de la SDN de sucursales.***
Nota: Elaboración propia

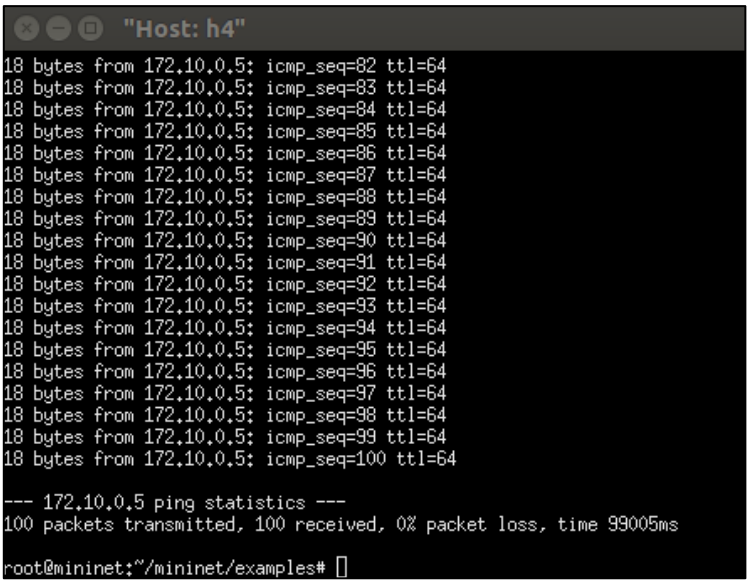


Figura 27. ***ICMP entre hosts de la SDN de matriz***
Nota: Elaboración propia

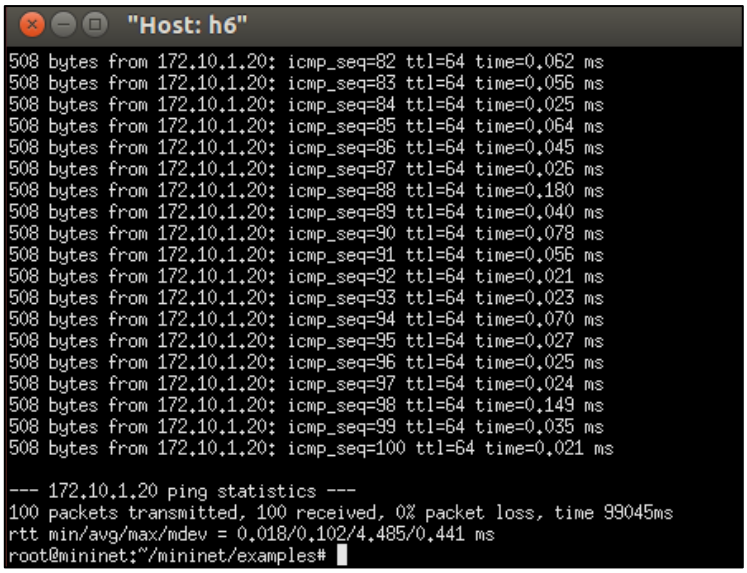


Figura 28. ***ICMP SD-WAN***
Nota: Elaboración propia

Pruebas de Desempeño

Las pruebas de desempeño tienen por objeto medir cuantitativamente una determinada métrica, en este caso se llevaron a cabo pruebas de latencia, tomando como referencia los tiempos de respuesta al enviar tramas de 64, 512, y 1518 bytes como lo indica el RFC 2544 de "Benchmarking Methodology".

Estos resultados deben validarse en contraste con las pruebas realizadas antes de ejecutar la transición, en el anexo F parte F.2) se indica con detalle los datos tomados antes de la transición, mientras que, en las figuras 29 y 30 se tiene la comparación de la red mpls/ip tradicional y la red definida por software:

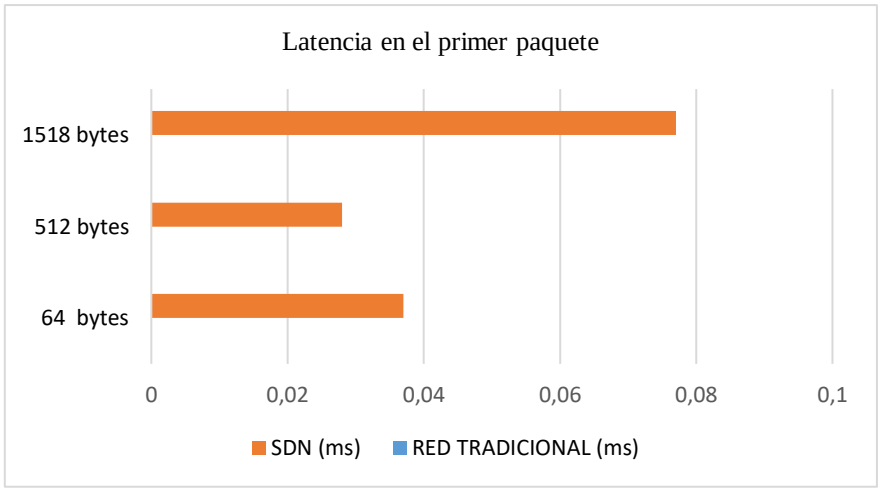


Figura 29. ***Latencia en el primer paquete: red tradicional vs red definida por software***
Nota: Elaboración propia

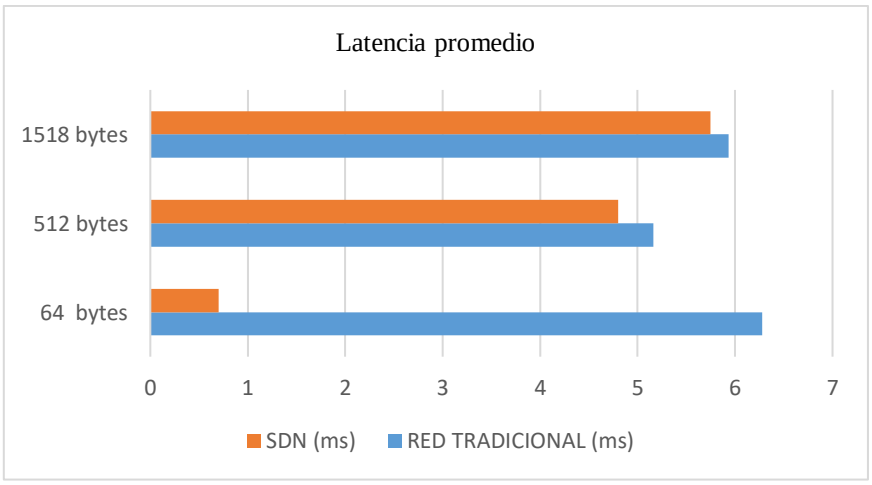


Figura 30. ***Latencia promedio: red tradicional vs red definida por software***
Nota: Elaboración propia

Nótese que la latencia del primer paquete icmp en la red tradicional, es inferior a la latencia introducida por la red SDN; esto se debe a que, en la red definida por software, el momento que llega el primer paquete, el Controlador OpenFlow aún no sabe cómo encaminarlo, entonces procede a encapsularlo dentro del protocolo OpenFlow y posteriormente lo reenvía; este procedimiento inicial utiliza un determinado tiempo que se

traduce como latencia en la red. Sin embargo, a medida que se continúan enviando los paquetes esta latencia va disminuyendo, hasta volverse mínima, mejorando el rendimiento de la red.

4.1.5. Resumen

Este escenario comprende una topología sencilla sin redundancia, por cuanto se trata de un esquema con equipos de capa tres que soportan OpenFlow a los cuales se les añade un dispositivo inteligente que centraliza la red (Opendaylight), como indica la figura 31:

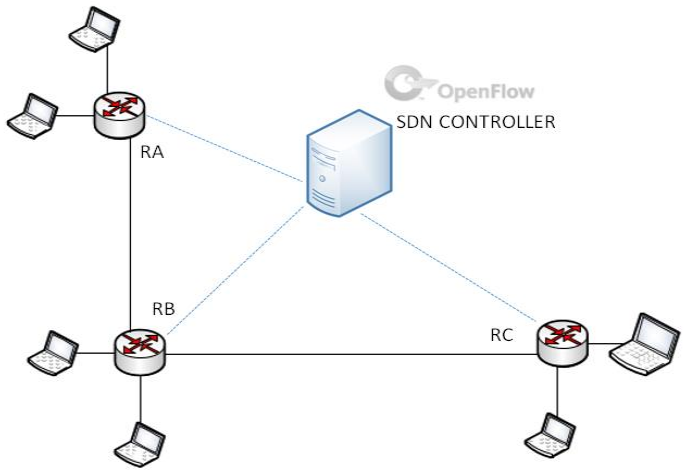


Figura 31. ***Transición Greenfield a SDN.***
Nota: Elaboración propia

Dado que se sigue el enfoque incremental de la metodología, y teniendo un resultado exitoso en la WAN, lo siguiente podría enfocarse en el aprovisionamiento de aplicaciones, y nuevamente se repetiría el ciclo.

4.2. Segundo Escenario: Esquema Mixto

Se expone un escenario de red corporativa con su oficina central que aloja servidores físicos y virtuales y sus sucursales que forman una topología redundante en su mayoría tipo árbol o anillo. En este tipo de entornos el concepto de redundancia se encuentra consolidado, así como también el tema de virtualización de servicios.

4.2.1. Aplicación Fase I

Se procede a aplicar la metodología, cuya **primera fase** corresponde a la recopilación de toda la información de red actual, servicios, protocolos, hardware, entre otros. No se debe olvidar clarificar el propósito de la migración.

La tabla 15 muestra la descripción de la Empresa Y, y la tabla 16 indica los componentes de red.

Tabla 15.
Descripción de la red inicial de la Empresa Y.

Item	Parámetro	Descripción
1	Target	Entidades financieras en desarrollo, internacionales de comercialización.
2	Modo operacional	Red MPLS/IP
3	Redundancia	HSRP y a nivel de capa física.
4	Herramientas de gestión	Monitoreo y administración con software propietario.
5	Disponibilidad	Alta disponibilidad por los servicios 24/7.

Nota: Elaboración propia.

Tabla 16.
Componentes de red de la Empresa Y

Item	Servicio / Componente	Descripción
1	Número de hosts de usuario	500
2	Topología de red	Tipo anillo
3	Ancho de banda	10Gps

4	Firewall	Equipos propietarios (ASA, Fortinet)
5	Servidor de correo	Zimbra / Centos 7, Hosting CPanel
6	Servidores web	Servicios virtualizados
7	Servidor de aplicaciones	Sistemas contables, de ventas y financieros.
8	Switches Layer 2	Capa 2 marca Cisco
9	Routers Layer 3	Capa 3 marca Cisco / Mikrotik
10	Protocolos de comunicación	BGP, OSPF, MPLS

Nota: Elaboración propia.

Se planteó el escenario de la figura 32 el cual se compone del típico escenario de la empresa en crecimiento, que consta de cuatro sitios o subredes: datacenter y varias sucursales. El core de red está diseñado por cuatro switches de capa tres que forman una topología redundante tipo anillo. Cada enlace WAN es de 10Gbps, siendo que, cada uplink de éstos, se conecta a switches capa 2 que convergen la red según cierta segmentación.

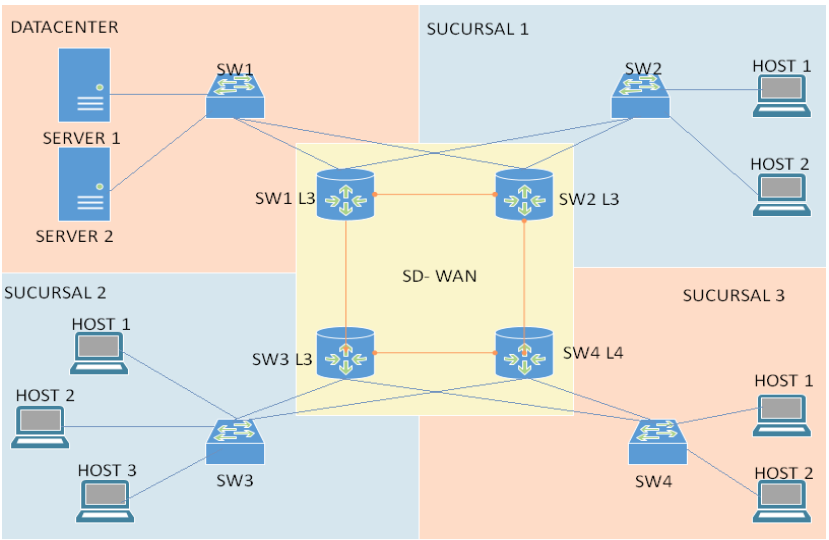


Figura 32. ***Topología segundo caso: esquema de red mixto.***
Nota: Elaboración propia

Siguiendo el proceso metodológico se procedió a simular en la herramienta GNS3, la topología propuesta en la figura 33, para establecer las métricas de comparación. En el Anexo G, parte G.1) se evidencian todas las pruebas de conectividad realizadas. La topología simulada en GNS3 se indica en la figura 31, para la cual se utilizaron routers marca cisco serie 7200, que manejan MPLS/IP, cuyo direccionamiento ip se indica en la tabla 17.

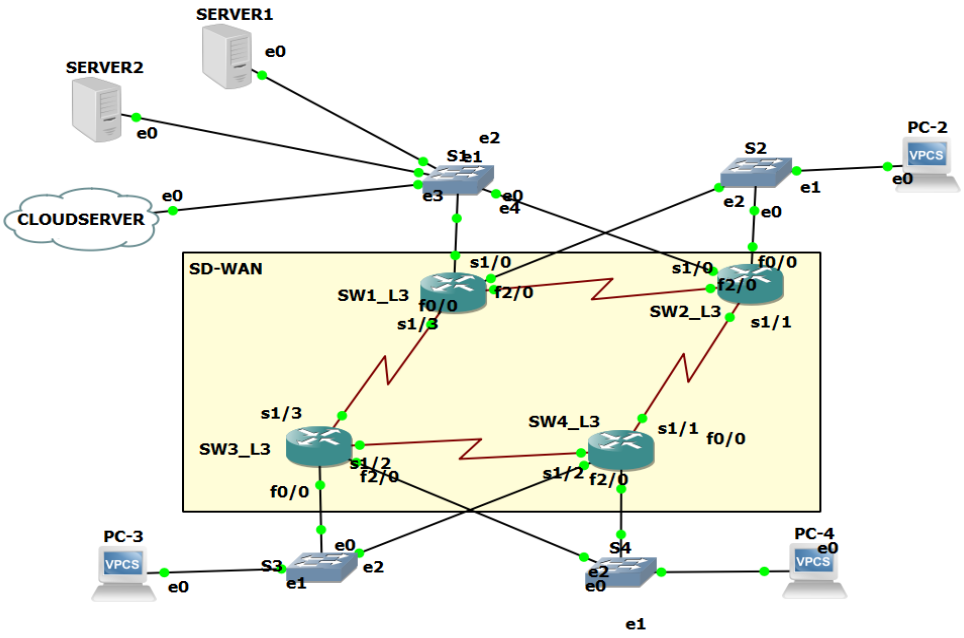


Figura 33. ***Topología red tradicional: escenario mixto, segundo caso en GNS3***
Nota: Elaboración propia.

Tabla 17.
Direccionamiento red tradicional esquema mixto en GNS3

Dispositivo	Interfaz	Dirección IP	Máscara de subred	Gateway predeterminado
SW1_L3	S1/0	10.10.10.1	255.255.255.252	N/A
	S1/3	10.10.40.2	255.255.255.252	N/A
	F0/0	192.168.10.1	255.255.255.0	N/A
	F2/0	192.168.20.2	255.255.255.0	N/A
SW2_L3	S1/0	10.10.10.2	255.255.255.252	N/A
	S1/1	10.10.20.1	255.255.255.252	N/A
	F0/0	192.168.20.1	255.255.255.0	N/A
	F2/0	192.168.10.2	255.255.255.0	N/A
SW3_L3	S1/2	10.10.30.2	255.255.255.252	N/A
	S1/3	10.10.40.1	255.255.255.252	N/A
	F0/0	192.168.40.1	255.255.255.0	N/A
	F2/0	192.168.30.2	255.255.255.0	

	S1/1	10.10.20.2	255.255.255.252	N/A
SW4_L3	S1/2	10.10.30.1	255.255.255.252	N/A
	F0/0	192.168.30.1	255.255.255.0	N/A
	F2/0	192.168.40.2	255.255.255.0	N/A
SERVER1	Eth0	192.168.10.200	255.255.255.0	192.168.10.254
SERVER2	Eth0	192.168.10.201	255.255.255.0	192.168.10.254
SERVER 3	Eth0	192.168.10.202	255.255.255.0	192.168.10.254
CLOUD SERVER	N/A	192.168.10.203	255.255.255.0	192.168.10.254
PC2	Eth0	192.168.20.21	255.255.255.0	192.168.20.254
PC3	Eth0	192.168.30.31	255.255.255.0	192.168.20.254
PC4	Eth0	192.168.40.41	255.255.255.0	192.168.20.254

Nota: Elaboración propia.

Como indica la metodología, en esta fase se tiene toda la recopilación de información acerca de la estructura de red actual, a continuación, en la tabla 18, se muestran los puntos mínimos que deberá cumplir la red NFV/SDN.

Tabla 18.
Checklist de red escenario mixto

Item	Parámetro		Observaciones
1	SEGURIDAD		
	Tiene un firewall para proteger su red interna del acceso no autorizado.	☑	
	Las contraseñas de los dispositivos de red se manejan por roles.	☑	Solamente se tiene el rol de administrador.
	Cada cambio de configuración es documentado y almacenado en una bitácora.	☑	
	Se utilizan solamente canales seguros para acceder a la red.	☑	Direcciones ip públicas asignadas a un servidor de resolución de dominios que deben estar disponibles 24/7.
	Se deshabilitan reglas de firewall innecesarias.	☑	Equipo de administración de red consolidado, que realiza diferentes funciones como: Tecnologías de la Información, Soporte Técnico, Networking, Sistemas y Control de calidad de procesos tecnológicos.

2	DISPOSITIVOS DE RED	
	Descarga de firmware, actualizaciones y parches de fuentes validadas.	☑
	Se tiene una lista de todo el hardware de red, dispositivo, ubicación, número de serie.	☑
	Se utiliza protocolos de administración como SNMP.	☑
3	GESTIÓN DE SOFTWARE	
	Uso exclusivo de software compatible, libre o con licencia.	☑
	Software pirata se elimina de los equipos	☑
	Uso de redes privadas virtuales para acceso remoto.	☑

Nota: Elaboración propia.

El siguiente paso consiste en determinar los dispositivos que serán parte de la capa de infraestructura, analizar las brechas actuales si existieren, para esto se debe seguir en enfoque incremental; aplicando el concepto de migración por porciones de red.

Se sugiere empezar por la parte más aislada de red, entendiéndose por aislada, aquella en donde los servicios siguen siendo de propósito de particular, pero ya combinados con hosting u otro tipo de arrendamientos.

Siguiendo el ejemplo de la Empresa Y, se tienen servidores http, resolución de dominios, ftp, correo y aplicaciones. Estos se encuentran en hardware específico y solamente el servidor de correo se aloja en hosting. Así se sugiere, aprovisionar el hardware disponible con todos los recursos físicos y sobre esto levantar las aplicaciones necesarias; en un proceso totalmente transparente para el usuario.

Aplicando el concepto de virtualización de funciones de red se hace posible la consolidación de diversas funciones de red, facilitando su administración. Para esto ya se

requiere la existencia de un VMM (Virtual Machine Monitor) o un hipervisor, que vendría a ser la capa intermedia entre el hardware subyacente y cada máquina virtual.

En conclusión, para el ejemplo planteado es posible consolidar los servicios nombrados anteriormente a través de un solo hipervisor, que acorde, a las necesidades podrá ser alojado o no.

4.2.2. Aplicación Fase II

El objetivo de esta fase es dejar la red en un estado "bare", es decir, lista para la transición, por lo que, se deben homologar la versión OpenFlow a utilizar y planear las estrategias de seguridad correspondientes, las cuales se obtienen a partir de la documentación recabada en la fase 1.

Capa de virtualización: como se mencionó anteriormente, la capa de virtualización estará conformada por VirtualBox 6.0.22, que corresponde a un hipervisor alojado sobre una máquina con Windows 10 Pro, de 64bits, procesador Intel (R) Core (TM) i7 con 16.0 GB de memoria RAM; y el controlador de código abierto OpenDaylight.

Elección del controlador SDN: en la tabla 6, se tiene la comparación entre controladores de código abierto con soporte OpenFlow, para los ejemplos presentados en este trabajo se utilizó OpenDaylight.

Dispositivos a migrar: se trata de aprovisionar recursos físicos que podrían estar subutilizados, se empezará con la migración de:

- Servidor http
- Servidor ftp
- SD-WAN

La tabla 19 es un ejemplo de las características mínimas que se deben exponer por cada servidor para determinar cuál resulta más robusto para el aprovisionamiento de recursos. En el ejemplo se muestran las características de un servidor Hp Proliant DL380 Gen 9.

Tabla 19.
Características de un servidor Hp ProLiant DL380 Gen9

Item	Característica	Descripción
1	Fabricante	HP
2	Modelo	HPE ProLiant DL380 Gen9
3	Número de procesadores	2
4	Velocidad de procesador	3,5 GHz
5	Factor de forma del chasis	Bastidor
6	Tipo de fuente de alimentación	Ranura flexible
7	Memoria	3 TB
8	Ranuras de memoria	24 DIMM
9	Tipo de memoria	DDR4
10	Controlador de almacenamiento	Dynamic Smart Array P840

Nota: Tomado de Digital Data Sheel Hewlett Packard Enterprise (HP, 2018)

Elección de versión: se trata de revisar que los dispositivos que forman parte de la SD-WAN tengan soporte OpenFlow para la misma versión, por lo que podría ser necesario realizar actualizaciones de firmware.

Por ejemplo, Cisco ofrece un Plug-in para sus dispositivos compatible; la matriz de compatibilidad se encuentra en el anexo I, y en la tabla 20 se observa el soporte del controlador, que indica que a partir de la serie 9300 los equipos ya permiten configurar el plugin para redes definidas por software.

Tabla 20.
Soporte para el complemento OpenFlow en equipos Cisco

Item	Versión OpenFlow	Dispositivos compatibles
1	OpenFlow 1.0	Cisco Nexus serie 9300, 3000, 31128PQ1 3232C, 9500, 9200.
2	OpenFlow 1.3	Ixia, 3264Q

Nota: Tomado de Guía de configuración 1.3 de OpenFlow (CISCO, 2017)

El plugin consiste en una aplicación que se levanta sobre el contenedor de servicios virtuales a nivel del sistema operativo en un dispositivo. El complemento de Cisco para OpenFlow se entrega en un paquete de aplicación virtual abierta (OVA). El paquete OVA se instala y activa en el dispositivo a través de la CLI (CISCO, 2017).

Antes de instalar y activar Cisco Plug-in para OpenFlow , se debe asegurar de que exista un paquete OVA compatible con el dispositivo en un servidor FTP conectado.A través de los comandos que se muestran en la tabla 21, es posible configurar las entradas de flujo en dispositivos cisco:

Tabla 21.
Configuración de entradas de flujo OpenFlow en equipos Cisco.

Paso	Comando	Descripción
1	enable	Habilita el modo EXEC privilegiado.
2	configure terminal	Ingresa al modo de configuración global.
3	hardware profile tcam region vacl 0	Configura el tamaño de la región TCAM para las listas de control de acceso (ACL) de VLAN.
4	hardware profile tcam region e-vacl 0	Configura el tamaño de la región TCAM para las ACL de VLAN de salida.
5	hardware profile tcam region racl 0	Configura el tamaño de la región TCAM para las ACL del enrutador.
6	hardware profile tcam region e-racl 0	Configura el tamaño de la región TCAM para las ACL del enrutador de salida.
7	hardware profile tcam region qos 256	Configura el tamaño de la región TCAM para QoS
8	hardware access-list tcam region openflow 1408 hardware access-list tcam region openflow 1408 double-wide region openflow 1408 double-wide	Configura el tamaño de la región TCAM para las ACL de interfaz. Para adaptarse a los criterios de coincidencia adicionales de las direcciones MAC de origen y destino, el conmutador Cisco Nexus 3000 admite una nueva región TCAM, ifacl de doble ancho , que es una ACL de interfaz de doble ancho. Los tamaños de doble ancho ifacl e ifacl para Cisco Nexus 3172 son 3072 y 1536, respectivamente

9	exit	Sale del modo de configuración global y entra en el modo EXEC privilegiado.
10	copy running-config startup-config	Guarda el cambio de forma persistente a través de reinicios y reinicia copiando la configuración en ejecución a la configuración de inicio.
11	reload	Vuelve a cargar el sistema operativo de un dispositivo para que pueda iniciarse el soporte del contenedor de servicios virtuales para el hardware del dispositivo.

Nota: Tomado de Guía de configuración 1.3 de OpenFlow (CISCO, 2017)

Para el ejemplo, se asumió que los dispositivos cisco manejan openFlow en diferentes versiones, por lo que habría que homologarlas mediante la actualización del contenedor de servicios virtuales cisco.

En equipos Mikrotik, en cambio, se tiene compatibilidad a partir de la serie Mikrotik Router Board 750GL en su sistema operativo RouterOS 6.42.1, por lo que el primer paso sería actualizar el firmware de los equipos a ésta versión o superior. Paso siguiente, acceder al equipo mediante winbox y arrastrar los paquetes para su instalación, por último, se deberá reiniciar el dispositivo. Para muestra de esto, (véase figura 34), se tuvo acceso a un switch mikrotik 750GL, que a través de la línea de comandos se añadió:

[admin@MikroTik]>/openflow add name=ofswitch1 controllers=192.168.2.254

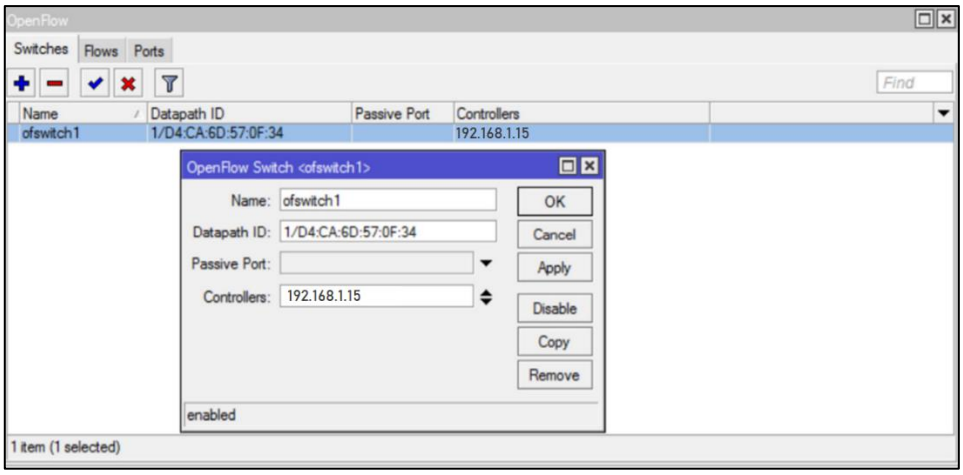


Figura 34. ***Ejemplo de configuración OpenFlow en Mikrotik.***
Nota: Elaboración propia.

Nótese en la figura 35, que ahora en las interfaces del switch, ya reconoce la interfaz ethernet como puerto OpenFlow:

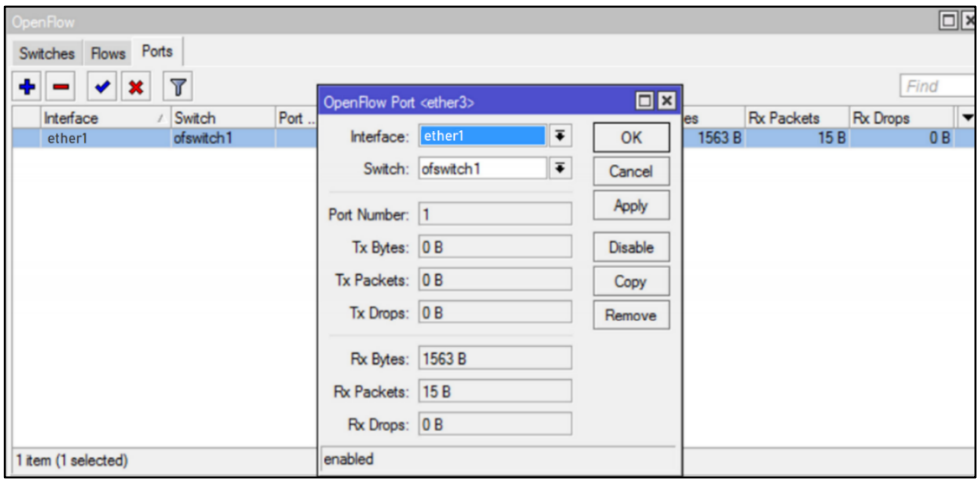


Figura 35. ***Ejemplo de configuración Interfaz OpenFlow en Mikrotik.***
Nota: Elaboración propia.

Estrategias de Seguridad: se propone un nivel de seguridad elevado basado en roles de usuario, tanto a nivel físico como lógico.

En consecuencia, y siguiendo el estándar ETSI NFV 001 ahora se tiene una red en estado “bare”, lista para una transición.

4.2.3. Aplicación Fase III

La etapa siguiente es la transición, en la cual se empieza con el proceso de cambios, para ello se procedió a emular la topología de red de la figura 30 en la herramienta Mininet.

Diseño de la Red Definida por Software con Mininet

Se realizó el levantamiento de la topología en Mininet, siguiendo el mismo esquema configurado en el simulador de redes ip GNS3, a través de miniedit. La figura 36 muestra el diseño planteado.

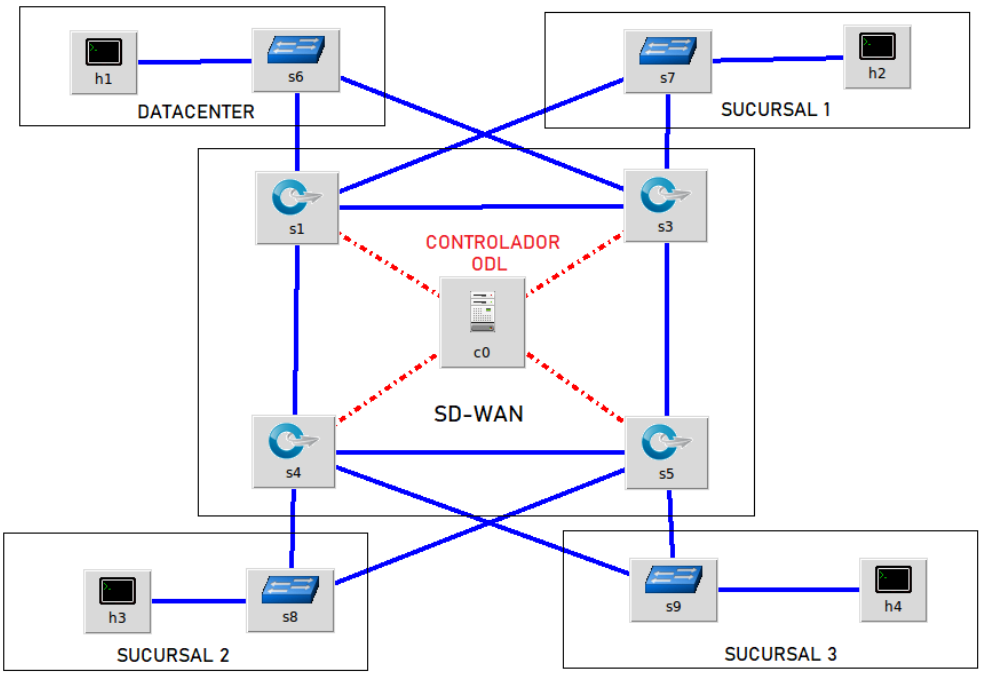


Figura 36. ***Simulación de red SDN en mininet: esquema mixto.***
Nota: Elaboración propia.

Se utilizó el mismo direccionamiento de la topología realizada en GNS3, y siguiendo el procedimiento explicado en el apartado 5.1.1. Finalmente se accede a la web de OpenDaylight para lo cual primero se debe generar tráfico mediante el comando *pingall* y luego, se podrá observar la topología de la figura 37.

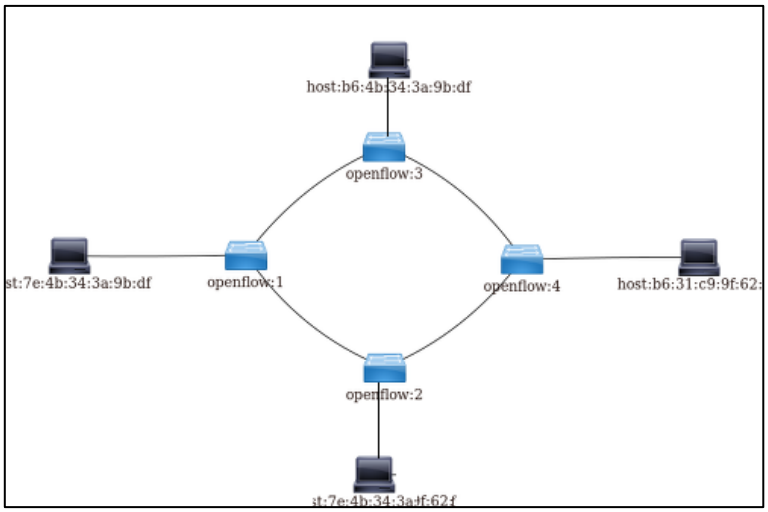


Figura 37. ***Topología de red vista desde ODL: esquema mixto.***
Nota: Elaboración propia.

Como se puede observar en la figura 38, se tiene migrada la parte WAN de la red, por lo que los dispositivos ya son visibles a través de ODL. Y dependiendo de los módulos instalados se puede acceder a Yang como se indica en la figura 36.

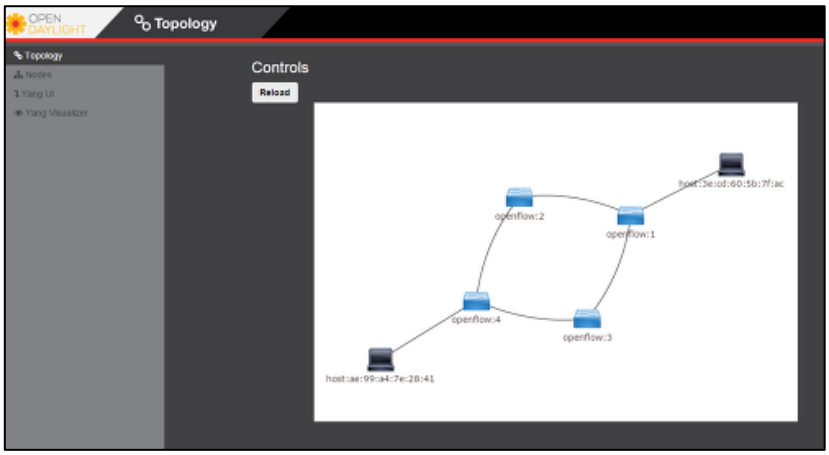


Figura 38. ***Panel de navegación ODL: esquema mixto.***
Nota: Elaboración propia.

4.2.4. Aplicación Fase IV

La etapa de validación permite monitorear permanentemente los cambios realizados; a continuación, las pruebas de conectividad y desempeño.

Pruebas de Configuración

Mediante el envío de solicitudes ICMP se verificó la conectividad entre dispositivos: switches virtuales, controlador OpenFlow y hosts. Las figuras 39, 40 y 41 muestran los resultados mientras que en el Anexo G parte G.1) se evidencia el detalle de todas las pruebas realizadas.

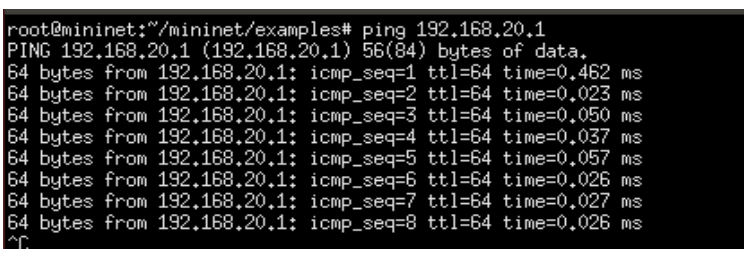


Figura 39. ***ICMP entre SW_L3 de la SDN***
Nota: Elaboración propia

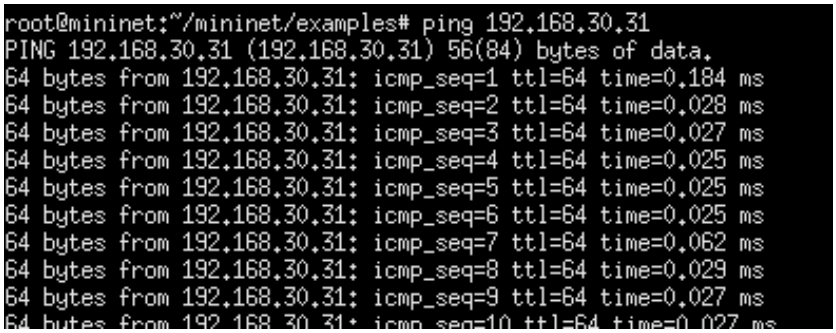


Figura 40. ***ICMP entre hosts de sucursales***
Nota: Elaboración propia

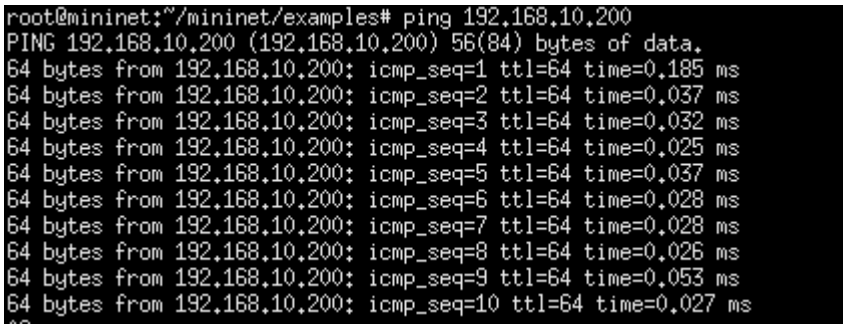


Figura 41. ***ICMP entre datacenters y sucursales***
Nota: Elaboración propia

Pruebas de Desempeño

Mediante el envío de tramas de 64, 512 y 1518 bytes como indica el RFC 2544 se puede medir cuantitativamente la latencia en la presente red. Los resultados obtenidos después de la transición se deben validar respecto a los datos recopilados antes de efectuar cambios; el anexo G parte G.2) muestra la data recopilada antes del procedimiento. Las figuras 42 y 43 muestran la comparación entre la red ip tradicional y la red definida por software.

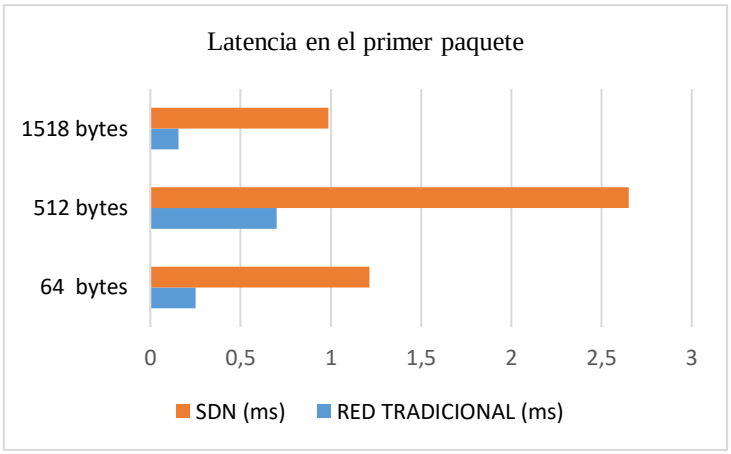


Figura 42. ***Latencia en el primer paquete: red tradicional vs red definida por software: esquema mixto.***
Nota: Elaboración propia

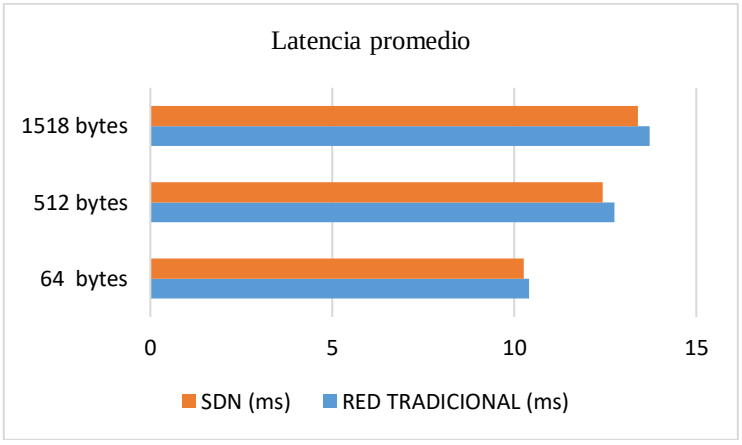


Figura 43. ***Latencia promedio: red tradicional vs red definida por software: esquema mixto.***
Nota: Elaboración propia

4.2.5. Resumen

Este escenario corresponde a una topología tipo anillo que añade redundancia por cuanto se orienta a organizaciones que deben mantener una alta disponibilidad de sus servicios. El primer paso fue homologar la versión y soporte del protocolo OpenFlow, para, posteriormente realizar la migración. La figura 44 muestra el resultado.

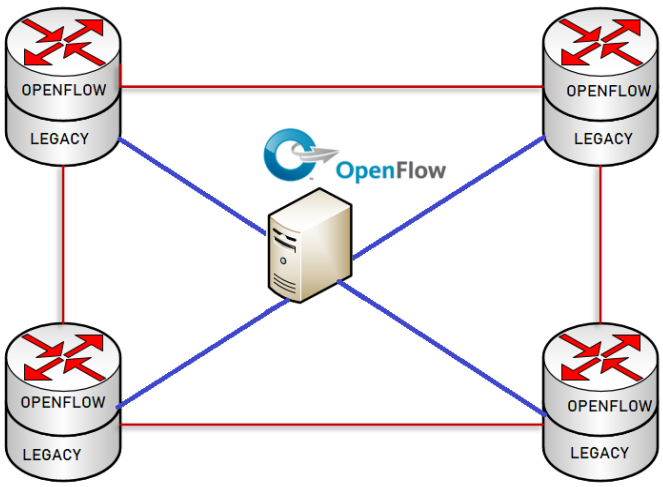


Figura 44. ***Transición esquema mixto a SDN.***
Nota: Elaboración propia

4.3. Tercer Escenario: Esquema Híbrido

Probablemente es el entorno más utilizado en redes empresariales puesto que combina una red legacy o tradicional con uso de infraestructura en la nube, creando un escenario híbrido además de incorporar dispositivos multivendor. La inclusión de las redes definidas por software en las empresas requiere la actualización de los equipos de red heredados, lo que supone altos costos para la organización. La solución a ello, es el despliegue gradual de SDN en forma de redes híbridas.

Este escenario se basa en los esquemas de red mucho más extensos, el entorno de simulación se muestra en la figura 45, comprende las máquinas virtuales de OpenDaylight y Mininet, las cuales se ejecutan en instancias separadas. Se hizo uso de la herramienta GNS3 a fin de proporcionar conectividad entre ODL y Mininet que pertenecen a la misma área de red local. A continuación, se describen los procesos que se llevarán a cabo en cuanto al plano de datos y plano de control:

- Plano de datos: se ejecutará a través de los switches virtuales que operan con Openvswitch, sistema operativo que permite levantar los procedimientos adecuados para la conmutación de paquetes además de ofrecer interoperabilidad con protocolos como 802.1q, LACP u OpenFlow.
- Plano de control: ODL se encargará de la administración de la red definida por software a través de OpenFlow y su interfaz DLUX.

4.3.1. Aplicación Fase I

Al igual que en los casos anteriores, se realiza el mapeo de red, en la figura 43 se propone el esquema cuyo hipervisor sigue siendo la máquina virtual que aloja mininet en una instancia, el controlador OpenDayligh en otra instancia y emula la red física tradicional en GNS3.

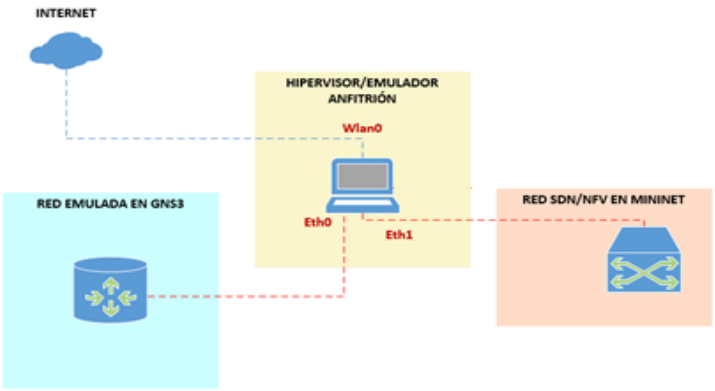


Figura 45. ***Simulación de esquema de red híbrido.***
Nota: Elaboración propia

Al realizar la descripción del escenario propuesto se tiene que es una red extensa que ya opera con dispositivos particulares y funciones de red en la nube; por lo que se debe aplicar el enfoque gradual que sugiere la metodología propuesta; por ejemplo, lograr la migración de cierta parte de la pila TCP/IP en el punto de convergencia entre los servicios virtualizados y el hardware heredado. Esto implica que, la utilización de la SDN se va a limitar a cierta capa de la pila TCP/IP correspondiente al nivel de acceso.

4.3.2. Aplicación Fase II

En este apartado es importante delimitar la porción de red que estará implicada en el proceso de transición, es decir los switches SDN deben coexistir con el hardware heredado; por lo que identificar los equipos de enrutamiento constituye la tarea principal de este apartado.

Para el escenario propuesto se requiere lograr la coexistencia entre el segmento de red física y el segmento de red virtualizado, por tanto, los equipos que se van configurar son el router de borde y el controlador ODL.

Doble Pila entre la Red Definida por Software y la Red Tradicional

La red tradicional utiliza protocolos de enrutamiento estáticos o dinámicos para su convergencia, sin embargo, al implementar dispositivos openflow como se verá posteriormente, estos no se comunicarían a través de los protocolos tradicionales como IGP, RIP, OSPF, lo harán a través de los flujos openflow, por lo tanto es necesario un mecanismo de transición que permita la convergencia "doble pila."

El controlador SDN seleccionado OpenDaylight permite la integración del protocolo de puerta de enlace de frontera BGP (o Border Gateway Protocol) definido por (Rekhter, 2006) en el RFC4271 que lo describe como un mecanismo para enrutar dominios sin clase entre sistemas autónomos (AS), en este caso permitirá el intercambio de paquetes entre el dominio de la red definida por software y la red tradicional.

Una vez implementado el controlador SDN con los módulos que permitirán la convergencia BGP – red tradicional – red sdn, se puede decir que la arquitectura se encuentra en un estado "bare" por lo que se puede continuar con la etapa siguiente (Véase anexo C para instalación de ODL)

4.3.3. Aplicación Fase III

El primer paso de la transición será configurar el controlador que se encargará de la gestión centralizada, en este caso OpenDaylight (Véase anexo H.1). Lo siguiente será emular la red virtualizada a través de Mininet, para ello es posible ejecutar el comando *sudo mn –controller=remote,ip=192.168.1.6,port=6633 –topo=tree,depth=3* o realizarlo a través de la herramienta miniedit .

Se hizo uso de la herramienta GNS3 para representar la arquitectura de red tradicional y brindar conectividad hacia el controlador openflow, la máquina virtual Mininet con la red que se implementaría después utilizando el direccionamiento de la tabla 22.

Tabla 22.
Direccionamiento ip del escenario híbrido.

Dispositivo	Dirección Ip	Máscara	Gateway
Controlador SDN	192.168.1.11	255.255.255.0	
Mininet	192.168.1.20	255.255.255.0	
Router de borde	192.168.1.250	255.255.255.0	
Host red tradicional	10.10.10.2	255.255.255.0	10.10.10.1

Nota: Elaboración propia

El controlador SDN se comunicará con la topología realizada en mininet a través de openflow mientras que, con el router tradicional lo hará a mediante BGP.

Topología del Escenario Híbrido

La topología propuesta (véase figura 46) hace referencia a las infraestructuras de red estudiadas anteriormente, la cual se compone de las siguientes partes:

- Switches Core: dispositivos con redundancia y de convergencia rápida.
- Switches de distribución: convergencia con los switches de acceso, es posible implementar QoS.
- Switches de acceso: conectividad hacia el usuario final.

- Servicios (Datacenter): alta disponibilidad a servidores web, aplicaciones, correo electrónico, entre otros.

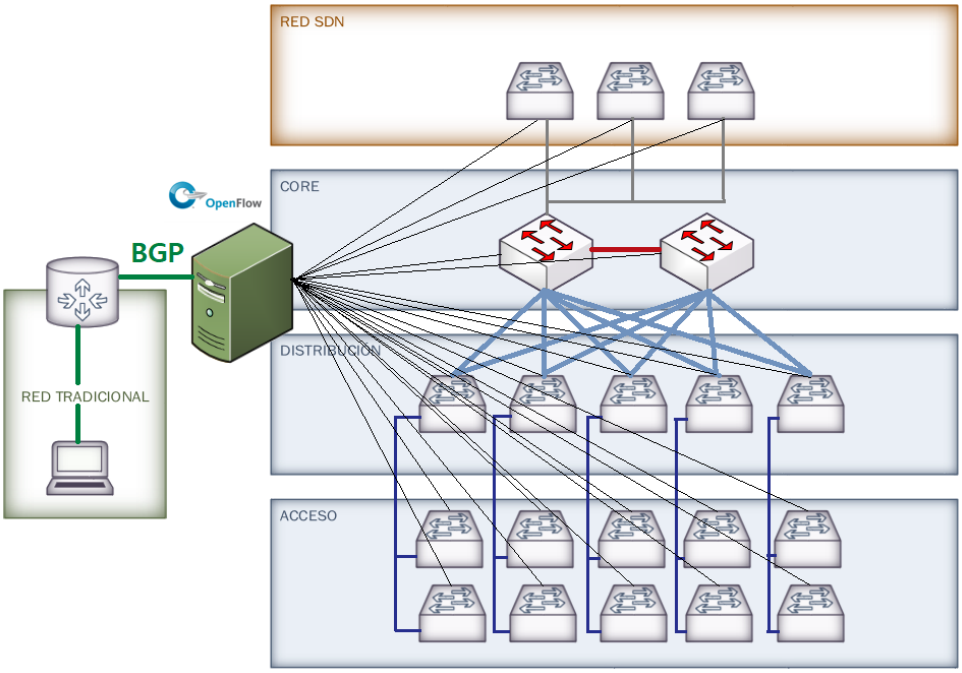


Figura 46. ***Topología de esquema de red híbrido.***
Nota: Elaboración propia

La figura 47 muestra el esquema de red simulado en la herramienta GNS3, la red tradicional se implementa desde el router R1 que es un cisco 7200 con hosts finales tales como PC1 . VirtualBox actúa como container de Mininet, el Switch ethernet a diferencia del router R1 no utiliza una imagen de IOS ya que solamente hace bridge hacia los hosts, se utilizó el controlador OpenDaylight dentro de la red local.

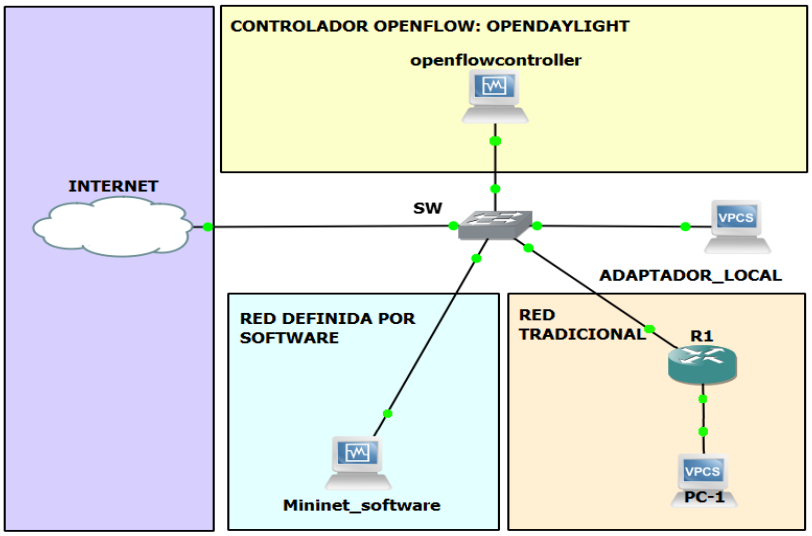


Figura 47. ***Diagrama del esquema de simulación en GNS3***
Nota: Elaboración propia

Redes Virtuales en OpenDaylight (VLAN sobre ODL)

Dentro de la red SDN, ODL brinda la posibilidad de implementar redes virtuales, por ejemplo, si se tiene una topología con tres switches y dos hosts por cada uno, es posible crear dos vlan (vlan 100 y vlan 200), asignando tres hosts a cada vlan. La figura 48 muestra la topología:

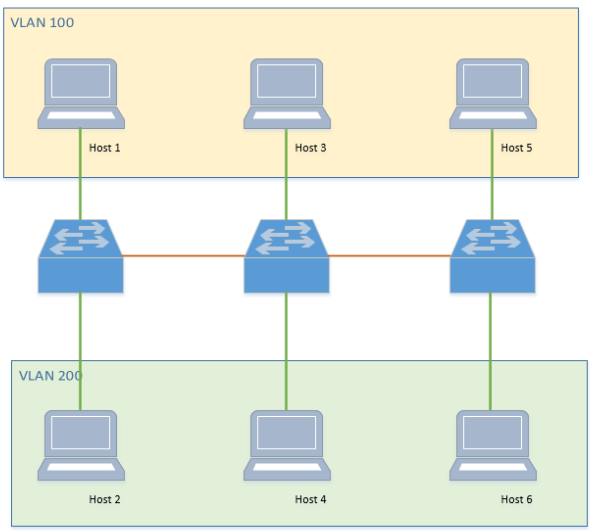


Figura 48. ***Redes virtuales sobre ODL***
Nota: Elaboración propia

Se utilizó OpenDaylight y VTN para la implementación de VLANS o aislamiento a nivel de capa 2, los switches expuestos son los OVS de mininet.

El primer paso será configurar el entorno mininet con la topología de la figura 45, dentro de las líneas de código se puede evidenciar que se están creando las interfaces virtuales en los hosts y a su vez indicando que se utilice el protocolo de enrutamiento intervlan 802.1Q, esto quiere decir que cada host tiene una etiqueta 802.1Q que indica el vlan id al que pertenece. Es posible que el paquete vconfig no esté instalado, por lo que será necesario ejecutar el comando de instalación de la figura 49.

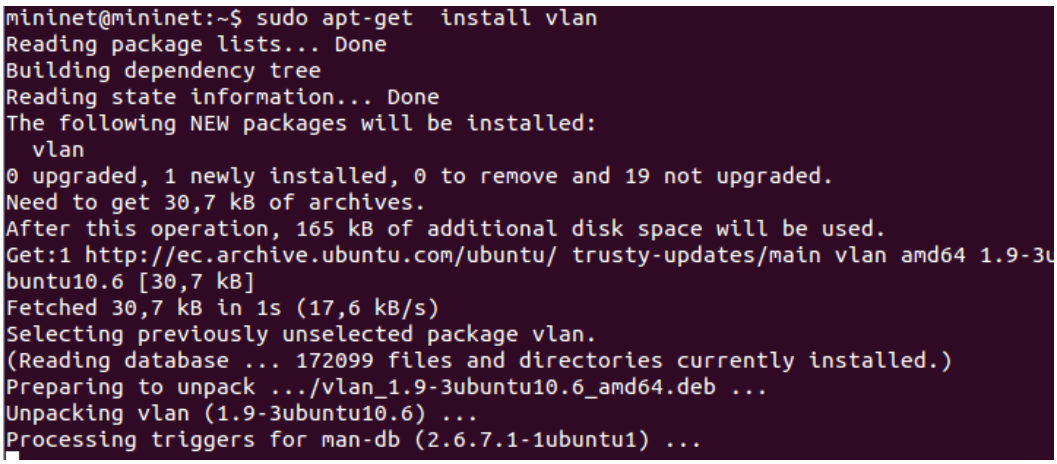


Figura 49. ***Comando de instalación de vlans en mininet***
Nota: Elaboración propia

Lo siguiente será ejecutar la topología mediante el comando de la figura 50:

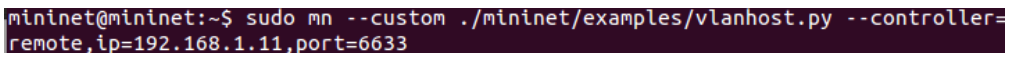


Figura 50. ***Topología de redes privadas virtuales***
Nota: Elaboración propia

Ahora ya será posible visualizar la topología desde ODL como en la figura 51. El script en Python se encuentra en el anexo H.2.

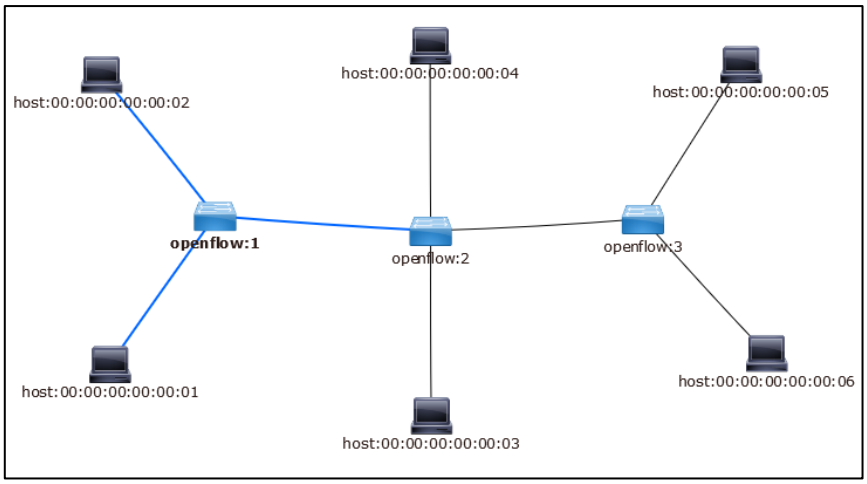


Figura 51. ***Redes virtuales sobre ODL***
Nota: Elaboración propia

4.3.4. Aplicación Fase IV

La fase de validación es la prueba final que permite corroborar la operatividad de la red mediante el uso normal de sus funciones. El monitoreo del rendimiento es el que permitirá recabar información para una posible fase de optimización.

Topología vista desde el controlador OpenDaylight

El escenario simulado se llevó a cabo conectando los openvswitch al controlador opendaylight dando como resultado el esquema de la figura 52. ODL muestra la topología de red a través del módulo DLUX.

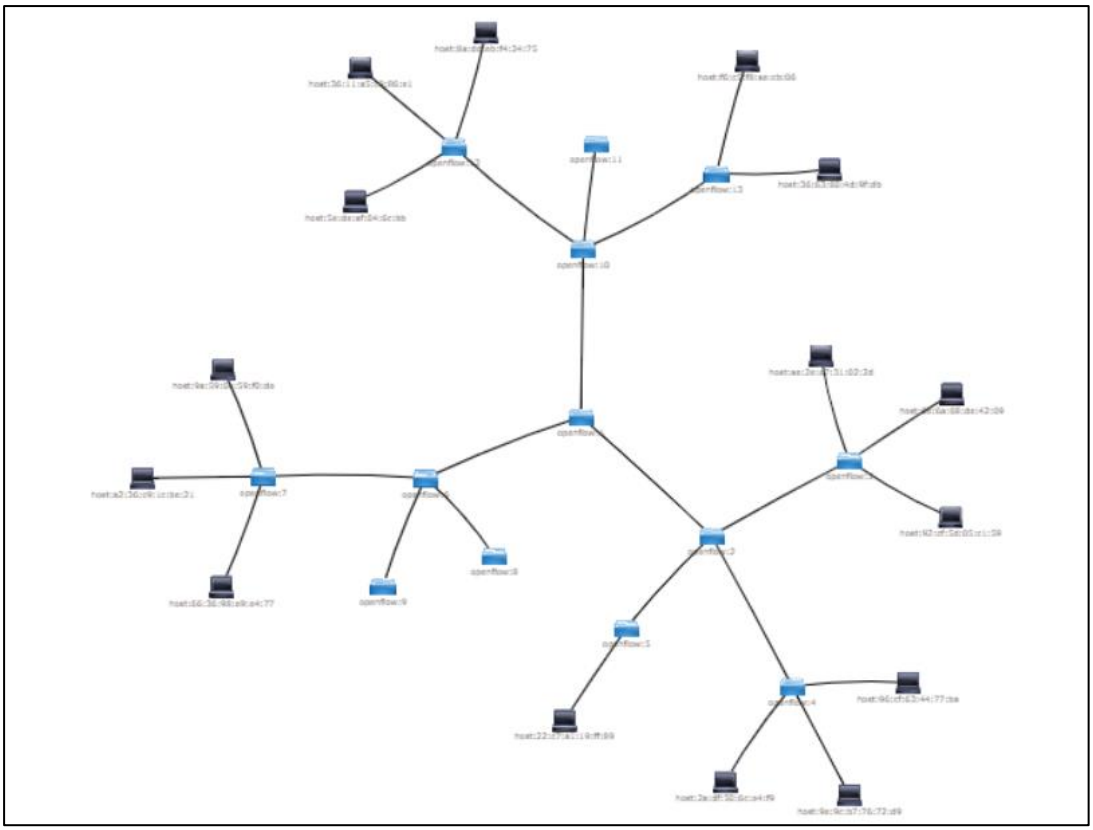


Figura 52. ***Red definida por software en Mininet***
Nota: Elaboración propia

Pruebas de Conectividad y Flujos Openflow

Mininet permite realizar pruebas dentro del entorno simulado a través de ping para el envío de paquetes icmp, pingall para verificar la conexión de toda la red (figura 53) y el comando iperf que a través del modelo cliente-servidor puede crear flujos openflow de extremo a extremo.

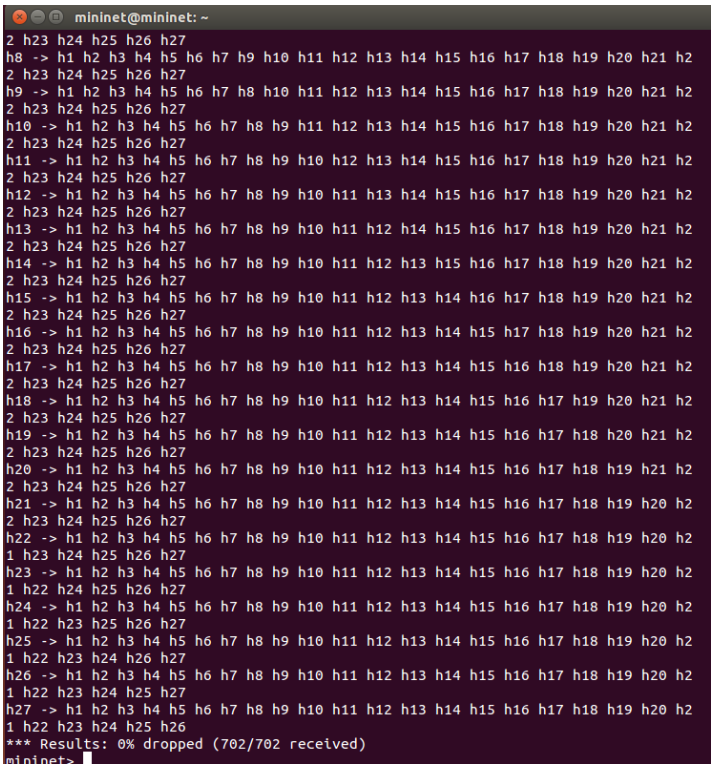


Figura 53. ***Conexión total de la red definida por software***
Nota: Elaboración propia

Adicional se realizaron pruebas con la herramienta IPERF que permitió probar el ancho de banda de tráfico UDP. Se configuró el host 1 (h1) como servidor que escuchará a través del puerto 8081 con el comando de la figura 54 y los hosts h5 y h10 como clientes (figura 55 y 56 respectivamente.)

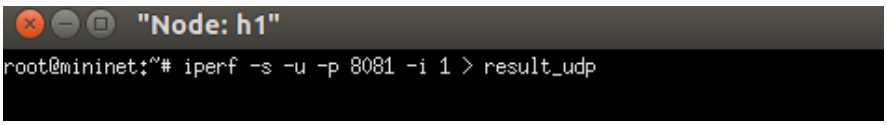


Figura 54. ***Configuración de host 1 como servidor udp***
Nota: Elaboración propia

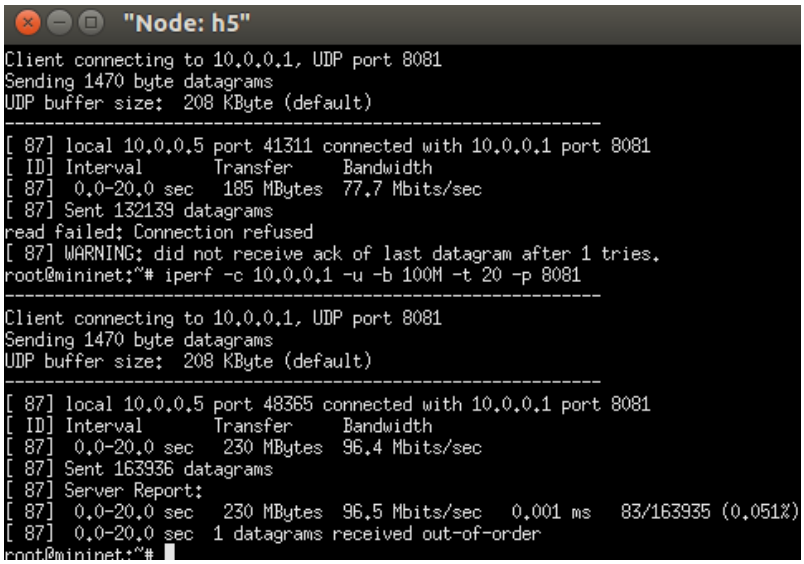


Figura 55. ***Prueba iperf con tráfico udp en host h5***
Nota: Elaboración propia

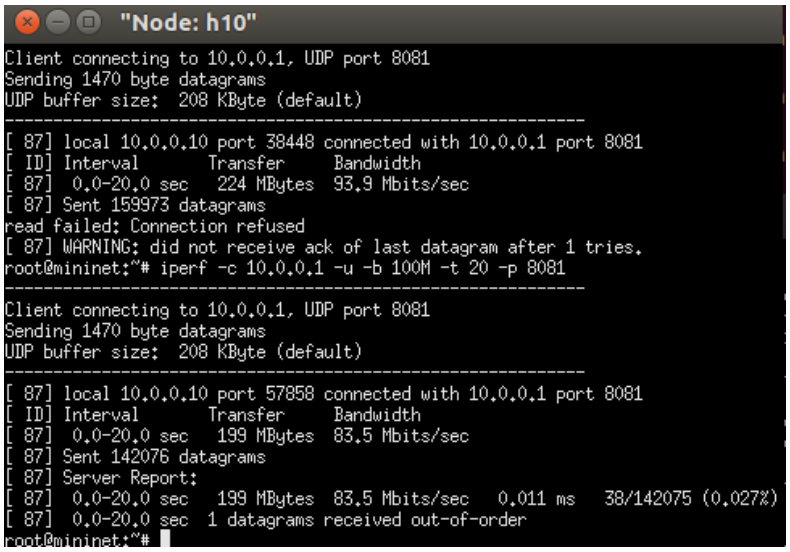


Figura 56. ***Prueba iperf con tráfico udp en host h10***
Nota: Elaboración propia

Convergencia SDN con Redes Tradicionales

BGP (Border Gateway Protocol) es el protocolo que permite la conectividad entre la red tradicional y la red definida por software, para ello se implementó el módulo bgp en el controlador SDN, de esta manera Opendaylight es capaz de almacenar los prefijos de red en la tabla RIB (Routing Information Base) (Chafloque, 2018).

Para el esquema de este escenario se utilizó un router cisco 7200 del lado de la red tradicional en el que se configuró bgp con los comandos de la figura 57.

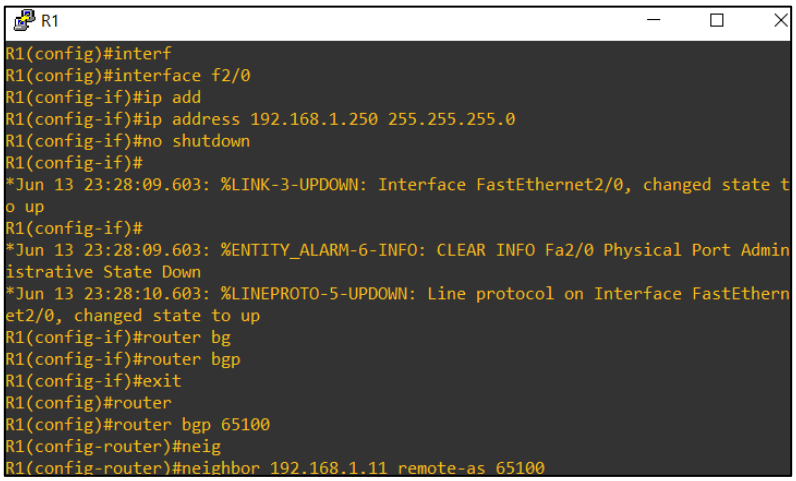


Figura 57. ***Configuración de BGP en router tradicional***
Nota: Elaboración propia

Como se puede observar en la figura 57 se utilizó 65100 como número de sistema autónomo, mismo que se debe configurar en el controlador ODL.

4.4. Aspectos Importantes para Considerar

4.4.1. Software Libre vs Hardware Propietario

Si bien es cierto, la presente investigación impulsa el uso de software libre, lo cual parecería que no requiere inversión, no obstante, se deben considerar costos de capacitación al personal de TI y tiempo invertido; en contraste se presentan también opciones en software propietario con soporte OpenFlow.

Es necesario hacer énfasis en que no se requieren switches exclusivamente SDN, lo ideal es utilizar equipamiento híbrido que permita el funcionamiento doble pila y la coexistencia de Capa 2 y Capa 3 sin obligar a las organizaciones a reemplazar abruptamente sus equipos de red.

Así, la ONF (Open Networking Foundation) propone un listado de fabricantes que proveen equipos de red híbridos aprobados para usar en ambientes tradicionales/sdn. La tabla 23 muestra los dispositivos sugeridos por los fabricantes destacados:

Tabla 23.
Productos aprobados por la ONF para el despliegue SDN

Item	Fabricante	Producto
1	Allied Telesis, Inc	AT-XS900MX 10Gb L3 stackable switches
2	Netvision Telecom Inc	NetOV-48X
3	Rujie Networks Co,. Ltd	RG-S8612E Rujie RG-S8600E
4	Huawei	S628X-E L2 Ethernet Sw 24 Ethernet 10/100/1000 Base-T,4 10GE SFP+
5	Hangzhou H3C	H3CS5560X-54AC-EI
6	ZTE Corporation	ZXR10 M6000-S
7	Hewlett Packard	HP 2920-24G

Nota: Elaboración propia. Tomado de (ONF,2018)

4.4.2. Soporte SDN/NFV

La diferencia más relevante entre las SDN y las redes tradicionales es que SDN se basa en software y las redes tradicionales en hardware; así SDN permite el aprovisionamiento de nuevos dispositivos a través de software en lugar de utilizar la infraestructura física, de ello se desprende que, aquellas organizaciones que tienen equipos obsoletos o descontinuados no podrían soportar una migración SDN/NFV, por cuanto los mismos equipos limitarían las técnicas de virtualización necesarias para la abstracción de recursos.

SDN se convirtió en una buena alternativa a las redes tradicionales puesto que permite a los administradores de TI suministrar recursos y anchos de banda según sea necesario sin requerir una inversión de infraestructura física adicional tomando en cuenta que sus equipos

soportan una configuración “híbrida”. Las redes tradicionales requieren hardware nuevo para aumentar su capacidad de red. El paradigma de SDN frente a las redes tradicionales trae consigo nuevos roles en cuanto a proveedores de internet, ahora serían: proveedor de servicios y proveedor de infraestructura.

En conclusión, la red definida por software (SDN) es un modelo de arquitectura de red que permite la gestión, el control y la optimización programáticos de los recursos de la red.

SDN desacopla la configuración de red y la ingeniería de tráfico de la infraestructura de hardware subyacente, para garantizar un control holístico y consistente de la red mediante API abiertas. La arquitectura de red tradicional ofrece una flexibilidad mínima para configuraciones de dispositivos de red. Un solo cambio puede tener un efecto en cascada sobre el rendimiento de la red y tiene el potencial de derribar toda la red.

Capítulo V

Conclusiones y Recomendaciones

5.1. Conclusiones

La arquitectura de una red definida por software (SDN) se basa en separar el plano de control del plano de datos, lo cual supone varias ventajas frente a una red tradicional, como una mayor flexibilidad y mejor orquestación de red. El despliegue completo de SDN requiere que todos los equipos de red heredados se encuentren actualizados; en la mayoría de casos esto representa una barrera, por lo que la solución óptima implica desplegar SDN de forma gradual como una red híbrida. En redes híbridas los dispositivos se pueden ir reemplazando parcialmente, siendo que, el controlador debe tener comunicación con el plano de control de los dispositivos heredados.

En ese sentido, se pudo mostrar que el primer escenario o esquema Greenfield resultó ser el menos complejo, por cuanto no hay necesidad de admitir la integración con una arquitectura de red existente que no esté basada en OpenFlow, en contraste con los esquemas híbridos, en donde los dispositivos existentes deben coexistir con los nuevos dispositivos OpenFlow. Los controladores OpenFlow y los equipos tradicionales necesitan intercambiar información de enrutamiento a través de un plano de control heredado.

De esta manera, es posible afirmar que en las redes definidas por software no existe diferenciación en cuanto a dispositivos de red, en contraste con las redes convencionales donde se hallan routers a nivel de red, o switches a nivel de enlace; ya que todos los equipos SDN son considerados “switches openFlow”, que se comunican a través de mensajes OpenFlow, mismos que utilizan criterios de capa enlace, red, transporte y aplicación según el modelo OSI.

En los escenarios planteados se visualiza el retardo que tiene el primer paquete ICMP de la red definida por software, esto se debe a que cuando se envía el primer mensaje, el router o switch desconoce como enrutarlo, entonces procede a encapsularlo en el protocolo OpenFlow y reenvía todo su contenido al controlador SDN (OpenDaylight en este caso), siendo éste el que se encarga de armar las tablas de flujo en cada switch OpenFlow; este procedimiento inaugural es el que introduce cierta latencia en el proceso de convergencia inicial; esta latencia se introduce por cada dispositivo openFlow, es por ello que se incrementa a medida que aumentan los dispositivos de red; no obstante para los icmps posteriores, esta latencia se va reduciendo por cuanto los flujos iniciales ya se almacenan en la memoria del controlador SDN quien se encarga de enrutar las tramas optimizando el rendimiento de red.

Dicho de otra manera, cuando la red va de pequeña a mediana escala, el mecanismo de enrutamiento es más rápido en una red tradicional, sin embargo, a medida que la red se expande SDN va tomando ventaja por cuanto SDN no requiere transmitir información entre switches sino solamente del enlace que necesita, logrando una convergencia superior frente a la red legacy.

El presente documento constituye una metodología práctica que permitirá orientar las redes tradicionales hacia escenarios inteligentes, surgió como respuesta a las demandas actuales de las tecnologías de información y las telecomunicaciones avizorando entornos definidos por software como mecanismo principal. Los procedimientos descritos en el presente trabajo de investigación son alentadores por cuanto evidencian el potencial que pueden alcanzar las redes definidas por software en aspectos relevantes como escalabilidad y tiempo de convergencia resultando de vital importancia la simulación de escenarios que permitieron evaluar el comportamiento real de las redes.

5.2. Recomendaciones

Para construir una infraestructura de TI altamente receptiva, resistente y automatizada se recomienda seguir las mejores prácticas de virtualización, agregar una capa de containers y microservicios, virtualizando firewalls, enrutadores u otros dispositivos que aporten agilidad a la capa red, finalmente se puede agregar SDN para su administración y por supuesto, realizar las pruebas de validación necesarias.

Para la aplicación de la metodología propuesta es indispensable evaluar el estado real de la red a la cual se va aplicar la transición, no obstante, se presentaron algunos ejemplos, en donde la idea principal es desplegar equipos de red híbridos o doble pila que permitan la interoperabilidad entre arquitecturas de redes tradicionales y redes definidas por software. Los tiempos de retardo (delays) y tiempos de convergencia pueden variar la ejecución con equipos reales, sin embargo como se utilizaron kernels reales, la proporción seguirá siendo la misma como se indica en las pruebas de desempeño presentadas.

El presente trabajo de investigación se llevó a cabo sobre una red virtualizada por lo que resultaría factible en trabajos futuros, utilizar equipos físicos con soporte para el protocolo OpenFlow, que permitan evaluar el comportamiento en un entorno real realizando varias pruebas para diferentes tipos de tráfico.

Referencias

Artmann, D., & Khondoker, R. (2018). Security Analysis of Software-Defined Networking and Network Function Virtualization. In *Lecture Notes in Networks and Systems* (Vol. 30). https://doi.org/10.1007/978-3-319-71761-6_4

Azarov, V. N., Saksonov, E. A., & Leokhin, Y. L. (2018). Analysis of Information Structure of the Corporate Network of Enterprise. *Proceedings of the 2018 International Conference "'Quality Management, Transport and Information Security, Information Technologies'", IT and QM and IS 2018*, 9–12. https://doi.org/10.1109/ITMQIS.2018.8524906

Báez, M. G., & Brunner, S. I. B. (n.d.). *Metodología DoRCU para la Ingeniería de Requerimientos*. 210–222.

Balaji Sivasubramanian, Erum Frahim, R. F. (2010). Analyzing the Cisco Enterprise Campus Architecture. Retrieved from Cisco Press website: https://www.ciscopress.com/articles/article.asp?p=1608131&seqNum=3

Basu, S. (2017). Evaluation of Data Centre Networks and Future Directions Sohini Basu. *Thesis*, (September).

Ben-Asher, N., & Gonzalez, C. (2015). Effects of cyber security knowledge on attack detection. *Computers in Human Behavior*, *48*, 51–61. https://doi.org/10.1016/j.chb.2015.01.039

Cedeño Rodríguez, Y. V. (2017). *EVALUACIÓN DE LA INFRAESTRUCTURA TECNOLÓGICA BASADO EN ESTÁNDARES DE CONTROL INTERNO CASO: EMPRESA NATIONALTIRE EXPERTS S.A.*

Chafloque, J. (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. *Lima-Perú*, 124. Retrieved from https://hdl.handle.net/20.500.12672/10017

CISCO. (2017). Cisco Plug-in for OpenFlow Configuration Guide 1.3. Retrieved from https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus/openflow/b_openflow_agent_nxos_1_3/Cisco_Plug_in_for_OpenFlow.html

CISCO. (2018). Arquitectura de redes empresariales Cisco ONE: una base automatizada de reconocimiento de aplicaciones para la empresa moderna. Retrieved from https://www.cisco.com/c/dam/global/es_mx/assets/ofertas/catalyst/pdfs/en_white_paper_wp_cte_es_xl_37467.pdf

Community Cisco. (2015). *Network Design Requirements : Analysis and Design Principles*. 3–29. Retrieved from https://www.iso.org/isoiec-27001-information-security.html

Cordero, C. F. (2017). *Diseño y Despliegue de Funciones de Red Virtualizadas (NFV) usando Redes Definidas por Software (SDN) dentro de una infraestructura Virtual, aplicando balanceo de carga y seguridad distribuida en IPv6*.

Durairaj, M., & Kannan, P. (2014). A Study On Virtualization Techniques And Challenges In Cloud Computing. *International Journal of Scientific & Technology Research*, *3*(11), 147–151.

E. Rosen; Cisco Systems, Inc; A. Viswanathan; Force10 Networks, I. (2001). Request for Comments: 3031. Retrieved from https://www.ietf.org/rfc/rfc3031.txt

Errera, D. W. (2014). OpenFlow-enabled SDN and Network Functions Virtualization. *AORN Journal*, *23*(6). https://doi.org/10.1016/S0001-2092(07)70284-3

ETSI. (2014). Network Functions Virtualisation (NFV); Virtual Network Functions

Architecture. *Etsi Gs Nfv*, *1*, 1–50.

ETSI. (2019). *ETSI Technologies NFV*. Retrieved from https://www.etsi.org/technologies/nfv

Franco García, F. A. (2019). Diseño de la infraestructura de redes para la mejora de la comunicación de datos en la empresa SEAFROST fundamentado en la norma TIA/EIA-942A. *Angewandte Chemie International Edition, 6(11), 951–952.*, 5–24.

Galarza-Macancela, C. V. (2018). Diseño e implementación de una red de datos segura para la Pontificia Universidad Católica del Ecuador, Santo Domingo. *Dominio de Las Ciencias*, *4*(2), 123. https://doi.org/10.23857/dc.v4i2.781

García Ibáñez, J. F. (2016). *Estudio de las tecnologías SDN y NFV*.

García Molinero, J. (2015). *Proyecto De Red Informática Corporativa Para Empresa Comercializadora De Electricidad*.

Gns3. (2020). GNS3. Retrieved from https://www.gns3.com/software

Guerra, V. H. (2020). Diseño e Implementación de la red de datos del laboratorio centro de desarrollo de software y productos IOT de la facultad de ingeniería de la Universidad Católica de Santiago de Guayaquil. *Angewandte Chemie International Edition, 6(11), 951–952.*, 5–24.

Hakiri, A., Gokhale, A., Berthou, P., Schmidt, D. C., & Gayraud, T. (2015). Software-defined networking: Challenges and research opportunities for future internet. *Computer Networks*, *75*(PartA), 453–471. https://doi.org/10.1016/j.comnet.2014.10.015

Hassan, A., Mondol, R. K., & Hasan, M. R. (2015). Computer network design of a company-A simplistic way. *ICACCS 2015 - Proceedings of the 2nd International Conference on Advanced Computing and Communication Systems*, (January 2015), 3–7. https://doi.org/10.1109/ICACCS.2015.7324121

Hat, R. (2019). What is virtualization? Retrieved from https://opensource.com/resources/virtualization

Hernandez Valencia, E., Izzo, S., & Polonsky, B. (2015). How will NFV/SDN transform service provider OpEx? *IEEE Network*, *29*(3), 60–67. https://doi.org/10.1109/MNET.2015.7113227

Horvath, I. (2021). What is COBIT 5 Framework? Retrieved from https://www.invensislearning.com/blog/what-is-cobit-5/

IETF. (1981). Internet Protocol. Retrieved from https://tools.ietf.org/html/rfc791

Internet Engineering Task Force. (2010). YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF). Retrieved from https://tools.ietf.org/html/rfc6020

Jammal, M., Singh, T., Shami, A., & Li, Y. (2018). Software-Defined Networking : State of the Art and Research Challenges. *Elsevier's Journal of Computer Networks*, 1–24.

Ji, J., Xing, F.-F., & Zang, Y.-Q. (2015). *The Design of Enterprise Network Information Sharing Scheme Based on Security Technology*. (Icitmi), 623–626. https://doi.org/10.2991/icitmi-15.2015.102

Kim, H. K., So, W. H., & Je, S. M. (2019). A big data framework for network security of small and medium enterprises for future computing. In *Journal of Supercomputing* (Vol. 75). https://doi.org/10.1007/s11227-019-02815-8

Libardo Niño, C., & Vento Serrato, C. A. (2016). *Diseño de una Red LAN para la Empresa INTEL CORP* (Vol. 66).

Mecías Serrano, Í., Molina, L. J., & Zúñiga, A. R. (2019). Diseño de una red de datos para el mejoramiento de la gestión de comunicación interna en UNIANDES Quevedo. *Journal*

of Wind Engineering and Industrial Aerodynamics, *26*(3), 1–4. Retrieved from https://doi.org/10.1007/s11273-020-09706-3%0Ahttp://dx.doi.org/10.1016/j.jweia.2017.09.008%0Ahttps://doi.org/10.1016/j.energy.2020.117919%0Ahttps://doi.org/10.1016/j.coldregions.2020.103116%0Ahttp://dx.doi.org/10.1016/j.jweia.2010.12.004%0Ahttp://dx.doi.o

Miranda Moreira, C. M. (2015). *Diseño E Implementación Del Esquema De Seguridad Perimetral Y Monitoreo Para La Red De Datos En Una Empresa Industrial*. 1–93.

Mtumbuka, C. M. (2020). *THE IMPACT OF CLOUD COMPUTING IN REGARD TO COST AND SECURITY . A CASE OF VODACOM TANZANIA PUBLIC LIMITED COMPANY By CORRINNE MARY MTUMBUKA A Dissertation Submitted in Partial Fulfilment of the Requirements for the Award of Masters for Information Technolo*.

Open Networking Foundation. (2014). SDN Migration Considerations and Use Cases. *ONF Whitepapaer*. Retrieved from https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-sdn-migration-use-cases.pdf

Open Networking Foundation. (2021). Open Networking Foundation. Retrieved from https://opennetworking.org/mission/

Oroya Acosta, M. F. (2019). *Rediseño de la Red LAN en la Empresa VLACAR S.A.C.*

Osaba, M. N. (2016). VIRTUALIZACIÓN EN REDES DEFINIDAS POR SOFTWARE. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Owens, H. (2016). *Provisioning End-To-End Quality of Service for Real-Time Interactive Video Over Software-Defined Networking*.

Pérez-Virgen, H. L., Salamando-Mejía, C. A., & Valencia-Ayala, L. S. (2013). Levantamiento De Requerimientos Basados En El Conocimiento Del Proceso. *Revista Científica*, *2*(16), 42. https://doi.org/10.14483/23448350.4022

Rajendra, C., & Syed Farrukh, H. (2016). *Network Functions Virtualization (NFV) with a Touch of SDN*.

Ramirez, M., & Lopez, A. (2018). Redes de datos definidas por software - SDN, arquitectura, componentes y funcionamiento. *Journal de Ciencia e Ingeniería*, *10*(2145–2628), 55–61. Retrieved from https://bit.ly/2DEBYVP

Ramiro, R. (2017). *Seguridad en las redes definidas por software*. Retrieved from https://ciberseguridad.blog/seguridad-en-las-redes-definidas-por-software-sdn/

Rekhter, Y. (2006). A Border Gateway Protocol. Retrieved from https://datatracker.ietf.org/doc/html/rfc4271

Roncancio R, Ginna & Sáenz G, C. (2016). Propuesta de implementación de las tecnologías NFV y SDN y su utilización en la red de comunicaciones (CASO DE ESTUDIO UTM). *IOSR Journal of Economics and Finance*, *3*(1), 56. https://doi.org/https://doi.org/10.3929/ethz-b-000238666

Safe Cisco. (2015). *SAFE Overview Guide*. (November).

Saigushev, N. Y., Mikhailova, U. V., Vedeneeva, O. A., & Tsaran, A. A. (2018). Information Systems at Enterprise. Design of Secure Network of Enterprise. *Journal of Physics: Conference Series*, *1015*(4). https://doi.org/10.1088/1742-6596/1015/4/042054

Salih, R., & Alsarhan, S. (2016). *Computer Network Design for Universities in Developing Countries*. (2). Retrieved from http://scholar.valpo.edu/itcrprhttp://scholar.valpo.edu/itcrpr/2

Salman, O., Elhajj, I. H., Kayssi, A., & Chehab, A. (2016). SDN controllers: A comparative study. *Proceedings of the 18th Mediterranean Electrotechnical Conference: Intelligent and Efficient Technologies and Services for the Citizen, MELECON 2016*, (978), 18–20. https://doi.org/10.1109/MELCON.2016.7495430

Shao Ying Zhu, Sandra Scott-Hayward, Ludovic Jacquin, R. H. (2017). *Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications*.

Singh, S. (2019). *Network requirement analysis and plan*. (October), 0–12. https://doi.org/10.13140/RG.2.2.24957.31203

Smith, M. (2018). Serenity Networks. Retrieved from How to Install Openstack Ocata on a Single Server, Using Devstack website: https://serenity-networks.com/how-to-install-openstack-ocata-on-a-single-server-using-devstack/

Spera, P. C., Development, B., Logicalis, M., & Cone, S. (2014). *Software Defined Network: el futuro de las arquitecturas de red*. 42–45.

Tannady, H., Andry, J. F., Sudarsono, B. G., & Krishartanto, Y. (2020). Enterprise Architecture Using Zachman Framework at Paint Manufacturing Company. *Technology Reports of Kansai University*, *62*(4), 1869–1883.

Torres, A. (2020). ¿Qué es el hosting, para que sirve y qué tipos hay? Retrieved from https://www.comparahosting.com/p/que-es-hosting/#¿que-es-un-hosting?

Triana, J. (2017). *Estado del Arte sobre SDN y NFV*.

Vaidy A, Guru Kannan, R. R. (2020). Simplify SD-WAN connectivity with AWS Transit Gateway Connect. Retrieved from https://aws.amazon.com/es/blogs/networking-and-content-delivery/simplify-sd-wan-connectivity-with-aws-transit-gateway-connect/

Verdezoto, B. (2015). *Diseño de una red LAN para el transporte de voz, datos y video para el Municipio del cantón Valencia Provincia de Los Ríos.* 1–164. Retrieved from http://bibdigital.epn.edu.ec/handle/15000/10312

ANEXOS

Anexo A: Entrevista realizada a líderes de departamentos tecnológicos

ITEM	PREGUNTA	SÍ	NO	OBS
1	¿El departamento de tecnología cuenta con Políticas establecidas de TI?			
2	¿Existe un manual de procesos exclusivo para el Departamento de TI?			
3	¿Cuenta con una infraestructura dedicada a los equipos tecnológicos? (Centros de datos, cuarto de equipos, cuarto de telecomunicaciones)			
4	¿Cuenta con un inventario tecnológico de los equipos que maneja el departamento?			
5	¿Existen procesos específicos para cada integrante del departamento de TI? (Mantenimiento, Desarrollo, Networking, entre otros)			
6	¿Tiene el control total de gestión de cambios de su red de datos?			
7	¿Ha estado en riesgo de perder información sensible a causa de fallos físicos en sus equipos?			
8	¿Puede monitorear su red de datos proactivamente?			
9	¿Dispone de fuentes de alimentación con redundancia o sistemas de energía ininterrumpida?			
10	¿La red de datos de su empresa está preparada para el crecimiento de aquí en 10 años ?			
11	¿El Departamento de TI se encarga de la gestión de hardware de red cuando se lo requiere o lo envían a contratistas?			
12	¿Su empresa dispone de algún proyecto de innovación destinado a la mejora tecnológica incluyendo un presupuesto destinado para su financiación?			
13	¿Cuáles son los servicios y aplicaciones que utilizan en su empresa? Web, correo, entre otros.	N/A	N/A	……..

UNIVERSIDAD TÉCNICA DEL NORTE

INSTITUTO DE POSGRADO

REPORTE DE ENTREVISTA DIRGIDA HACIA LÍDERES DE DEPARTAMENTOS DE TI

El siguiente reporte de entrevista se presenta únicamente con fines investigativos y constituye parte fundamental del presente trabajo, por temas de seguridad y confidencialidad se ha resguardado la identidad de los entrevistados.

Desarrollo:

1. **¿El departamento de tecnología cuenta con Políticas establecidas de TI?**
   No exclusivamente, pero existe un plan de contingencia en el cual, como departamento hemos abordado algunas temáticas. Este plan de contingencia está en proceso de aprobación.

2. **¿Existe un manual de procesos exclusivo para el Departamento de TI?**
   No, pero en el manual de procedimientos de la empresa se detalla el departamento de TI.

3. **¿Cuenta con una infraestructura dedicada a los equipos tecnológicos? (Centros de datos, cuarto de equipos, cuarto de telecomunicaciones)**
   No, los equipos se encuentran en un rack en la oficina de TI

4. **¿Cuenta con un inventario tecnológico de los equipos que maneja el departamento?**
   Sí, un archivo con macros en el que se registra manualmente los datos de los equipos.

5. **¿Existen procesos específicos para cada integrante del departamento de TI? (Mantenimiento, Desarrollo, Networking, entre otros)**
   El personal de TI realiza múltiples funciones, no existe una segmentación.

6. **¿Tiene el control total de gestión de cambios de su red de datos?**
   No, el proveedor de internet maneja las configuraciones de alto nivel.

7. **¿Ha estado en riesgo de perder información sensible a causa de fallos físicos en sus equipos?**
   Sí, por ello se realizan procedimientos de backup diarios, no obstante, su restablecimiento provoca indisponibilidad por varias horas.

8. **¿Puede monitorear su red de datos proactivamente?**
   Hasta hace algunos meses sí lo hacíamos mediante PRTG no obstante la empresa decidió no renovar la licencia.

9. **¿Dispone de fuentes de alimentación con redundancia o sistemas de energía ininterrumpida?**
   Debido a inconvenientes pasados, se realizó la configuración de un UPS de 6000Kva para protegernos en caso de cortes de energía.

UNIVERSIDAD TÉCNICA DEL NORTE

INSTITUTO DE POSGRADO

REPORTE DE ENTREVISTA DIRGIDA HACIA LÍDERES DE DEPARTAMENTOS DE TI

10. **¿La red de datos de su empresa está preparada para el crecimiento de aquí en 10 años ?**
    Personalmente pienso que se deben realizar algunos cambios para que el soporte de networking funcione por 10 años, desde el tiempo que llevo aquí 6 años aproximadamente, no se ha tomado demasiada importancia al departamento.

11. **¿El Departamento de TI se encarga de la gestión de hardware de red cuando se lo requiere o lo envían a contratistas?**
    Nosotros nos encargamos del proceso de cotización y compra con la aprobación del Gerente.

12. **¿Su empresa dispone de algún proyecto de innovación destinado a la mejora tecnológica incluyendo un presupuesto destinado para su financiación?**
    Cada inicio de año contable se presenta la cotización del presupuesto requerido, pero no por varios factores no llega a darse.

13. **¿Cuáles son los servicios y aplicaciones que utilizan en su empresa? Web, correo, entre otros.**
    Servidor de aplicaciones del sistema interno, base de datos, correo electrónico y transferencia de archivos.

Para constancia firma:

Firma o sello de la empresa

ENTREVISTADO

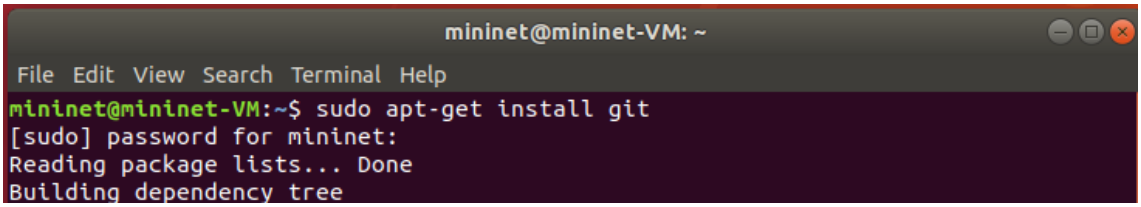
Marcela Elizabeth López

ENTREVISTADOR

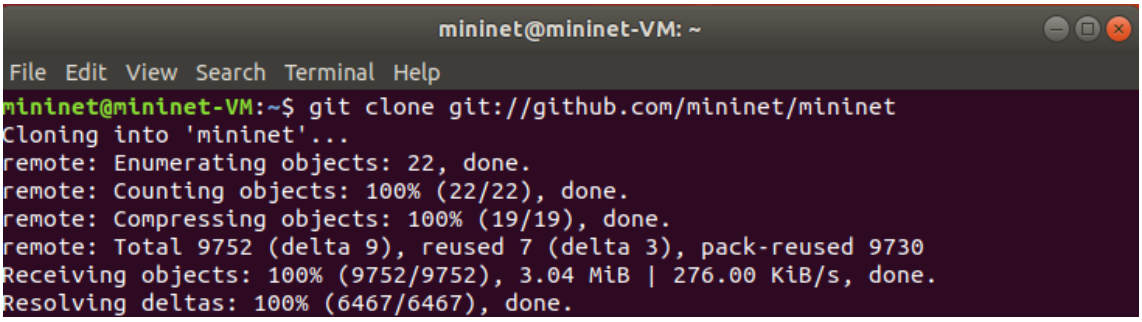
Anexo B: Instalación de Mininet

En la web oficial de mininet http://mininet.org/download/ se presenta el procedimiento para su instalación que puede realizarse de cuatro formas: máquina preempaquetada, nativa, por paquetes y actualización a partir de una versión existente. En este caso se llevó a cabo una instalación nativa desde la fuente con los siguientes pasos:

- Instalar git para clonar el repositorio.



- Obtener el código fuente desde github.



- Con el comando *git tag* se muestran las versiones disponibles y se procede a seleccionar la versión deseada, en este caso la 2.2.1.

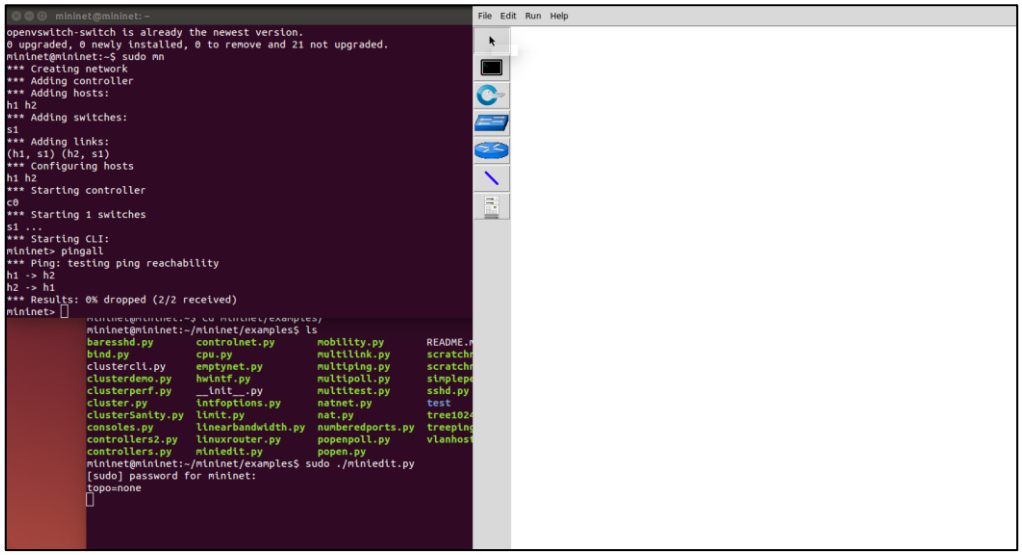
- Una vez listo el árbol de origen se instala Mininet con todo lo que incluye la VM a través del comando *mininet/util/install.sh –a.*

```
mininet@mininet-VM:~$ sudo mininet/util/install.sh -a
Detected Linux distribution: Ubuntu 18.04 bionic amd64
sys.version_info(major=3, minor=6, micro=9, releaselevel='final', serial=0)
Detected Python (python3) version 3
Installing all packages except for -eix (doxypy, ivs, nox-classic)...
Install Mininet-compatible kernel if necessary
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Hit:2 http://ec.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://ec.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metada
ta [38,6 kB]
Get:5 http://ec.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Me
tadata [42,1 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe DEP-11 64x64 Ic
ons [99,7 kB]
```

Si la instalación fue correcta se prueba conectividad con el comando *sudo mn –test pingall* que creará la topología por defecto y realizará ping entre los hosts.

```
mininet@mininet-VM: ~
File  Edit  View  Search  Terminal  Help
mininet@mininet-VM:~$ sudo mn --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.899 seconds
```

Mininet trae consigo la aplicación GUI de miniedit, la cual se encuentra dentro del directorio /mininet/examples y se ejecuta con *sudo ./miniedit.py*



Anexo C: Instalación de OpenDaylight

OpenDaylight (ODL) puede instalarse a través de los repositorios Github o mediante el archivo comprido.

El primer paso será preparar el sistema operativo para recibir los paquetes y aplicaciones actualizados. Esto se realiza mediante los comandos *sudo apt-get update y sudo apt-get upgrade.*

En este caso, se procederá a instalar ODL mediante el archivo ZIP, para ello es necesario ejecutar la versión java 8, instalando JRE. El comando *sudo apt-get -y install openjdk-8-jre* permite instalar java JRE.

```
mininet@mininet-VM:~$ sudo apt-get -y install openjdk-8-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libgif7 openjdk-8-jre-headless
```

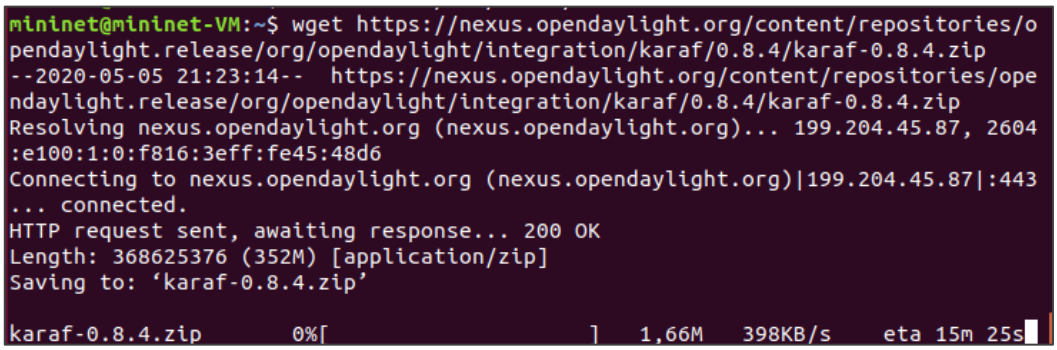
Lo siguiente será configurar la variable de entorno para que el servidor apunte a java 8, en este punto se actualiza el archivo BASHRC.

```
mininet@mininet-VM:~$ echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/j
re'>>~/.bashrc
mininet@mininet-VM:~$ source ~/.bashrc
mininet@mininet-VM:~$ echo $JAVA_HOME /usr/lib/jvm/java-8-openjdk-amd64/jre/usr/
lib/jvm/java-8-openjdk-amd64/jre /usr/lib/jvm/java-8-openjdk-amd64/jre
/usr/lib/jvm/java-8-openjdk-amd64/jre /usr/lib/jvm/java-8-openjdk-amd64/jre/usr/
lib/jvm/java-8-openjdk-amd64/jre /usr/lib/jvm/java-8-openjdk-amd64/jre
mininet@mininet-VM:~$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64/jre
mininet@mininet-VM:~$
```

A continuación, se deberá descargar el archivo comprimido del repositorio, para ello se creará un directorio donde se alojará la descarga mediante el comando *mkdir.*

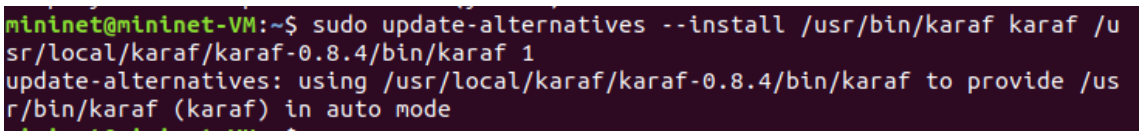
```
mininet@mininet-VM:~$ sudo mkdir /usr/local/karaf
```

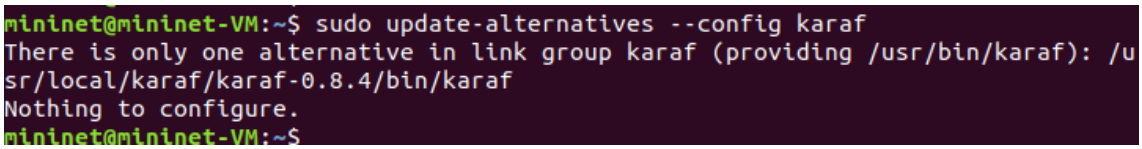
Ahora, con el comando wget se procederá a realizar la descarga de karaf en formato comprimido .zip.



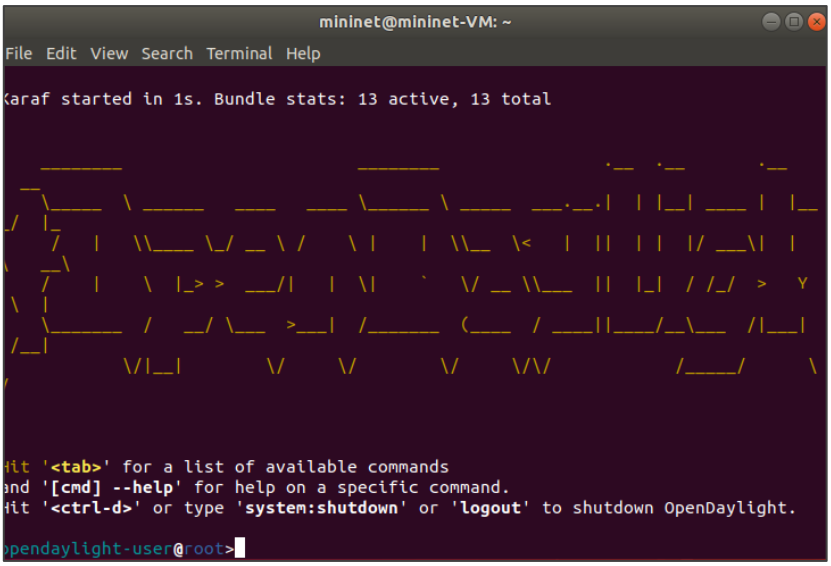
Luego, se mueve el archivo zip y se descomprime, para el presente trabajo de investigación se utilizó la versión 0.8.4.

Con el comando *sudo update-alternatives --install /usr/bin/karaf karaf /usr/local/karaf/karaf-0.8.4/bin/karaf 1* se procede a instalar karaf.

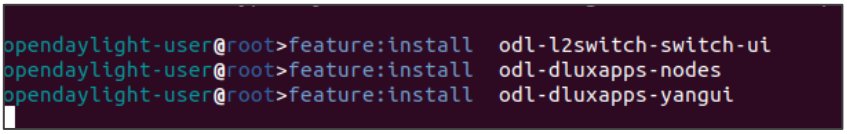




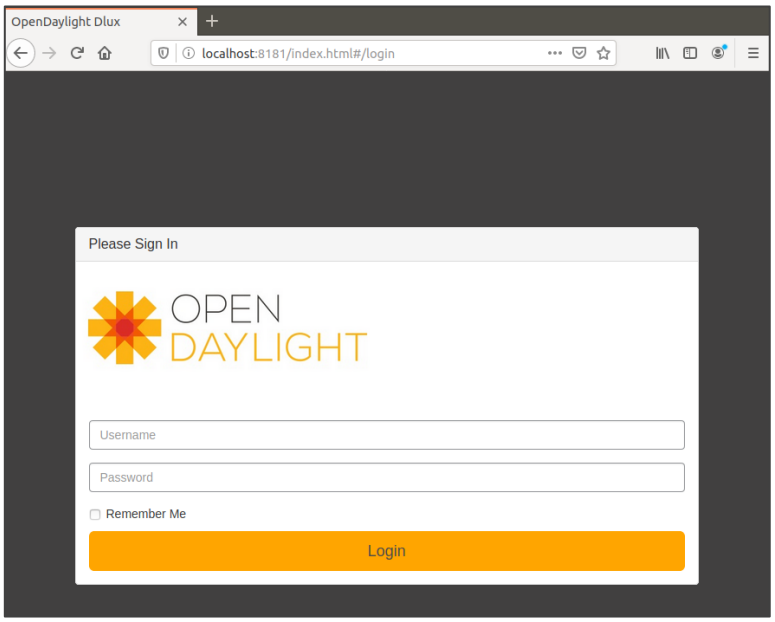
OpenDaylight necesita escribir un archivo PID en */usr/bin /karaf*, lo que requiere privilegios sudo. Se debe ejecutar el comando karaf a través de sudo y la bandera -E para mantener la variable de entorno $ JAVA_HOME .



ODL tiene otras aplicaciones que no aparecerán mientras no se las habilite, para ello se pueden utilizar los comandos a continuación e instalar las funciones deseadas.

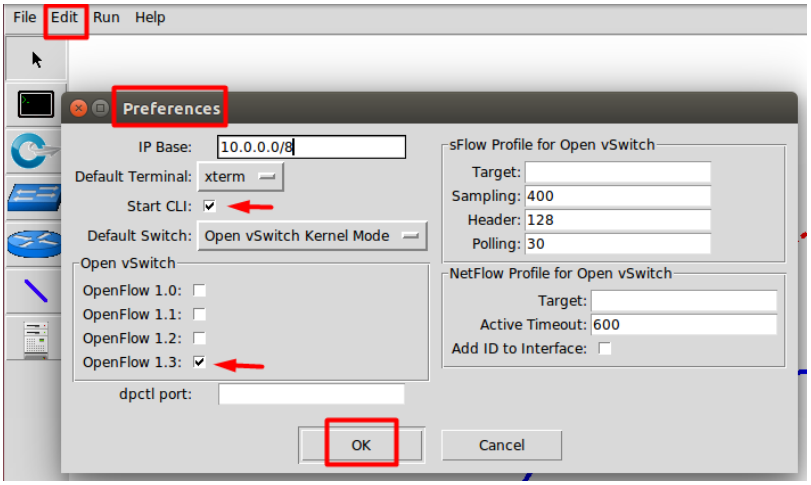


Finalmente para iniciar sesión bastará con escribir http: //<your-karaf-ip>:8181 en el navegador web y listo. El usuario y contraseña por defecto es *admin*.



Anexo D: Integración de OpenDaylight con Mininet

Miniedit permite modificar varios parámetros desde la opción “Edit”, para el presente trabajo se habilitó la opción “Start CLI” a fin de extraer un command line para cada dispositivo de red, se cambió la versión de OpenFlow a 1.3 y la red 10.0.0.0/8 corresponde al recurso utilizado (veáse figura xxx):

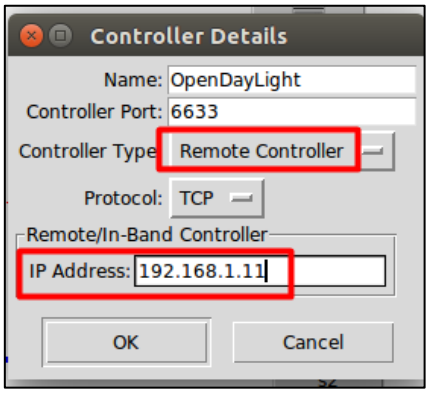


Para lograr la integración del controlador OpenDaylight en Mininet se debe colocar la dirección ip del controlador en la topología creada en Mininet, así por ejemplo en la figura a continuación se tiene la línea de comandos:

```
mininet@mininet-VM:~$ sudo mn --controller=remote,ip=192.168.1.8,port=6633
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

En donde la ip 192.168.1.8 y el port 6633 corresponden a la dirección ip del controlador ODL y su puerto de salida, respectivamente.

En el caso de utilizar miniedit para crear topologías personalizadas es necesario configurar en la interfaz gráfica haciendo click derecho sobre el controlador OpenDaylight. Se indica la configuración en la figura xxx:

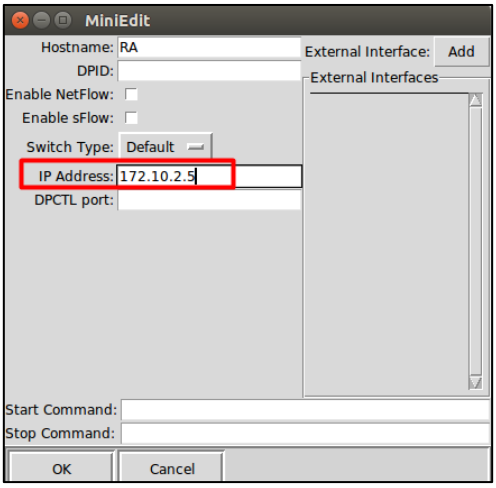


En “Controller type” se especifica que es un controlador remoto, la ip que tiene asignada y el puerto 6633 por defecto. Una vez realizadas las configuraciones previas, ya se puede empezar a levantar la topología deseada. Mientras se realizan configuraciones, estas se traducen a la terminal y se pueden observar los cambios.

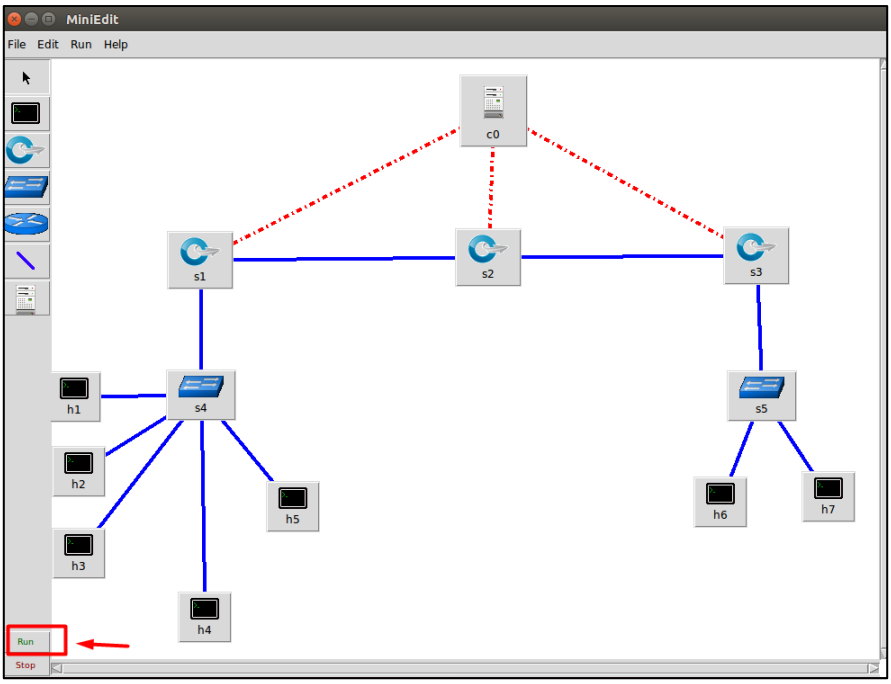
Anexo E: Creación de topologías personalizadas en mininet

Dentro del directorio /mininet/examples se arranca miniedit con *sudo ./miniedit.py*, y se accede a la interfaz gráfica en la cual, basta con arrastrar los componentes y crear la topología deseada.

Lo siguiente consiste en agregar el direccionamiento ip a los dispositivos, para ello se hará click derecho sobre el dispositivo y propiedades, allí se debe configurar la dirección ip en el campo "ip address", este procedimiento deberá realizarse en cada host:



Una vez que se hayan conectado todos los dispositivos requeridos en el área de trabajo, ya se puede arrancar la red haciendo click en "Run". Se mostrará en la consola la configuración realizada, y de no existir errores ya estará lista para analizar. La figura xxx muestra como arrancar la topología diseñada:



Se recomienda guardar la topología en formato .py para analizarla posteriormente y ejecutarla con el comando *sudo python mitopología.py*.

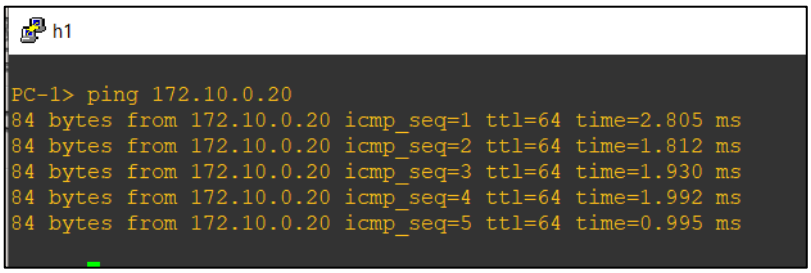
```
mininet@mininet: ~/mininet/examples
mininet@mininet:~$ cd mininet/examples/
mininet@mininet:~/mininet/examples$ sudo python caso1.py
[sudo] password for mininet:
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h5 h2 h4 h1 h6 h3 h7
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet>
```

Anexo F: Escenario Greenfield

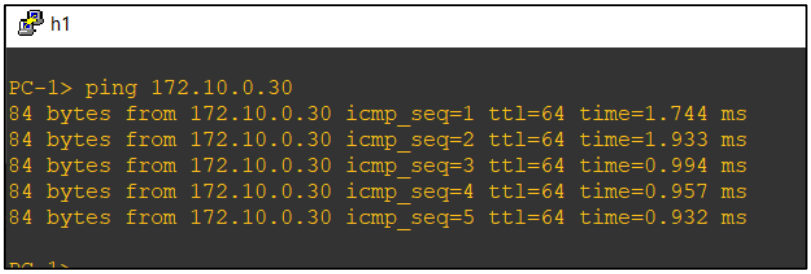
F.1. Pruebas de configuración red tradicional MPLS/IP

Paquetes icmp exitosos entre hosts de la subred Matriz.

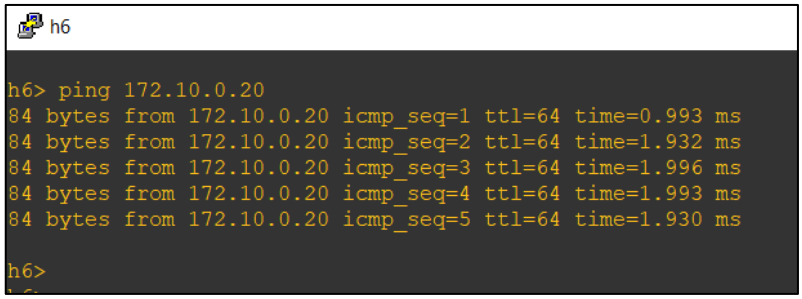
h1 a h2



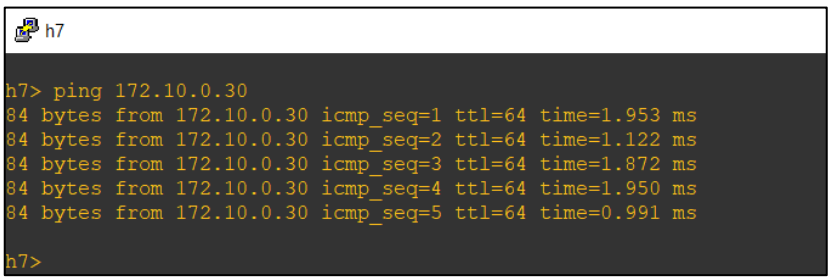
h1 a h3



h6 a h2

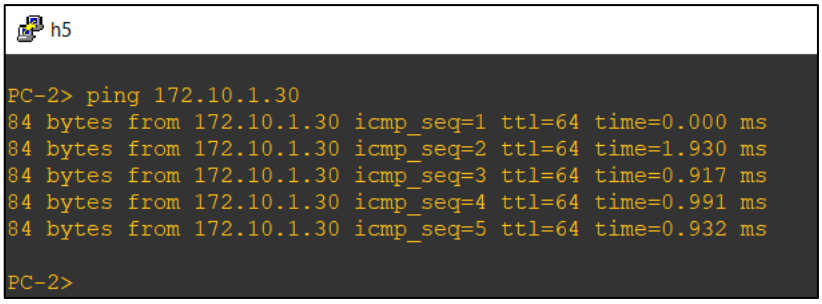


h7 a h3



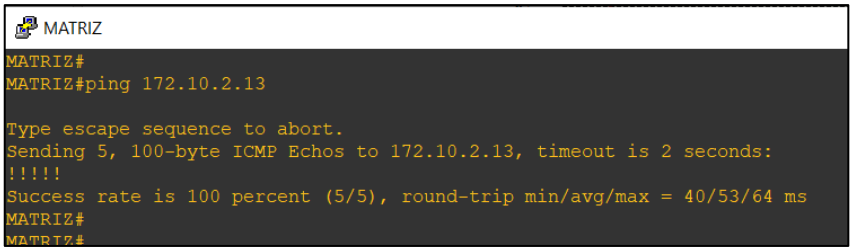
Paquetes icmp exitosos entre hosts de la subred de Sucursales.

h5 a h4

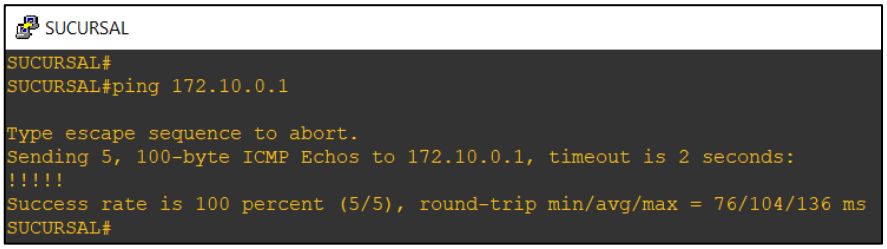


Paquetes icmp exitosos entre routers de la conexión WAN.

Matriz a sucursales



Sucursales a matriz



F.2. Pruebas de desempeño red tradicional

Para la recolección de data, se tomaron en cuenta muestras de paquetes icmp con diferentes tamaños de trama que van desde los 64bytes, 512 bytes, 1518 bytes hasta 4096 bytes como se indica en el RFC 2544.

El comando utilizado se indica a continuación, en donde l representa el ta,maño del paquete en bytes y c el número de veces que se enviará la solicitud eco icmp.

ping 172.10.1.10 –l 64 –c 10

Latencia entre el host h1 a h2 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.0 ms	0.909 ms	0.4545 ms
512 bytes	0.0 ms	0.920 ms	0.46 ms
1518 bytes	0.0 ms	0.997 ms	0.4985 ms
Latencia entre el host h1 a h3 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.0 ms	0.944 ms	0.472 ms
512 bytes	0.0 ms	0.928 ms	0.464 ms
1518 bytes	0.904 ms	0.998 ms	0.951 ms
Latencia entre el host h6 a h2 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.0 ms	0.893 ms	0.4465
512 bytes	0.0 ms	0.918 ms	0.459 ms
1518 bytes	0.0 ms	0.999 ms	0.4995 ms
Latencia entre el host h7 a h3 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.0 ms	0.902 ms	0.451 ms
512 bytes	0.0 ms	0.908 ms	0.454 ms
1518 bytes	0.0 ms	0.996 ms	0.498 ms
Latencia entre el hosts de la subred sucursales			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.0 ms	0.911 ms	0.4555 ms
512 bytes	0.0 ms	0.909 ms	0.4545 ms
1518 bytes	0.874 ms	0.907 ms	0.8905 ms
Latencia entre hosts remotos			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	4.841 ms	7.710 ms	6.276 ms
512 bytes	5.654 ms	4.668 ms	5.161 ms
1518 bytes	4.534 ms	7.331 ms	5.932 ms

F.3. Script en Python

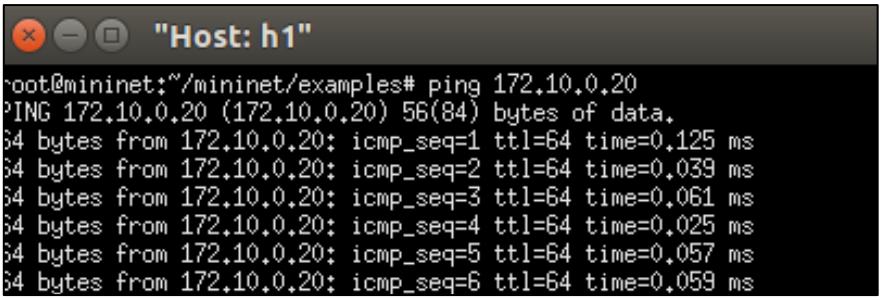
```
#!/usr/bin/python

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.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                   build=False,
                   ipBase='172.10.0.0/17')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                         controller=RemoteController,
                         ip='127.0.0.1',
                         protocol='tcp',
                         port=6633)

    info( '*** Add switches\n')
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch)
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch)
    s4 = net.addSwitch('s4', cls=OVSKernelSwitch, failMode='standalone')
    s5 = net.addSwitch('s5', cls=OVSKernelSwitch, failMode='standalone')
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch)

    info( '*** Add hosts\n')
    h5 = net.addHost('h5', cls=Host, ip='172.10.0.5', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='172.10.0.20', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='172.10.0.4', defaultRoute=None)
    h1 = net.addHost('h1', cls=Host, ip='172.10.0.10', defaultRoute=None)
    h6 = net.addHost('h6', cls=Host, ip='172.10.1.10', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='172.10.0.30', defaultRoute=None)
    h7 = net.addHost('h7', cls=Host, ip='172.10.1.20', defaultRoute=None)



    info( '*** Add links\n')
    net.addLink(s1, s2)
    net.addLink(s2, s3)
    net.addLink(s1, s4)
    net.addLink(s4, h4)
    net.addLink(s4, h5)
    net.addLink(s5, h6)
    net.addLink(s5, h7)
    net.addLink(s3, s5)
    net.addLink(h1, s4)
    net.addLink(s4, h2)
    net.addLink(h3, s4)

    info( '*** Starting network\n')
    net.build()
    info( '*** Starting controllers\n')
    for controller in net.controllers:
        controller.start()

    info( '*** Starting switches\n')
    net.get('s3').start([c0])
    net.get('s1').start([c0])
    net.get('s4').start([])
    net.get('s5').start([])
    net.get('s2').start([c0])

    info( '*** Post configure switches and hosts\n')
    s3.cmd('ifconfig s3 172.10.2.13')
    s1.cmd('ifconfig s1 172.10.2.5')
    s2.cmd('ifconfig s2 172.10.2.9')

    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNetwork()
```

F.4. Pruebas de configuración SDN

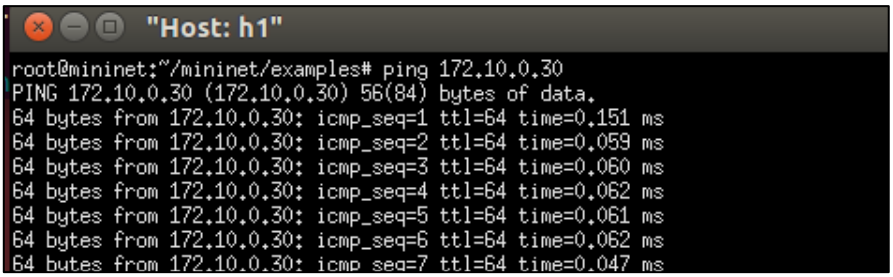
A continuación, se muestran las pruebas exitosas de conectividad entre los hosts de cada subred y la SDN.

Paquetes icmp exitosos entre hosts de la subred Matriz.

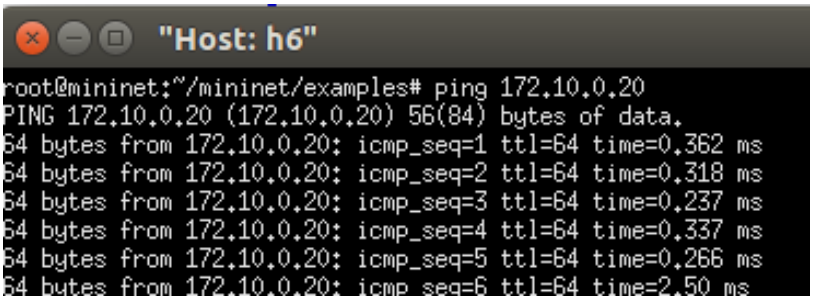
h1 a h2



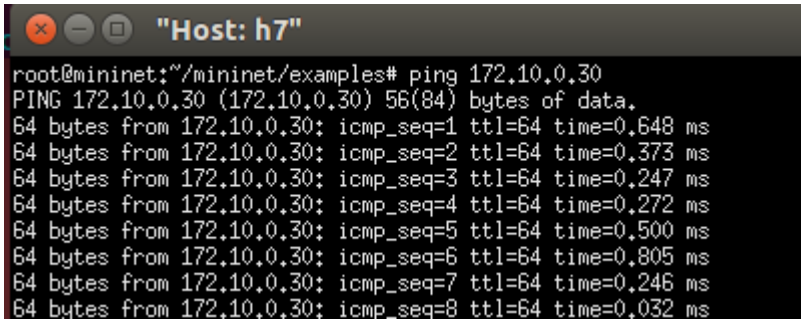
h1 a h3



h6 a h2

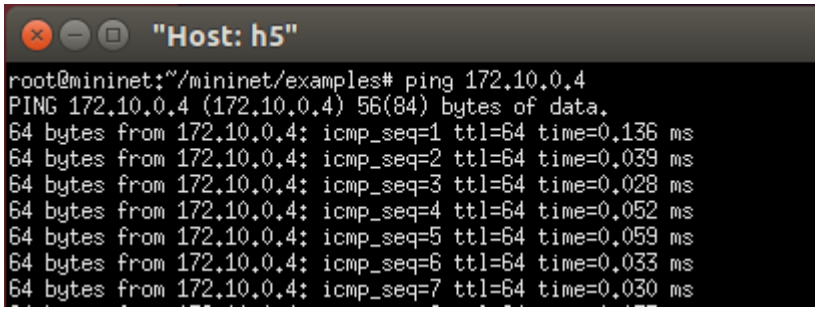


h7 a h3



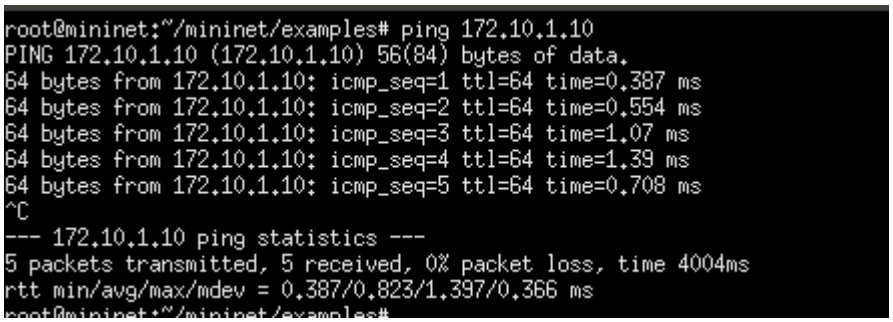
Paquetes icmp exitosos entre hosts de la subred de Sucursales.

h5 a h4

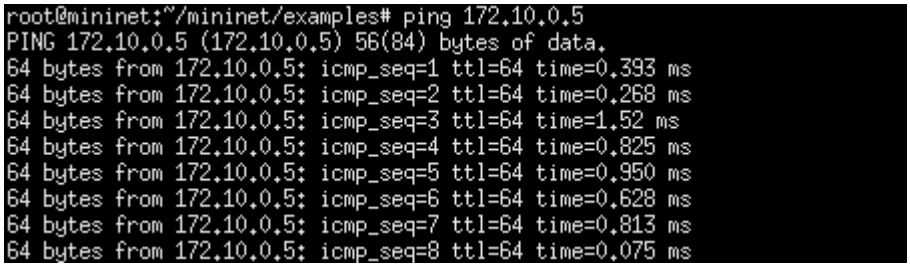


Paquetes icmp exitosos entre routers de la conexión WAN.

Matriz a sucursales



Sucursales a matriz



F.5. Pruebas de desempeño SDN

A fin de contrastar los resultados de la red definida por software versus la red tradicional, se llevaron a cabo las pruebas de desempeño de la misma manera que se lo hizo con la red tradicional, es decir, mediante paquetes icmp con tramas que van desde los 64bytes, 512 bytes, 1518 bytes hasta 4096 bytes como se indica en el RFC 2544.

ping 172.10.1.10 –l 64 –c 10

Latencia entre el host h1 a h2 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.037 ms	0.64 ms	0.033 ms
512 bytes	0.028 ms	0.053 ms	0.35 ms
1518 bytes	0.077 ms	0.146 ms	0.030 ms
Latencia entre el host h1 a h3 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.022 ms	0.64 ms	0.33 ms
512 bytes	0.032 ms	0.024 ms	0.028 ms
1518 bytes	0.081 ms	0.192 ms	0.134 ms
Latencia entre el host h6 a h2 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.028 ms	0.58 ms	0.304 ms
512 bytes	0.039 ms	0.045 ms	0.042 ms
1518 bytes	0.069 ms	0.177 ms	0.123 ms
Latencia entre el host h7 a h3 de la subred Matriz			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.016 ms	0.6 ms	0.616 ms
512 bytes	0.028 ms	0.83 ms	0.429 ms
1518 bytes	0.031 ms	0.2 ms	0.116 ms

Latencia entre el hosts de la subred sucursales			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.015 ms	0.56 ms	0.288 ms
512 bytes	0.009 ms	0.73 ms	0.37 ms
1518 bytes	0.423 ms	0.189 ms	0.306 ms
Latencia entre hosts remotos			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.5 ms	0.9 ms	0.7 ms
512 bytes	4.1 ms	5.5 ms	4.8 ms
1518 bytes	5.3 ms	6.2 ms	5.75 ms

Anexo G: Escenario Mixto

G.1. Pruebas de conectividad red tradicional

Paquetes icmp exitosos entre switches de la SD-WAN

S2 a S1

```
SW2_L3#ping 192.168.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/16/40 ms
SW2_L3#
```

S1 a S3

```
SW1_L3#ping 192.168.40.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/22/32 ms
SW1_L3#
```

S1 a S4

```
SW1_L3#ping 192.168.40.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/25/32 ms
SW1_L3#
```

S2 a S3

```
SW2_L2#ping 192.168.30.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/16/36 ms
SW2_L2#
```

S2 a S4

```
SW2_L3#ping 192.168.40.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/29/52 ms
SW2_L3#
```

S3 a S4

```
SW3_L3#ping 192.168.40.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/24/64 ms
SW3_L3#
```

Paquetes icmp exitosos entre hosts de la misma sucursal

Server 1 a Server 2

SERVER1

```
VPCS> ping 192.168.10.201
84 bytes from 192.168.10.201 icmp_seq=1 ttl=64 time=0.898 ms
84 bytes from 192.168.10.201 icmp_seq=2 ttl=64 time=1.892 ms
84 bytes from 192.168.10.201 icmp_seq=3 ttl=64 time=1.906 ms
84 bytes from 192.168.10.201 icmp_seq=4 ttl=64 time=0.000 ms
84 bytes from 192.168.10.201 icmp_seq=5 ttl=64 time=0.000 ms
```

Paquetes icmp exitosos entre hosts de diferentes sucursales

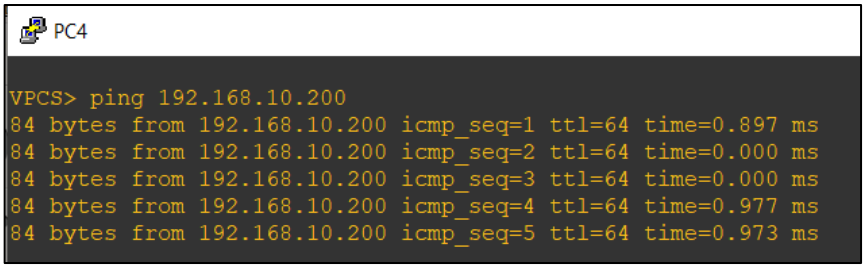
PC-2 a PC-3

PC2

```
VPCS> ping 192.168.30.31
84 bytes from 192.168.30.31 icmp_seq=1 ttl=64 time=0.859 ms
84 bytes from 192.168.30.31 icmp_seq=2 ttl=64 time=0.975 ms
84 bytes from 192.168.30.31 icmp_seq=3 ttl=64 time=0.978 ms
84 bytes from 192.168.30.31 icmp_seq=4 ttl=64 time=0.000 ms
84 bytes from 192.168.30.31 icmp_seq=5 ttl=64 time=0.000 ms
```

Paquetes icmp exitosos entre hosts y datacenter

PC4 a Server 1



G.2. Pruebas de desempeño red tradicional

Latencia en la SD-WAN: SW2_L3 a SW1_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.027 ms	0.71 ms	0.3685 ms
512 bytes	0.031 ms	0.36 ms	0.1955 ms
1518 bytes	0.065 ms	0.192 ms	0.1285 ms
Latencia en la SD-WAN: SW1_L3 a SW3_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.046 ms	0.67 ms	0.358 ms
512 bytes	0.051 ms	0.031 ms	0.041 ms
1518 bytes	0.122 ms	0.203 ms	0.1625 ms
Latencia en la SD-WAN: SW1_L3 a SW4_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.039 ms	0.67 ms	0.3545 ms
512 bytes	0.041 ms	0.054 ms	0.0475 ms
1518 bytes	0.082 ms	0.183 ms	0.1325 ms
Latencia en la SD-WAN: SW2_L3 a SW3_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.019 ms	0.72 ms	0.3695 ms
512 bytes	0.033 ms	0.63 ms	0.3315 ms
1518 bytes	0.041 ms	0.156 ms	0.0985 ms
Latencia en la SD-WAN: SW2_L3 a SW4_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.019 ms	0.61 ms	0.3145 ms
512 bytes	0.023 ms	0.75 ms	0.3865 ms
1518 bytes	0.516 ms	0.192 ms	0.354 ms
Latencia en la SD-WAN: SW3_L3 a SW4_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	5.9 ms	6.7 ms	6.3 ms
512 bytes	6.033 ms	7.8 ms	6.92 ms
1518 bytes	8.1 ms	9.5 ms	8.8 ms
Latencia entre servidores			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	7.1 ms	10.8 ms	8.95 ms

512 bytes	8.6 ms	11.6 ms	10.1 ms
1518 bytes	8.4 ms	18.2 ms	13.3 ms
Latencia entre hosts de diferentes sucursales			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	3.1 ms	9.9 ms	6.5 ms
512 bytes	8.23 ms	12.9 ms	10.565 ms
1518 bytes	8.1 ms	18.5 ms	13.3 ms
Latencia entre hosts y el datacenter			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	9.97 ms	10.85 ms	10.41 ms
512 bytes	9.08 ms	16.43 ms	12.755 ms
1518 bytes	9.33 ms	18.1 ms	13.715 ms

G.3. Script en Python

```
#!/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.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                   build=False,
                   ipBase='192.168.0.0/16')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                         controller=RemoteController,
                         ip='127.0.0.1',
                         protocol='tcp',
                         port=6633)

    info( '*** Add switches\n')
    s8 = net.addSwitch('s8', cls=OVSKernelSwitch, failMode='standalone')
    s6 = net.addSwitch('s6', cls=OVSKernelSwitch, failMode='standalone')
    s4 = net.addSwitch('s4', cls=OVSKernelSwitch)
    s5 = net.addSwitch('s5', cls=OVSKernelSwitch)
    s9 = net.addSwitch('s9', cls=OVSKernelSwitch, failMode='standalone')
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch)
    s7 = net.addSwitch('s7', cls=OVSKernelSwitch, failMode='standalone')
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch)

    info( '*** Add hosts\n')
    h1 = net.addHost('h1', cls=Host, ip='192.168.0.1', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='192.168.0.2', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='192.168.0.4', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='192.168.0.3', defaultRoute=None)
```

```
    info( '*** Add links\n')
    net.addLink(s6, s1)
    net.addLink(s7, s3)
    net.addLink(s5, s9)
    net.addLink(s8, s5)
    net.addLink(s9, s4)
    net.addLink(s6, s3)
    net.addLink(s7, s1)
    net.addLink(s1, s4)
    net.addLink(s4, s5)
    net.addLink(s5, s3)
    net.addLink(s1, s3)
    net.addLink(s6, h1)
    net.addLink(s7, h2)
    net.addLink(s9, h4)
    net.addLink(s8, h3)
    net.addLink(s4, s8)

    info( '*** Starting network\n')
    net.build()
    info( '*** Starting controllers\n')
    for controller in net.controllers:
        controller.start()

    info( '*** Starting switches\n')
    net.get('s8').start([])
    net.get('s6').start([])
    net.get('s4').start([c0])
    net.get('s5').start([c0])
    net.get('s9').start([])
    net.get('s1').start([c0])
    net.get('s7').start([])
    net.get('s3').start([c0])

    info( '*** Post configure switches and hosts\n')

    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
```

Paquetes icmp exitosos entre hosts y datacenter

PC4 a Server 1

```
root@mininet:~/mininet/examples# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=1 ttl=64 time=0.462 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from 192.168.20.1: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 192.168.20.1: icmp_seq=4 ttl=64 time=0.037 ms
64 bytes from 192.168.20.1: icmp_seq=5 ttl=64 time=0.057 ms
64 bytes from 192.168.20.1: icmp_seq=6 ttl=64 time=0.026 ms
64 bytes from 192.168.20.1: icmp_seq=7 ttl=64 time=0.027 ms
64 bytes from 192.168.20.1: icmp_seq=8 ttl=64 time=0.026 ms
^C
```

Figura 58. ***ICMP entre SW_L3 de la SDN***
Nota: Elaboración propia

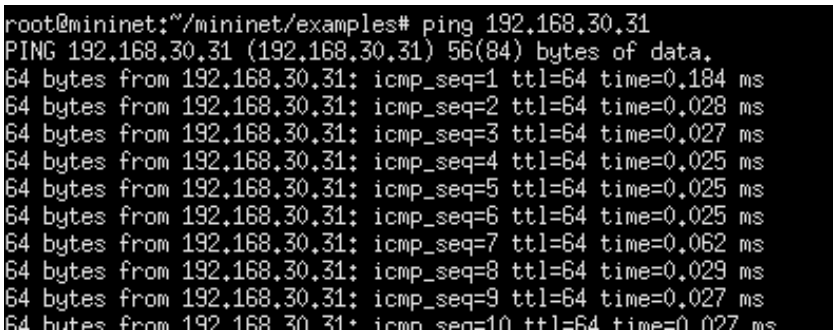


Figura 59. ***ICMP entre hosts de sucursales***
Nota: Elaboración propia

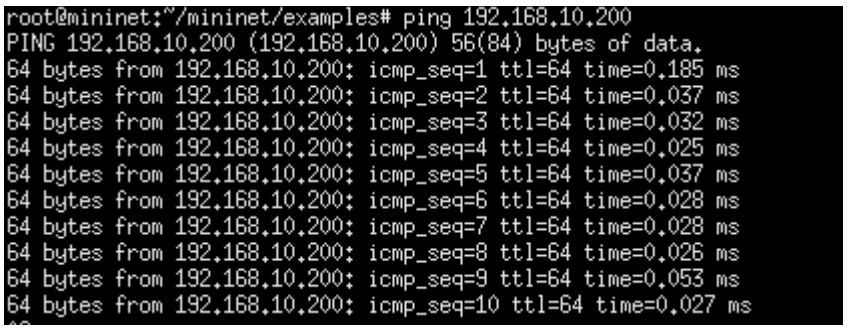


Figura 60. ***ICMP entre datacenters y sucursales***
Nota: Elaboración propia

G.5. Pruebas de desempeño SDN

Latencia en la SD-WAN: SW2_L3 a SW1_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.32 m	0.42 ms	0.37 ms
512 bytes	0.451 ms	0.503 ms	0.477 ms
1518 bytes	0.521 ms	0.598 ms	0.560 ms
Latencia en la SD-WAN: SW1_L3 a SW3_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.09 ms	0.121 ms	0.106 ms
512 bytes	0.07 ms	0.092 ms	0.081 ms
1518 bytes	0.181 ms	0.206 ms	0.1935 ms
Latencia en la SD-WAN: SW1_L3 a SW4_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.023 ms	0.033 ms	0.028 ms
512 bytes	0.042 ms	0.0612 ms	0.0516 ms
1518 bytes	0.052 ms	0.072 ms	0.062 ms
Latencia en la SD-WAN: SW2_L3 a SW3_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.021 ms	0.036 ms	0.0285 ms

512 bytes	0.043 ms	0.052 ms	0.0475 ms
1518 bytes	0.638 ms	0.902 ms	0.77 ms
Latencia en la SD-WAN: SW2_L3 a SW4_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	0.0222 ms	0.53 ms	0.2761 ms
512 bytes	0.0312 ms	0.66 ms	0.3456 ms
1518 bytes	0.615 ms	0.183 ms	0.399 ms
Latencia en la SD-WAN: SW3_L3 a SW4_L3			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	6.1 ms	6.6 ms	6.35 ms
512 bytes	6.055 ms	7.369 ms	6.712 ms
1518 bytes	8.329 ms	8.942 ms	8.636 ms
Latencia entre servidores			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	7.193 ms	8.203 ms	7.698 ms
512 bytes	8.351 ms	8.567 ms	8.459 ms
1518 bytes	8.615 ms	8.936 ms	8.776 ms
Latencia entre hosts de diferentes sucursales			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	3.128 ms	8.5 ms	5.814 ms
512 bytes	8.635 ms	10.65 ms	9.6425 ms
1518 bytes	8.909 ms	15.751 ms	12.33 ms
Latencia entre hosts y el datacenter			
Longitud de la trama	Tiempo mínimo	Tiempo máximo	Tiempo promedio
64 bytes	9.99 ms	10.52 ms	10.255 ms
512 bytes	9.123 ms	15.742 ms	12.433 ms
1518 bytes	9.421 ms	17.369 ms	13.395 ms

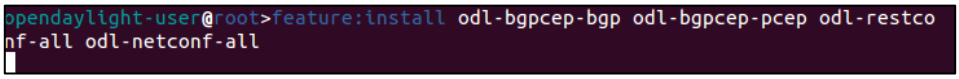
Anexo H: Escenario Híbrido

H.1. Instalación de BGP en el controlador OpenDaylight

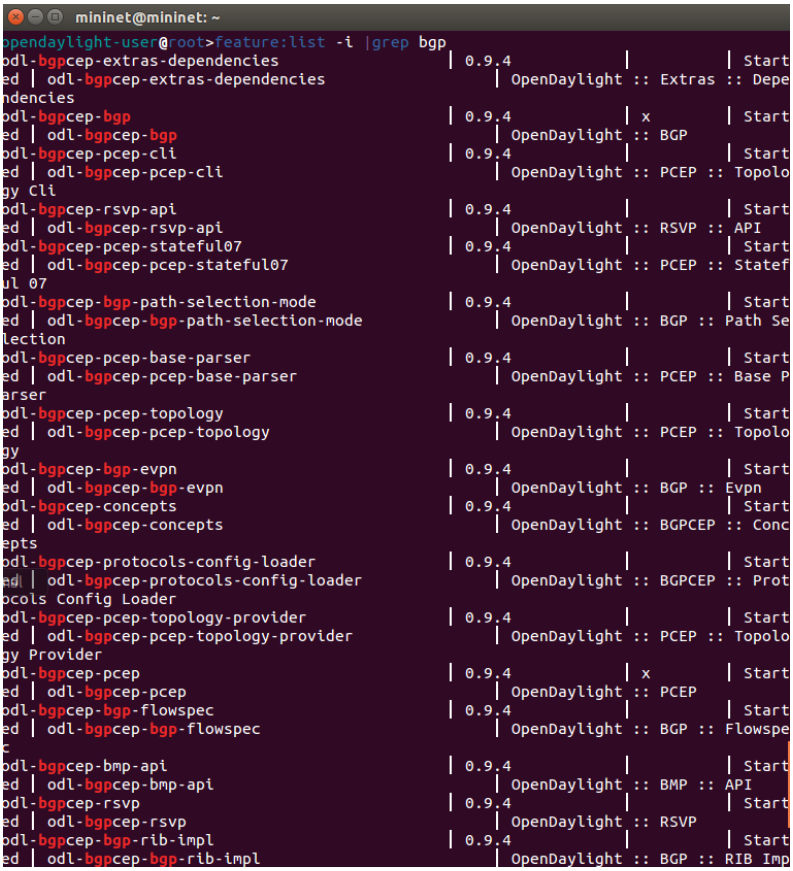
Para configurar BGP-PCEP, el controlador ODL utiliza RESTCONF con el objetivo de editar la configuración predeterminada en los archivos XML.

Lo primero es iniciar la distribución karaf de ODL con el comando *sudo –E karaf* e instalar las dependencias necesarias para incorporar BGP, con el comando:

feature:install odl-bgpcep-bgp odl-bgpcep-pcep odl-restconf-all odl-netconf-all



Para asegurarse de que la instalación esté correcta, se puede verificar los registros con el siguiente comando, que enumerará las funciones de BGP que están instaladas:



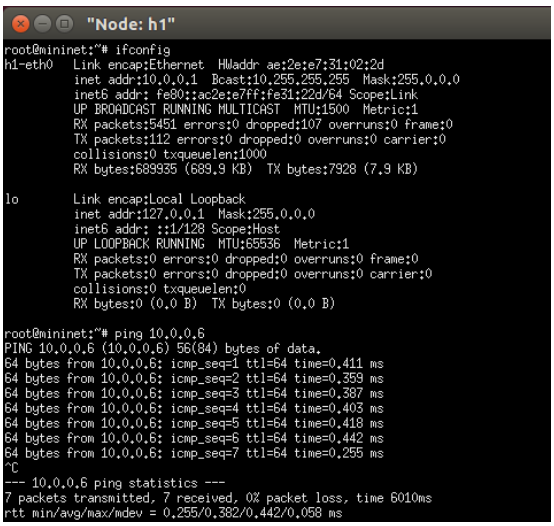
Se puede acceder a los registros de Karaf para verificar si hay errores usando el siguiente comando:

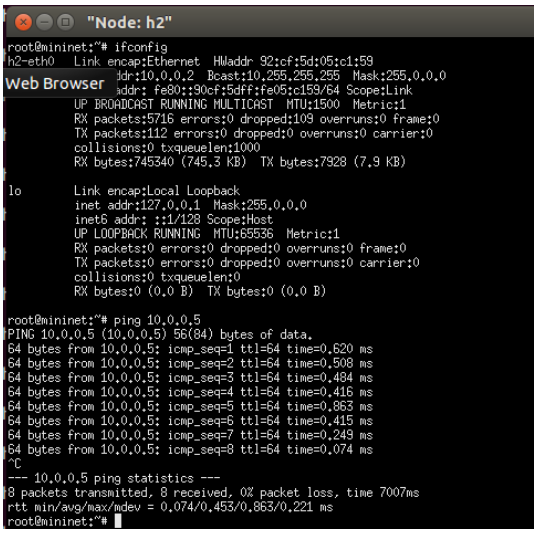
H.2. Script en Python para la creación de vlans

```
#!/usr/bin/python

from mininet.node import Host, RemoteController
from mininet.topo import Topo
import apt

#Note Vlan package check only work with ubuntu
#Please comment the package check if your running the script other
than ubuntu

#package check Start
cache = apt.Cache()
if cache['vlan'].is_installed:
    print "Vlan installed"
else:
    print "ERROR:VLAN package not  installed please run sudo apt-
get install vlan"
    exit(1)
#package check End

class VLANHost( Host ):
        def config( self, vlan=1, **params ):
                """Configure VLANHost according to (optional)
parameters:
                        vlan: VLAN ID for default interface"""
                r = super( Host, self ).config( **params )
                intf = self.defaultIntf()
# remove IP from default, "physical" interface
                self.cmd( 'ifconfig %s inet 0' % intf )
# create VLAN interface
                self.cmd( 'vconfig add %s %d' % ( intf, vlan ) )
# assign the host's IP to the VLAN interface
                self.cmd( 'ifconfig %s.%d inet %s' % ( intf, vlan,
params['ip'] ) )
# update the intf name and host's intf map
                newName = '%s.%d' % ( intf, vlan )
# update the (Mininet) interface to refer to VLAN interface name
                intf.name = newName
# add VLAN interface to host's name to intf map
                self.nameToIntf[ newName ] = intf
                return r
```

```
class MyTopo( Topo ):
    "Configuracion de 802.1Q"

    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        host1=self.addHost( 'h1', cls=VLANHost, vlan=100)
        host2=self.addHost( 'h2', cls=VLANHost, vlan=200)
        host3=self.addHost( 'h3', cls=VLANHost, vlan=100)
        host4=self.addHost( 'h4', cls=VLANHost, vlan=200)
        host5=self.addHost( 'h5', cls=VLANHost, vlan=100)
        host6=self.addHost( 'h6', cls=VLANHost, vlan=200)

        s1 = self.addSwitch( 's1' )
        s2 = self.addSwitch( 's2' )
        s3 = self.addSwitch( 's3' )

        self.addLink(s1,host1)
        self.addLink(s1,host2)
        self.addLink(s1,s2)
        self.addLink(s2,host3)
        self.addLink(s2,host4)
        self.addLink(s2,s3)
        self.addLink(s3,host5)
        self.addLink(s3,host6)
topos = { 'simplevlan': ( lambda: MyTopo() ) }
```

H.3. Pruebas de configuración





"Node: h3"

```
root@mininet:~# ifconfig
h3-eth0   Link encap:Ethernet  HWaddr 6e:6a:68:de:42:09
          inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::6c6a:68ff:fede:4209/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5822 errors:0 dropped:111 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:766162 (766.1 KB)  TX bytes:7928 (7.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mininet:~# ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.406 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.411 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.436 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.378 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.453 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.347 ms
^C
--- 10.0.0.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 0.347/0.405/0.453/0.036 ms
root@mininet:~#
```

"Node: h4"

```
root@mininet:~# ifconfig
h4-eth0   Link encap:Ethernet  HWaddr 9e:9c:b7:76:72:d9
          inet addr:10.0.0.4  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::9c9c:b7ff:fe76:72d9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5801 errors:0 dropped:112 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:767516 (767.5 KB)  TX bytes:7928 (7.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mininet:~# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.368 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.374 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.401 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.390 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.389 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.351 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.189 ms
^C
--- 10.0.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 0.189/0.351/0.401/0.071 ms
root@mininet:~#
```

"Node: h5"

```
root@mininet:~# ifconfig
h5-eth0   Link encap:Ethernet  HWaddr 2a:df:50:6c:e4:f9
          inet addr:10.0.0.5  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::28df:50ff:fe6c:e4f9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5823 errors:0 dropped:115 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:772239 (772.2 KB)  TX bytes:7928 (7.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mininet:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.207 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.028 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.029 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.031 ms
^C
--- 10.0.0.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6000ms
rtt min/avg/max/mdev = 0.028/0.056/0.207/0.061 ms
root@mininet:~#
```

"Node: h6"

```
root@mininet:~# ifconfig
h6-eth0   Link encap:Ethernet  HWaddr 96:cf:63:44:77:ba
          inet addr:10.0.0.6  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::94cf:63ff:fe44:77ba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5841 errors:0 dropped:116 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:775812 (775.8 KB)  TX bytes:7928 (7.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mininet:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.391 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.547 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.444 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.371 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.380 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.515 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.306 ms
^C
--- 10.0.0.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6002ms
rtt min/avg/max/mdev = 0.306/0.422/0.547/0.078 ms
```

"Node: h11"

```
h11-eth0  Link encap:Ethernet  HWaddr a2:36:c9:1c:be:21
          inet addr:10.0.0.11  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::a036:c9ff:fe1c:be21/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7501 errors:0 dropped:179 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1063882 (1.0 MB)  TX bytes:7928 (7.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mininet:~# ping 10.0.0.18
PING 10.0.0.18 (10.0.0.18) 56(84) bytes of data.
64 bytes from 10.0.0.18: icmp_seq=1 ttl=64 time=0.375 ms
64 bytes from 10.0.0.18: icmp_seq=2 ttl=64 time=0.540 ms
64 bytes from 10.0.0.18: icmp_seq=3 ttl=64 time=0.393 ms
64 bytes from 10.0.0.18: icmp_seq=4 ttl=64 time=0.430 ms
64 bytes from 10.0.0.18: icmp_seq=5 ttl=64 time=0.416 ms
64 bytes from 10.0.0.18: icmp_seq=6 ttl=64 time=1.07 ms
64 bytes from 10.0.0.18: icmp_seq=7 ttl=64 time=0.203 ms
^C
--- 10.0.0.18 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 0.203/0.490/1.073/0.255 ms
```

"Node: h12"

```
root@mininet:~# ifconfig
h12-eth0  Link encap:Ethernet  HWaddr 66:36:98:e9:e4:77
          inet addr:10.0.0.12  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::6436:98ff:fee9:e477/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7527 errors:0 dropped:188 overruns:0 frame:0
          TX packets:115 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1065722 (1.0 MB)  TX bytes:8222 (8.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mininet:~# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.288 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.067 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.040 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.038 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.037 ms
^C
--- 10.0.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5998ms
rtt min/avg/max/mdev = 0.037/0.078/0.288/0.086 ms
root@mininet:~#
```

Anexo I: Matriz de compatibilidad de Cisco para OpenFlow

Releases	Supported Platforms	Feature Information
Cisco Plug-in for OpenFlow Release 1.3	The supported platforms Nexus 3000 Series Devices Nexus 3100 Series Devices Nexus 9300 Series Devices	For Cisco Nexus 3000 and Cisco Nexus 3100 Series devices, the Cisco Plug-in for OpenFlow Release 1.3 needs to be used for NX-OS release 7.0(3) and later.
Cisco Plug-in for OpenFlow Release 1.1.5	The supported platforms are Nexus 3000 Series Devices. The Nexus 3548-X device is supported in NX-OS software release 6.0(2)A6(2) and higher.	Cisco Plug-in for OpenFlow supports OFA decommissioning.
Cisco Plug-in for OpenFlow Release 1.1.1	The supported platforms are: Nexus 3000 Series Devices Nexus 5000 Series Devices Nexus 6000 Series Devices	Cisco Plug-in for OpenFlow now supports Nexus 5000 and 6000 Series.
Cisco Plug-in for OpenFlow Release 1.1	The supported platforms are Nexus 3000 Series Devices.	The OpenFlow hybrid (ships-in-night) model is supported. L3 ACL and L2 MAC forwarding tables are supported and can be configured using pipelines. Transport Layer Security (TLS) is supported in Cisco Plug-in for OpenFlow and controller communications. VLAN priority has been introduced as a flow action. The following commands have been introduced: clear openflow, max-backoff, probe-interval, rate-limit, tls trust-point. The controller command has been modified to include the no-tls keyword.
Cisco Plug-in for OpenFlow Release 1.0.1	The supported platforms are Nexus 3000 Series Devices.	The following flow actions are supported: Modify source MAC address Modify destination MAC address
Cisco Plug-in for OpenFlow Release 1.0	The supported platforms are Nexus 3000 Series Devices.	Cisco Plug-in for OpenFlow supports OpenFlow 1.0, and helps networks become more open, programmable, and application-aware.

Nota: Tomado de Guía de configuración 1.3 de OpenFlow (CISCO, 2017)