UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE INGENIERÍA ELECTRÓNICA

DISEÑO DE UNA RED PARA EL USO DE APLICACIONES DE VIDEO MEDIANTE REDES DEFINIDAS POR SOFTWARE (SDN)

Trabajo de titulación previo a la obtención del

Título de Ingeniero Electrónico

AUTOR: Danny Alexander Jácome Paredes

Jimmy Paolo Naranjo Pachacama

TUTOR: Juan Carlos Domínguez Ayala

Quito-Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Danny Alexander Jácome Paredes, con documento de identificación N° 1726596495 y Jimmy Paolo Naranjo Pachacama, con documento de identificación N° 1718608522; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 20 de septiembre del año 2022

Atentamente,

…… ……

Danny Alexander Jácome Paredes

1726596495

Jimmy Paolo Naranjo Pachacama

1718608522

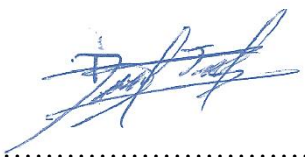
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Danny Alexander Jácome Paredes, con documento de identificación N° 1726596495 y Jimmy Paolo Naranjo Pachacama, con documento de identificación N° 1718608522, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del Proyecto Técnico: "Diseño de una red para el uso de aplicaciones de video mediante redes definidas por software (SDN)¨, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 20 de septiembre del año 2022

Atentamente,



…….

Danny Alexander Jácome Paredes

1726596495

……..…….

Jimmy Paolo Naranjo Pachacama

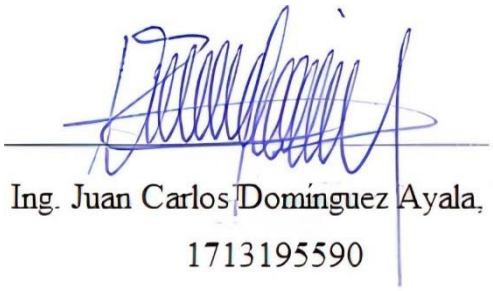
1718608522

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Carlos Domínguez Ayala con documento de identificación N° 1713195590, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación DISEÑO DE UNA RED PARA EL USO DE APLICACIONES DE VIDEO MEDIANTE REDES DEFINIDAS POR SOFTWARE (SDN), realizado por Danny Alexander Jácome Paredes, con documento de identificación N° 1726596495 y Jimmy Paolo Naranjo Pachacama, con documento de identificación N° 1718608522, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 20 de septiembre del año 2022

Atentamente,



Ing. Juan Carlos Domínguez Ayala, MSc
1713195590

DEDICATORIA

Dedico este proyecto de titulación a mis padres Hernán y Miriam quienes me han apoyado, han sido amorosos y creyeron en mi haciendo sacrificios que me han permitido cumplir las metas personales y profesionales que me he planteado. También a mis hermanas Erika y Odalis por su apoyo, ayuda y compañía incondicional durante esta jornada de crecimiento personal. Juntos hemos tenido experiencias agridulces y enriquecedoras, pero siempre se han encontrado en el momento en que más los necesitaba.

Danny Alexander Jácome Paredes

Dedico este proyecto principalmente a mis padres Blanca, Eduardo y a mi hermano Stalin, que siempre ha estado presentes en todo este proceso de crecimiento educativo y profesional, por todo el amor incondicional, la confianza, el sacrificio, por siempre darme palabras de aliento para culminar mi carrera**.**

Jimmy Paolo Naranjo Pachacama

AGRADECIMIENTOS

Agradezco a Dios, a mis padres y a mi hermano por habernos permitido culminar con este objetivo. A la Universidad Politécnica Salesiana por habernos brindado la oportunidad de habernos desarrollado como estudiantes y como personas de bien.

A nuestro tutor, el Ing. Juan Carlos Domínguez que, gracias a su ayuda y su disposición con sus conocimientos, nos ayudó a la realización nuestro trabajo.

Jimmy Paolo Naranjo Pachacama

Quiero agradecer a mis padres, hermanas y amistades que han brindado su ayuda y confianza para lograr esta meta. También a las personas que nos ayudaron a desarrollar nuestro interés por los conocimientos requeridos para la realización de este trabajo de titulación. A la Universidad Politécnica Salesiana, a sus docentes y colaboradores, por proporcionar lo necesario para poder desarrollarnos y afrontar nuestra vida profesional. Agradezco también a nuestro tutor el Ing. Juan Carlos Domínguez por su apoyo, ayuda, su disposición y su confianza colocada en nosotros durante la realización este proyecto.

Danny Alexander Jácome Paredes

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

RESUMEN

El objetivo del actual proyecto es diseñar una red LAN que emplea redes definidas por software (SDN) y hace uso de las capacidades de un equipo FortiGate en un entorno simulado, también se analizará el uso de aplicaciones de video gracias a herramientas de monitorización de tráfico como Wireshark en GNS3.

La ejecución de OpenDaylight junto a la tecnología de Fortinet permitirá una óptima reacción en tiempo real a los picos de tráfico y una adaptación del uso de los enlaces de red eficiente, permitiendo una fácil configuración, administración y flexibilidad. Siendo el objetivo la priorización el tráfico de las aplicaciones de video.

Se utilizaron métricas para realizar un análisis de rendimiento como son la latencia, pérdida de paquetes, jitter, observando el tráfico de la red, throughput, disposición del controlador SDN y el desempeño de aplicaciones de video en la red.

Palabras Clave

OpenDaylight, FortiGate, Jitter, Wireshark, Open vSwitch, GNS3, Iperf, Latencia, Bandwidth.

ABSTRACT

The objective of the current project is to design a LAN network that employs software defined networking (SDN) and makes use of the capabilities of FortiGate equipment in a simulated environment, and will also analyze the use of video applications thanks to traffic monitoring tools such as Wireshark in GNS3.

The implementation of OpenDaylight together with Fortinet's technology will allow an optimal real-time reaction to traffic peaks and an efficient adaptation of the use of network links, allowing easy configuration, management and flexibility. The goal is to prioritize the traffic of video applications.

Metrics were used to perform a performance analysis such as latency, packet loss, jitter, observing network traffic, throughput, SDN controller layout and video application performance on the network.

Key Words

OpenDaylight, FortiGate, Jitter, Wireshark, Open vSwitch, GNS3, Iperf, Latency, Bandwidth

INTRODUCCIÓN

Las SDN (redes definidas por software) son parte de un conjunto de tecnologías de comunicación que se han desarrollado para la construcción de una arquitectura que divide el control y los datos para la obtención de redes programables, flexibles y automatizables que incrementen sus funcionalidades, independizándose de la infraestructura física. Debido al gran impacto que ha tenido las nuevas tecnologías como son NFV o SDWAN, una de las mejores opciones actualmente es llevar a cabo un estudio de la red SDN y su posibilidad de uso, para este trabajo se desea implementar una red a través del simulador GNS3, siendo este un software el más factible para crear una red virtual ajustada a parámetros reales.

De tal modo, se construyó una guía de instalación y configuración sobre SDN en GNS3, donde se expone el paso a paso para establecer las conexiones iniciales para simular redes, complementado por un documento donde se enseña el marco teórico actual relacionado a las SDN, la práctica de algunos comandos generales de Open vSwitch, su administración mediante el controlador OpenDaylight, y el protocolo estándar OpenFlow.

Para medidas de seguridad en la red se usa Firewall de Fortinet, la misma que depura el tráfico de esta, así evitando daños a una institución de amenazas internas y externas. El proyecto se organiza en los siguientes capítulos:

En el capítulo 1, se muestra el desarrollo del tema del proyecto, la justificación respectiva, el planteamiento del problema, los objetivo principal y objetivos específicos planteados, los métodos de investigación a utilizar en el desarrollo del proyecto según los propuesto.

En el capítulo 2, se indica el estado del arte en el cual se definirá elementos que se utilizaran en el diseño de la red sea estos el programa de simulación, los controladores y elementos de red.

En el capítulo 3, se procede a diseñar la topología en la cual se indicará la instalación y configuración del software usado en la red.

En el capítulo 4, se desarrolla las pruebas de funcionamiento, así como la aplicación de calidad de servicio, el ancho de banda y demás elementos que interfieren en el análisis de diseños de redes.

En el capítulo 5, se procede a realizar un análisis de resultados.

Al final se redactará las conclusiones y recomendaciones de acuerdo con las diferentes simulaciones realizadas en la consola de GNS3 y la toma de resultados.

CAPÍTULO 1

ANTECEDENTES

1.1 Descripción del Problema

La red tradicional puede llegar a ser muy complicada de administrar si se desea configurarla hacia los problemas que se han generado en la actualidad, esta problemática se soluciona mediante las redes de nueva generación como es SDN.

El incremento del uso de las redes por el ingreso de nuevos usuarios debido a las modalidades de trabajo y estudio a distancia ha producido problemas como los escases de ancho de banda, problemas de seguridad, problemas de infraestructura entre otros.

En la nueva modalidad de estudio y trabajo se utiliza en gran mayoría servicios de aplicaciones de video, los mismos que se han vuelto indispensables en las comunicaciones; estos servicios exigen a la red mayores valores de ancho de banda, lo cual produce cuellos de botella en el canal utilizado y genera problemas que necesitan ser solucionados en este nuevo ambiente de la red.

En este nuevo escenario producto de la pandemia del coronavirus, las organizaciones han optado por migrar hacia estas redes de nueva generación para solventar estos problemas, lo cual ha planteado la pregunta de que si estas nuevas redes pueden soportar este escenario.

1.2 Alcance e Importancia

El proyecto consiste de la implementación en un simulador de una red que maneje la arquitectura SDN (redes definidas por software), para en la misma poder visualizar el efecto que producen las aplicaciones de video, para esto se plantea diferentes entornos haciendo uso del controlador OpenDaylight y de un simulador de red denominado GNS3 en el cual se utilizará línea de comandos CLI para la configuración de los dispositivos en la topología.

1.3 Delimitación

Este proyecto técnico de titulación está dirigido a estudiantes, docentes de la UPS y personas que deseen implementar aplicaciones de video streaming mediante redes definidas por software.

1.4 Justificación

Este proyecto de investigación proporciona a los estudiantes de la Universidad Politécnica Salesiana una perspectiva sobre el alcance que poseen las redes definidas por software (SDN), con el objetivo de generar más investigaciones a partir de este trabajo.

El proyecto técnico ayudará a conocer el efecto que los servicios de video streaming producen en redes de nueva generación; los servicios que utilizan aplicaciones de este tipo han sido de mucha ayuda para el desarrollo de actividades como son teletrabajo, telestudio, telemedicina, etc.

La investigación se basa en aspectos académicos, científicos y de estrategias tecnológicas, direccionado a la creación de un entorno virtual para aplicaciones de video, proporcionando la recopilación bibliográfica requerida en el proceso de simulación.

Las redes SDN son más flexibles respecto a las redes convencionales en temas como son la seguridad, la detección de intrusiones, la supervisión de la red y el equilibrio de carga; debido a que se puede utilizar para optimizar los recursos que son necesarios en una red para aplicaciones de video.

1.5 Objetivos

1.5.1 Objetivo General

Diseñar una red definida por software (SDN), para el uso de aplicaciones de video.

1.5.2 Objetivos Específicos

- Analizar el estado del arte de las redes fundamentadas por software SDN, para establecimiento de requerimientos en aplicaciones de video.
- Diseñar una red SDN para que cumpla con los requerimientos de aplicaciones de video y comprobar los efectos que producen parámetros como son latencia, Jitter, perdida de paquetes.
- Simular la red SDN diseñada para la comprobación del funcionamiento de las aplicaciones de video bajo los parámetros de latencia, Jitter, perdida de paquetes.
- Analizar los resultados obtenidos en las aplicaciones de video empleados en la red SDN para la verificación del efecto producido por las pérdidas simuladas durante las pruebas de desempeño.

1.6 Metodología

Para el avance del trabajo de investigación se cumplirá un proceso sistemático, ordenado para validar los parámetros óptimos de trabajo:

Método Descriptivo

Se utilizará el método descriptivo en el análisis de las variables que se ven afectadas en la red SDN por el uso las aplicaciones de video, estas se registraran mediante software que permita recolectar los datos.

Método Experimental

El método experimental se utilizará mediante las pruebas a realizar de la generación de tráfico de la red SDN propuesta, los datos obtenidos se analizarán mediante métricas de red.

CAPÍTULO 2

ESTADO DEL ARTE

2.1 Redes Definidas por Software SDN

Las redes SDN pueden remontarse a la separación del plano de información y de control que se utilizaba anteriormente en la red telefónica pública conmutada como método para mejorar el aprovisionamiento y la administración mucho antes de que esta arquitectura se empezara a aplicar en las redes. El IETF (The Internet Engineering Task Force) comenzó a considerar diferentes formas de desacoplar las capacidades de control y de envío en una propuesta de norma de interfaz distribuida en 2004 denominada apropiadamente ForCES (Forwarding and Control Element Separation) (A. Crouch, 2010).

El primer Controlador que utilizaba OpenFlow fue NOX por el año 2008 cuando tuvo su primer lanzamiento, numerosos otros Controladores han sido lanzados al mercado desde NOX (Erickson, 2013). Pudiendo crear y controlar redes virtuales o redes de hardware mediante un control de software. Una red SDN se enfoca en separar el plano de control de su plano de datos, para ello utiliza una arquitectura que hace a la red más fácil y flexible de gestionar centralizando la administración y separando el plano de control de la función de transmisión de datos en dispositivos de red; la atención se centra en que la red mediante controladores basados en software o interfaces de programación de aplicaciones (API) puedan enrutar el tráfico de red y comunicarse con la infraestructura de hardware (Rapp, 2022).

Los proveedores de software de máquinas virtuales como VMware o empresas como Cisco ponen recursos en SDN en una estrategia para crear centros de datos definidos por software. Una parte de la industria de redes se unió en empresas de código abierto, como son, OpenFlow o el proyecto OpenDaylight en un esfuerzo para aislar perpetuamente la programación de las dependencias del hardware de red (Vizard, 2014).

Tecnologías emergentes como la virtualización de red o las tecnologías de contenedores hacen posible desplegar rápidamente aplicaciones mucho más complejas haciendo posible una mayor escalabilidad y haciendo necesario un mayor control del enrutamiento.

2.2 Funcionamiento de redes SDN.

Para el funcionamiento de las redes definidas por software o SDN, se proceden a separar el plano control de la red y la lógica de los controles de recursos realizado por el plano de datos que se efectúan en un dispositivo de red tradicional, por ello los ejemplos de modelos que utilizan SDN cuentan con alguna versión un controlador existente en el mercado y siguen una arquitectura de SDN que posee una API en dirección al sur (Southbound) y una API en dirección al norte (Northbound).

Los controladores son los denominados cerebros de la red, debido a que proporcionan una visión centralizada de la red, y permiten a los encargados de la gestión que a través de los conmutadores sea posible un control adecuado del tráfico de red. Una API hacia el sur se usa para la transmisión de la información a los conmutadores del plano de datos ubicados "debajo", siendo OpenFlow una API hacia el sur considerada un estándar en primer lugar y uno de los protocolos más comunes utilizados en SDN. Mientras que las API hacia el norte se utilizan para la comunicación entre las aplicaciones y la lógica del plano de aplicación "arriba", siendo de gran ayuda de los administradores de red para la configuración del tráfico y la implementación de servicios de manera programática (Craven, 2020).

2.3 Arquitectura SDN.

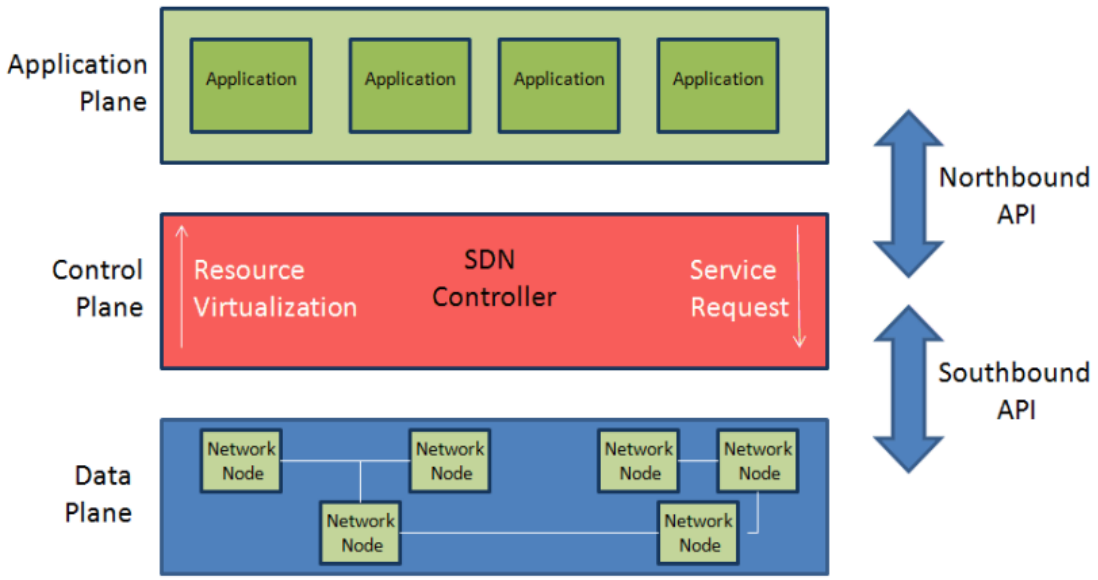
La arquitectura SDN se conforma de tres planos o tres capas diferentes que mediante un conjunto de distintas interfaces hacen posible su comunicación:

El plano de control, se encuentra en la capa media, es el encargado de tomar decisiones sobre el cómo se suceden los eventos en la red a uno o más dispositivo, y de disponer de tales dispositivos para la ejecución, el controlador proporciona una administración de las funciones en este plano para funciones de enrutamiento y conmutación (Blanco Pérez, 2019).

El plano de datos contiene en su capa a los aparatos de forwarding físicos y virtuales, los mismos que son encargados de soportar funciones como el encapsulado de las tramas, el juntar las tramas de red con la tabla de envíos, el filtrado de datos por medio de la ACL (lista de control de acceso), entre otros; (Jiménez, 2020) para lograr la comunicación de este plano con el anterior se utiliza una interfaz Southbound.

El plano de aplicaciones es donde se hallan las aplicaciones que especifican el proceder de la red, estas juntas forman el plano comunicándose con el controlador SDN para lograr la función de red (Blanco Pérez, 2019).

Figura 2.1 Arquitectura SDN



Arquitectura SDN, Northbound API y Southbound API (Alexander La rosa, 2021)

2.3.1 Interface hacia el sur (Southbound API).

Esta es utilizada por el controlador SDN para interactuar en ámbitos mediante protocolos como OpenFlow para un mayor nivel de comunicación junto a los conmutadores y nodos de red. Esto permite que el enrutador identifique la topología de la red y determine los flujos de manera dinámica según las demandas que se implementen en ese instante (Pinilla, 2015). Algunas de las API Southbound más usadas son OpenFlow, Netconf, OpFlex; algunos proveedores que admiten OpenFlow son IBM, Dell, Juniper, Arista y más.

2.3.2 Interface hacia el norte (Northbound API).

Las API Northbound se utilizan en la interacción del controlador de la red SDN y las aplicaciones para poder aplicar instrucciones que permitan una adecuada y automática gestión de la red; ya que también se utilizan estas para que el controlador anuncie los posibles cambios en la arquitectura esta permite de esta manera solicitar instrucciones nuevas a ejecutar (Pozuelo, 2016).

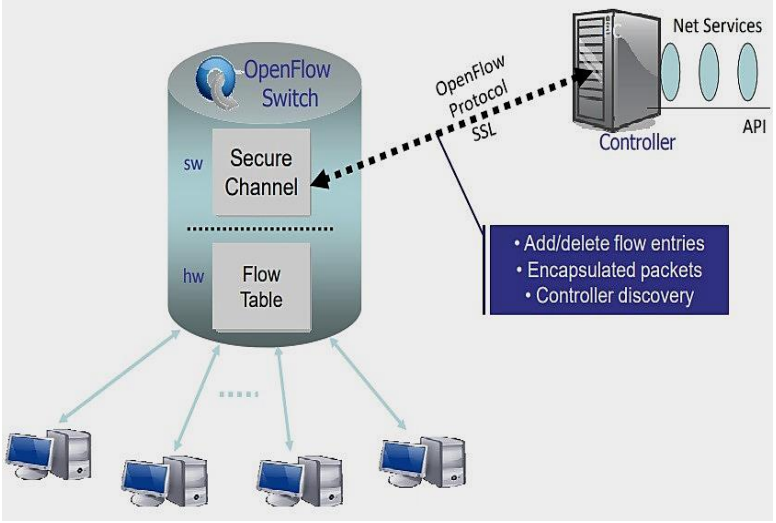
2.4 OpenFlow

El protocolo OpenFlow (OF) es comúnmente asociado a las redes SDN, siendo un protocolo de comunicación entre las capas dentro de esta arquitectura, permitiendo la manipulación del plano de control de los dispositivos OpenFlow utilizando un canal seguro que conecta cada switch con soporte de OpenFlow a un controlador. La conexión de la capa segura de transporte (TLS) con el controlador se inicia al instante de encenderse, siendo el puerto TCP

por defecto del controlador es el 6633. Un conmutador debe ser programable por el usuario con un certificado utilizado para la autenticación al controlador (certificado del controlador) y el otro para autorizar al switch OpenFlow (certificado del switch) (Azodolmolky, 2013).

La organización Open Networking Foundation (ONF) estandariza el protocolo que relaciona a las tecnologías Openflow, utiliza un modelo de código abierto con el objetivo de promover SDN con el objetivo de reagrupar componentes disgregados en sistemas integrados y completos acelerando la adopción de SDN y NFV al permitir el modificar la red en una plataforma innovadora para los diferentes servicios (Open Networking Foundation, 2022).

Figura 2.2 Arquitectura del Protocolo OpenFlow



Arquitectura detalla los protocolos en un switch OpenFlow (Zhu, 2018)

2.5 Switch OpenFlow.

Un Switch OpenFlow es un elemento de reenvío básico, al que se puede acceder a través del protocolo e interfaz OpenFlow; las arquitecturas SDN basadas en flujos como OpenFlow pueden requerir entradas adicionales en la tabla de reenvío, espacio en el búfer y contadores estadísticos que no son muy fáciles de implementar en los conmutadores tradicionales. En una red OpenFlow los conmutadores son de dos tipos: híbridos (con OpenFlow (habilitados para OpenFlow) y puros (sólo OpenFlow).

Los conmutadores híbridos trabajan con OpenFlow además del funcionamiento de los protocolos tradicionales de conmutación de capa 2 y capa 3, los conmutadores OpenFlow puros dependen completamente de un controlador para las decisiones de reenvío y no tienen funciones de control integrado. La mayoría de los conmutadores comerciales disponibles actualmente son híbridos; los conmutadores OpenFlow se controlan mediante una interfaz

abierta mediante sesión TLS basada en TCP, es importante que este enlace esté disponible y seguro. (Azodolmolky, 2013, p. 34)

La ruta de flujo en un conmutador OpenFlow se define mediante una tabla que contiene un conjunto de campos de encabezado de paquetes y una acción.

Cada entrada de flujo está asociada con una acción que le dice al conmutador OpenFlow cómo manejar el paquete. Si no hay ninguna acción de reenvío el paquete se descarta, la lista de acciones se procesa en el orden especificado; Sin embargo, no se garantiza el orden en que se envían los paquetes dentro de un solo puerto.

Los switches OpenFlow puros sólo admiten las acciones requeridas, mientras que los conmutadores OpenFlow híbridos también pueden admitir la acción NORMAL. Cualquiera de los dos tipos de conmutadores admite la acción FLOOD. Las acciones son:

FORWARD: Deben admitir el reenvío de paquetes a los siguientes puertos físicos y virtuales:

- ALL: Envía el paquete a todas las interfaces, excluyendo el puerto de entrada.
- CONTROLLER: Encapsula y envía el paquete al controlador.
- LOCAL: Envía el paquete a la pila de red local del switch.
- TABLE: Realiza la acción en la tabla de flujo.
- IN_PORT: Un puerto de entrada envía el paquete.

DROP: Indica que todos los paquetes coincidentes deben ser descartados.

NORMAL: Procesa el paquete utilizando la ruta de reenvío tradicional soportado por el switch (es decir, procesamiento tradicional L2, VLAN y/o L3).

FLOOD: Inunda el paquete a lo largo del árbol, sin incluir la interfaz entrante.

ENQUEUE: Reenvía el paquete a través de la cola adjuntada al puerto. El reenvío de paquetes está determinado por la configuración de la cola y es para soporte de QoS básico.

Modificar Campo: Las acciones de modificación de campos son:

- Set VLAN ID: Si no existe una VLAN se añade una nueva cabecera con la ID de VLAN añadida (datos asociados de 12 bits) y una prioridad de cero.

- VLAN Priority: Si no existe una VLAN se añade una nueva cabecera con la prioridad (datos asociados de 3 bits) y un ID de VLAN de cero Si ya existe una cabecera el campo de prioridad se sustituye por el valor especificado.

- Modifying the Ethernet Source/Destination MAC address: Reemplaza la dirección MAC de origen/destino de Ethernet con el nuevo valor (especificado como un dato de 48 bits).

- Modifying the IPv4 Source/Destination address: Reemplaza la dirección IP origen/destino existente por un nuevo valor (asociado con un dato de 32 bits) y actualiza la suma de comprobación IP (y la suma de comprobación TCP/UDP, si es requerido).

- Modifying the IPv4 ToS bits: Esto reemplaza el campo IP ToS existente con los 6 bits de datos asociados. Esta acción sólo es aplicable a los paquetes IPv4.

- Modifying the transport Source/Destination port: Reemplaza el puerto de origen/destino TCP/UDP con los datos de 16 bits asociados. Esta acción sólo es aplicable a los paquetes TCP y UDP (Azodolmolky, 2013, p. 14).

2.5.1 Tablas y entradas de flujo.

Las Flow Tables (tablas de flujo) contienen entradas las mismas que se componen de:

- Match fields: Empareja los paquetes. Se forman por el puerto de entrada y las cabeceras de los paquetes, y opcionalmente otros campos como los metadatos de una anterior tabla.
- Priority: Empareja la prioridad que coincide en la entrada de flujo.
- Counters: Se actualizan cuando los paquetes coinciden.
- Instructions: Modifican el conjunto de acciones.
- Timeouts: Cantidad máxima de tiempo o tiempo de inactividad (idle time) previo a que un flujo sea expirado en el conmutador OpenFlow.
- Cookie: Es un valor dado a los datos elegido por el controlador. Puede ser utilizado por este para dar filtro de los flujos, por sus estadísticas, la modificación y las solicitudes de eliminación. No se utilizan al procesar paquetes.
- Flags: Las banderas modifican la manera en que se funcionan las entradas de flujo, por ejemplo, remover mensajes de una entrada de flujo (Open Networking Foundation, 2015, p. 22).

Figura 2.3 Flujo OpenFlow

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

Componentes de un flujo en una tabla Openflow (Open Networking Foundation, 2015)

2.6 Procesamiento del Pipeline.

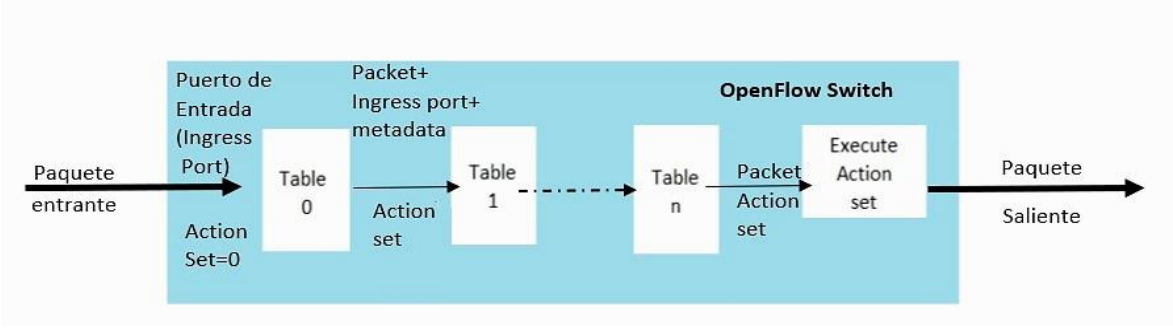
El pipeline de OpenFlow para cada conmutador contiene una tabla de flujo y cada tabla de flujo contiene varias entradas. Estas definen cómo interactúan los paquetes con la tabla de flujo.

La primera tabla de flujo es la que inicia el proceso de pipeline, verificando una coincidencia entre los paquetes y la Flow Table Entry (la entrada de flujo) en la tabla 0. Las tablas de entrada que quedan se utilizan en función de los resultados de la primera tabla.

Cuando la tabla de flujo procesa el paquete, se produce un proceso de coincidencia con las tablas y sus correspondientes Flow Table Entry. Si coinciden, se ejecuta la instrucción correspondiente contenida en el mismo. Si no se encuentra una instrucción el procedimiento del pipeline se finaliza y el paquete se envía en el puerto que corresponde. Otro escenario posible es cuando no hay entradas de tabla de flujo coincidentes con el paquete, en cuyo caso se invocará un error de tabla (table miss).

El comportamiento de una tabla depende de la configuración colocada, siendo el comportamiento común un descarte del paquete, el envío a una diferente tabla o el envío directo a un controlador (Blanco Pérez, 2019, pág. 25).

Figura 2.4 Pipeline Processing Openflow



Funcionamiento del Pipeline Processing de un Switch Openflow (Blanco Pérez, 2019)

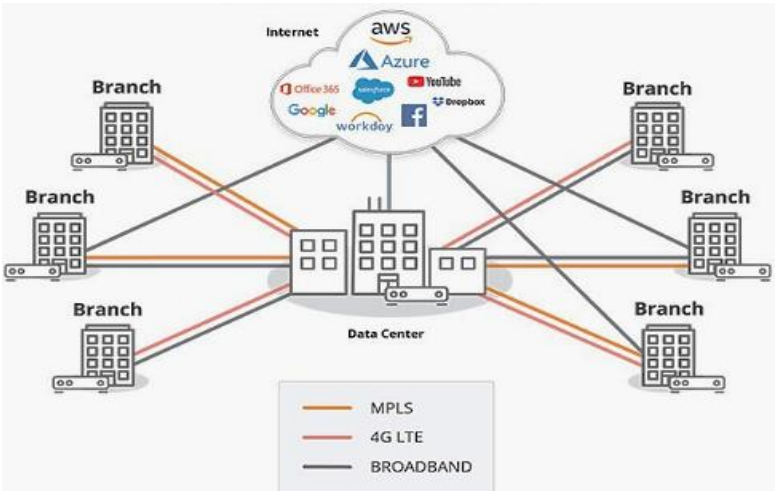
2.7 SD-WAN

SD-WAN es un conjunto de técnicas que hace énfasis en el potencial de cambiar ampliamente el campo de las redes WAN (red de área extendida) y se menciona como una alternativa a los servicios de optimización de WAN, VPN-MPLS, administración de redes. Igualmente, SD-WAN se comprende como un planteamiento único que permite enrutar el tráfico a sitios remotos mediante un mejor transporte. Capacidades avanzadas de monitoreo y gestión del tráfico de red en tiempo real (ARÉVALO, 2020).

2.7.1 Arquitectura SD-WAN.

Las WAN tradicionales, basadas en enrutadores tradicionales, nunca se han diseñado para tecnologías actuales como es la nube. El envío de todo el tráfico, incluido el tráfico destinado a la nube, normalmente se requiere que se enrute desde una sucursal a un centro de datos central al que se pueden aplicar servicios de inspección de seguridad avanzados. Los retrasos causados en el retorno de los datos afectan el rendimiento de la aplicación, lo que resulta en una experiencia de usuario deficiente y una productividad reducida (aruba, 2022).

Figura 2.5 Arquitectura SD-WAN



Descripción gráfica del funcionamiento de SD-WAN de Aruba (aruba, 2022)

2.7.2 Funcionamiento de la SD-WAN.

Con SD-WAN, un departamento de TI puede ofrecer enrutamiento, protección contra amenazas, descarga eficiente de circuitos costosos y simplificación de la gestión de la red WAN. SD-WAN presenta una alta disponibilidad para brindar un servicio predecible a todas las aplicaciones de misión crítica de múltiple conectividad híbrida activa-activa para un

tráfico de red que corresponde a un escenario de enrutamiento dinámico que permita el uso de aplicaciones compatibles para una implementación eficiente y una experiencia de usuario mejorada. Sustituye los costos operativos de servicios de conmutación de etiquetas multiprotocolo (MPLS) con una de banda ancha más barata y manejable (Cisco Systems, Inc, 2022).

La red SD-WAN con las políticas ya configuradas, supervisa de forma inteligente el rendimiento del enlace y comienza a enviar tráfico a lo largo de la ruta óptima en función del SLA especificado anteriormente. En este punto, el circuito no se interrumpe redirigiendo el tráfico a un enlace alternativo.

2.7.3 Tipos y Beneficios de la SD-WAN.

A medida que la tecnología ha evolucionado, las soluciones SD-WAN se han dividido en tres arquitecturas diferentes: sitio a sitio, nube e híbrida. De estas soluciones, la SD-WAN de sitio a sitio es la más básica e incluye solo conexiones de malla entre todas las sucursales de la empresa sin la necesidad de conectarse a una puerta de enlace en la nube.

La SD-WAN que funciona para la nube está dirigida principalmente a empresas que buscan mejorar su rendimiento y la confianza que poseen en sus aplicaciones de la nube. SD-WAN híbrida combina las arquitecturas anteriores pudiendo permitir una infraestructura WAN de malla completa fiable y una sólida conectividad con la nube.

SD-WAN posee varios beneficios respecto a las WAN tradicionales, las principales ventajas de SD-WAN que carece la WAN tradicional son las siguientes:

- Reducir los costos mediante el uso de opciones de conectividad de alta velocidad y bajo costo.
- Rentabilidad y fuerte impacto en las redes comerciales: desaparecen las fronteras geográficas y los modelos de pago se basan en el crecimiento.
- Mejora la visibilidad. Esto significa que puede ver su aplicación de manera inteligente desde el primer paquete de tráfico y tomar decisiones más inteligentes.
- Control de rutas múltiples porque se permiten diferentes conexiones para el tráfico, conexiones de banda ancha, túneles IPsec, etc.
- Reducir la complejidad. Es decir, se gestiona y controla desde un único panel con una interfaz gráfica de fácil uso.

- Introducir servicios basados en la nube. El tráfico tradicional de red de área amplia (WAN) generalmente pasa a través del centro de datos, lo que permite un filtrado continuo, pero aumentando la latencia de los servicios alojados en la nube. SD-WAN se diferencia de la WAN tradicional en la eliminación de tráfico de datos y mejora el rendimiento de las aplicaciones en la nube.
- SD-WAN proporciona QoS para la supervisión del tráfico en ese instante.
- La Capacidad para admitir simultáneamente el uso de aplicaciones que requieran de un alto ancho de banda.
- Seguridad avanzada con funciones que utilizan el cifrado de extremo a extremo y la autenticación de varios dispositivos.
- Se reduce el tiempo de implementación porque la implementación no requiere planificación previa ni apoyo logístico (ZTP). Por lo tanto, mejora la escalabilidad (ARÉVALO, 2020).

2.8 GNS3

GNS3 es probablemente uno de los softwares libres más conocidos y utilizados como una plataforma para el aprendizaje y la educación. A través de los años, estudiantes de bajo nivel de conocimiento, así como ingenieros de redes han utilizado GNS3 como una ayuda indispensable para practicar y al mismo tiempo irse preparando para diversos exámenes de certificación, pudiendo utilizarse para diversos casos, como pruebas de conceptos o exhibiciones comerciales, proporcionando una interfaz de uso fácil, agradable con el usuario y haciendo rentable el uso de software como SDN (Galaxy Technologies LLC., 2021).

Al GNS3 poseer una interfaz amigable para virtualizar dispositivos de hardware reales es utilizada para ejecutar varios programas de simulación de estos aparatos mediante emuladores como Dynamips, VMware y Qemu. Utiliza su software para simular imágenes IOS (sistema operativo de Cisco) que funciona utilizando Dynamips, emulación de Qemu de imágenes de contenedores Docker de múltiples proveedores para automatizar aplicaciones.

GNS3 consta de dos componentes principales para su funcionamiento que son el software cliente (Windows, MAC o Linux) y el hipervisor (máquina virtual). La parte del cliente es un ejecutable que funciona localmente en el sistema operativo huésped y sin el hipervisor.

Si su sistema operativo es Mac o Windows, es recomendando un hipervisor como VMware, VirtualBox o Hyper-V.

Permite emular y configurar varios dispositivos de red (Cisco, Fortinet, Juniper, Microsoft, Aruba), y de varios tipos de componentes de red que funcionan en un servidor GNS3, como firewalls, conmutadores, enrutadores y hosts. Puede proporcionar a los usuarios una diversidad de opciones, como generadores de tráfico, servidores, navegadores, herramientas de seguridad y más. Además, se permite el uso de varios lenguajes de programación, así como salidas a la red externa mediante NAT, nube, etc. (Jiménez, 2020, pág. 9)

2.8.1 Aspectos básicos de GNS3.

GNS3 originalmente solo permitía la emulación de equipos pertenecientes a Cisco haciendo uso de un software llamado Dynamips, cosa que ha cambiado puesto que ahora permite tantos dispositivos emulados de red de Cisco como de otros proveedores e incluso tecnología de contenedores.

Hoy en día GNS3, ha evolucionado para soportar diferentes dispositivos de muchos proveedores de red como FORTINET, JUNIPER entre otros. Su característica de Open Source permite emular varios sistemas operativos virtualmente y está estrechamente vinculado con (Telectrónika, 2018):

Dynamips: diseñado para la emulación de dispositivos CISCO, es proporcionando las ISO de estos dispositivos directamente los enrutadores emulados utilizados en el programa.

Qemu: hace posible la virtualización y la emulación de hardware con su característica de Open Source soportado en GNU/Linux, Windows, y otros admitidos

VirtualBox y VMware: software hipervisor.

Wireshark: muy utilizado para analizar protocolos de red.

2.8.2 Máquina virtual GNS3 (GNS3 VM).

La máquina virtual de GNS3 puede ejecutarse en una maquina huésped como una PC o de forma lejana mediante en un servidor en la nube, por lo que los aparatos a emular creados en la GUI se alojan y ejecutan desde la máquina virtualizada o un servidor local. La selección que es más confiable puede ser VMware Workstation o VirtualBox, siendo su propósito de instalación junto a el software GNS3 la creación de una topología más avanzada o la inclusión de dispositivos de red que requieran más que la plataforma QEMU como es por ejemplo FortiGate, FortiManager, FortiAnalyzer, etc. (ARÉVALO, 2020)

2.8.3 Las Ventajas y desventajas de GNS3.

El software GNS3 ofrece al usuario ventajas importantes con respecto a otros simuladores de red, como es Cisco Packet Tracer o EVE-NG. Sin embargo, así como tiene grandes virtudes cuenta también de características especiales que se detallan en la siguiente Tabla 2.1.

Tabla 2.1 Ventajas y desventajas de GNS3

VENTAJAS	**DESVENTAJAS**
Su instalación puede hacerse en diferentes SO.	Instalación puede resultar un tanto complicada dependiendo del SO.
Es de Código Abierto.	Consumo importante de RAM
Su Interfaz Gráfica de Usuario es fácil de entender.	Imágenes ISO se incluyen en el software.
Manipula ISO de dispositivos reales.	
Análisis en tiempo real de redes emuladas	
Conectar la red emulada a la internet	
Captura de paquetes usando Wireshark.	
Documentación suficiente y soporte directo en la web de GNS3.	

Tabla con las principales ventajas y desventajas de GNS3 *(ARÉVALO, 2020, pág. 24)*

2.9 Funcionamiento de OpenDaylight

El controlador se puede se puede implementar en varios entornos dentro de una red, funcionando mediante la interfaz denominada Northbound que permite la comunicación del controlador ODL y una variedad de aplicaciones este proceso se realiza mediante el uso de la interfaz API.

Por otro lado, la interfaz denominada Southbound se especifica mayormente por el protocolo OpenFlow, el cual hace posible la comunicación entre el controlador ODL y los conmutadores de la red, para la ejecución de cambios dinámicos de las configuraciones dependiendo de las necesidades de los usuarios, por lo que OpenDaylight, reduce el tiempo de la respuestas a los incidentes, los costos de gestión y vuelve más eficiente el despliegue de TI, uniéndose a diferentes plataformas mediante un API (OpenDaylight Project, 2021).

Por ello, las redes SDN pueden crear una infraestructura que sea más flexible mediante la automatización de las funciones, así como la personalización y creación de redes programables, haciendo posible dar soporte a la demanda constante de cambios de los usuarios, los dispositivos y datos que vayan accediendo a la red.

2.10 Open VSwitch

Open vSwitch funciona principalmente como un conmutador virtual en un entorno de máquinas virtuales. Además de mostrar las interfaces de control y la visibilidad de la capa de red virtual, está diseñado para aceptar en varios servidores físicos su distribución. Open vSwitch es compatible con tecnologías que utiliza Linux para la virtualización como son Xen/XenServer, KVM y VirtualBox.

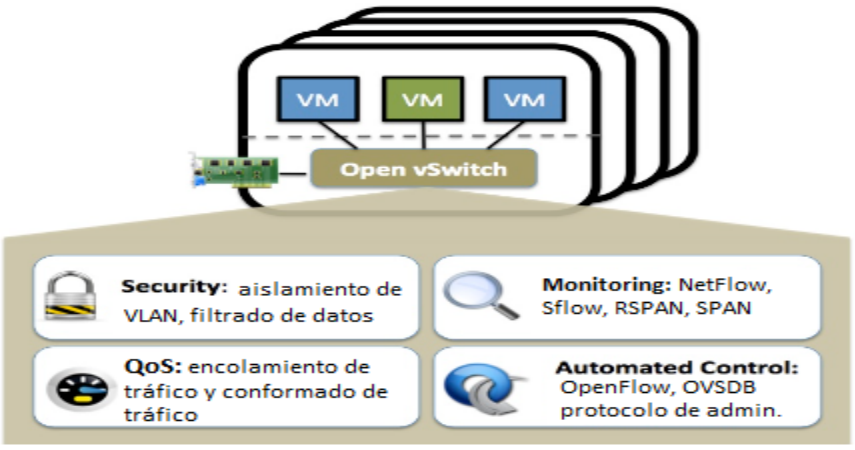
Open vSwitch funciona en una red que contiene controladores, o en su modo stand-alone como un switch Ethernet con un funcionamiento similar de los switches físicos o mediante órdenes transmitidas por consola; además que soporta la unión con OpenFlow de las redes definidas por software.

La versión actual de Open vSwitch posee características como son el modelo VLAN 802.1Q estándar con puertos troncales y de acceso, NIC bonding junto a LACP si es necesario en upstream switch, NetFlow, sFlow(R) mirroring, configuración de QoS, GRE, VXLAN, OpenFlow y numerosas extensiones (Linux Foundation, 2016).

2.10.1 Componentes de Open vSwitch.

Los componentes principales de la distribución actual son:

Figura 2.6 Arquitectura de Open vSwitch



Esquema de la arquitectura de Open vSwitch (Linux Foundation, 2016)

Ovs-vswitchd: Es un Daemon que el conmutador ejecuta junto al módulo kernel de Linux para su uso basado en flujos.

Ovsdb-server: Es un servidor de base de datos ligero del cual Ovs-vswitchd obtiene su configuración.

Ovs-dpctl: Esta herramienta ayuda en la configuración del kernel del conmutador.

Ovs-vsctl: Realiza la consulta y actualización de Ovs-vswitchd.

Ovs-appctl: Envía comandos a los daemons de Open vSwitch en ejecución (Linux Foundation, 2016).

2.11 Fortinet

Es una empresa especializada en el desarrollo y comercialización de software multinacional de origen estadounidense, que crea equipos y servicios de ciberseguridad como firewalls, sistemas antivirus y de prevención de intrusos. Security Fabric FORTINET posee una arquitectura que aborda desafíos de ciberseguridad muy importantes, incluidas los entornos de red de aplicaciones en la nube o los entornos móviles, y proporciona una plataforma enfocada en la ciberseguridad que ofrece: visibilidad extensa en la superficie que está en riesgo de un ataque digital para una administración de riesgos mejorada. Trabajos atomizados que aumentan la cantidad de flujo de las operaciones por su velocidad.

FORTINET tiene en su catálogo una amplia variedad de productos de software y hardware para servicios de conectividad, VoIP16, autenticación, SD-WAN y otras aplicaciones. Las más conocidas son los dispositivos FortiGate, FortiManager y FortiAnalyzer. Estos son temas relacionados que forman parte de la emulación y se explican junto con las características de Secure SD-WAN (Fortinet, Inc., 2022).

2.12 SD-WAN y Fortinet.

Secure SD-WAN de FORTINET incluye funciones de seguridad como NGFW y funciones como SD-WAN; es rápida, escalable y flexible para empresas que necesitan seguridad y priorizan la nube.

La solución segura de red SDWAN de Fortinet permite a las organizaciones transformar y proteger todos los bordes de la WAN. Su enfoque de red es centrado en la seguridad con una consola de administración centralizada y un sistema operativo, las organizaciones logran una

excelente experiencia de usuario, seguridad integrada y eficacia, continuidad y eficiencia de la postura de seguridad de la red (Fortinet, Inc., 2022).

La razón por que Secure SD-WAN es una buena opción para microempresas y empresas se explican en detalle a continuación:

- Fortinet brinda seguridad, enrutamiento avanzado, optimización de WAN y capacidades de redes SD-WAN en un solo equipo. Desde una perspectiva de seguridad de NGFW, se ha mejorado la detección y resolución de ataques.
- Rendimiento mejorado de las aplicaciones en la nube, es decir, las aplicaciones comerciales críticas se priorizan a través de el balanceo de una WAN más eficiente.
- Gestión centralizada, administrada en su totalidad desde un panel fácil de usar, mejora la escalabilidad brindando a las empresas un acceso rápido y fácil a las aplicaciones en la nube.
- Poco TCO: hasta un 50 % de mejora con respecto a la arquitectura tradicional.
- Implementación remota: Abastecimiento automático de un dispositivo (ZTP).

Cuando se trata de implementar una infraestructura WAN, se presenta la figura 2.7 en cual se presenta el Cuadrante Mágico de Gartner para mostrar los mejores o más importantes proveedores en el entorno WAN actual, incluido Fortinet.

Figura 2.7 Cuadrante de Gartner SD-WAN

Cuadrante mágico de Gartner infraestructura SD-WAN (2022 TECNASA INC., 2022)

2.13 FortiGate.

Los dispositivos FortiGate tienen funcionalidad SD-WAN incorporada, esto significa que sus usuarios pueden aprovechar características más avanzadas (como en el caso de SD-WAN) que no posean complejidad ni costos mayores. FortiGate SD-WAN tiene muchas funciones comunes de SD-WAN.

FortiGate SD-WAN permite un acceso a Internet directo para reducir las pérdidas mejorando el rendimiento de lo que usuario necesite en aplicaciones. Fortinet cumple con esto debido a que proporciona información sobre más de 3000 aplicaciones, además de que prioriza el tráfico crítico que la empresa necesita para obtener un alto rendimiento.

Finalmente, cabe mencionar que FortiGate tiene un alto rendimiento de IPSEC VPN utilizado en la industria y garantizando la eficiencia de las aplicaciones, que es un requisito fundamental de SD-WAN (Quanti , 2022).

2.14 Contenedores

Son una forma de virtualización del sistema operativo que está compuesta de capas que se apilan para formar una imagen. Se puede usar un solo contenedor para la ejecución de todo, desde microservicios y procesos de software hasta aplicaciones más grandes. El contenedor posee todos los ejecutables, código binario, bibliotecas y archivos de configuración necesarios. Sin embargo, a diferencia del método de virtualización de máquinas o servidores, un contenedor no contiene una imagen del sistema operativo. Esto reduce significativamente los gastos generales, lo que lo hace liviano y portátil. Para implementaciones de aplicaciones grandes, puede lanzar varios contenedores como uno o más clústeres.

Docker hace uso de estándares abiertos y se ejecuta en los sistemas operativos comunes, incluidos Linux, Microsoft Windows y otros ya sean locales o en la nube (NetApp, 2022).

Los contenedores poseen características que los hacen indispensables como son una menor carga de recursos, una mayor portabilidad, funcionamiento independiente y constante, capacidad de aplicar parches o escalar con mayor rapidez, mejor desarrollo en aplicaciones al acelerar los ciclos de desarrollo, prueba y producción.

2.14.1 Contenedores y Máquinas Virtuales.

La tecnología de contenedores a menudo se confunde con la máquina virtual (VM) o las técnicas de virtualización de servidores. Teniendo algunas similitudes, pero siendo los contenedores muy diferentes de las máquinas virtuales.

La máquina virtual se ejecuta en un entorno de hipervisor y cada una debe incluir su propio sistema operativo junto con bibliotecas, los archivos binarios, y aplicaciones correspondientes. Esto consume muchos recursos y genera muchos gastos generales, especialmente si se ejecuta varias máquinas virtuales en el mismo aparato, cada una con la necesidad de su propio sistema operativo.

De un modo opuesto, cada contenedor utiliza el mismo sistema operativo host o kernel del sistema y siendo más pequeño, a menudo de unos pocos megabytes de tamaño. Por lo general, esto hace que el contenedor tarde pocos segundos en iniciarse en comparación con la cantidad de tiempo necesaria para iniciar una VM (NetApp, 2022).

2.15 Uso de contenedores en GNS3.

Uno de los beneficios de usar una máquina virtual GNS3 como un simulador que admite el uso de imágenes de Docker. Un archivo de Docker es una imagen formada por capas que son utilizadas para ejecutar código desde un contenedor, formándose de instrucciones y ejecutando la aplicación completa basada en el núcleo del sistema operativo del host. Cuando se ejecuta la imagen, se convierte en una o más instancias del contenedor. La imagen de Docker se instala en GNS3 VM para que pueda cargarla más tarde (Jiménez, 2020).

2.16 Requisitos de video

La definición del formato de vídeo, lo que da como resultado la cantidad píxeles por pantalla, dependiendo de la resolución. La tasa de transmisión se refiere a cantidad de cuadros por segundo, si bien las técnicas de muestreo de video varían, cada píxel tiene aproximadamente 3 bytes de color y/o información de luminancia. Cuando toda esta información se factoriza da como resultado el total de información sin codificar (Cisco, 2017).

2.16.1 Códec.

Las palabras codificador-decodificador se abrevian para formar códec, este describe una manera en la que la información se codifica en un archivo y se decodifican cuando se reproduce, siendo el proceso de conversión de un codec a otro la transcodificación (Winchester School of Art, 2012).

Un video que no se ha comprimido es grande respecto a la cantidad de datos que posee, debido a las capacidades de limitación de los sistemas de compresión y almacenamiento, no es posible transmitir video sin comprimir (Juan A. Michell Martín, 2015).

2.16.2 H.264.

El codec H.264 o MPEG-4 AVC (Advanced Video Coding), es un protocolo que se desarrolló con un objetivo de lograr el mayor rendimiento de compresión de datos. El proceso de compresión/descompresión en este, no requiere una gran cantidad de recursos de procesamiento del equipo. Lo que permite que sea adecuado para su envió de información en la internet.

El formato de compresión de video H.264 es ideal para entregar señales de video y audio a una o más fuentes. Su uso es adecuado en específico para la transmisión de señales a largas distancias mediante los cables e infraestructuras existentes (Black Box , 2022).

2.16.3 VP9.

El codec VP9 es un codificador de video que se utiliza para comprimir y descomprimir video desarrollado por Google para proporcionar una alternativa a los códecs con licencia. Su licencia es gratuita y puede ser utilizada por cualquier fabricante de hardware o software. VP9 es una evolución de VP8 y tiene una calidad similar, aunque la compresión es un 50% más eficiente que su predecesora.

Esta compresión es muy útil cuando se almacena contenido en una CDN (red de distribución de contenido), que cobra por cada GB (gigabyte) almacenado y transmitido. Cuanto mayor sea la relación de compresión de video, menor será el costo a pagar (Ortiz, 2017).

2.17 Calidad de Servicio (QoS)

Hace referencia a la capacidad de una red para brindar un mejor servicio para el tráfico de datos en la red, a través de un conjunto de tecnologías que permiten que las aplicaciones soliciten y reciban niveles eficientes de servicio. La convergencia de aplicaciones que ofrecen servicios como voz, datos y video pueden causar una congestión de la red al transportarse por una infraestructura común.

Las aplicaciones hacen uso de protocolos orientados a la conexión, como lo es TCP, donde si se pierde un segmento, se reenvía otro, mientras que para las aplicaciones de voz y video las cuales presentan una mínima tolerancia a la pérdida de paquetes, siendo necesario la implementación de mecanismos que den prioridad al tráfico en caso de presentar una congestión en la red.

Al congestionarse la red la afectación la sufren todas las aplicaciones, pero siendo mayormente afectadas las aplicaciones de voz y video, incluso con la pérdida de la llamada.

Siendo importantes en redes convergentes características como son: ancho de banda disponible, retraso de extremo a extremo, jitter o fluctuación en el retraso y pérdida paquete (Cisco Systems, Inc, 2009).

2.17.1 Ancho de banda disponible.

El término bandwidth (ancho de banda) se refiere a los sistemas de comunicación que pueden transmitir servicios como datos, voz y video a altas velocidades a través de Internet y otras redes. El bandwidth óptimo disponible en una ruta es aquel en que posee un enlace que funciona correctamente haciendo uso de la menor cantidad de este recurso.

Un ancho de banda insuficiente hace que los servicios proporcionados por las aplicaciones se vean degradados por pérdida de paquetes causando un retraso (Ariganello, 2017). Los siguientes recursos pueden resolver estos problemas:

- Aumentando el ancho de banda. Es una forma efectiva de solucionar problemas que pueda causar su deficiencia, pero tiene la particularidad de ser costosa; en algunos casos es la mejor opción.
- Uso de mecanismos de clasificación y marcado de QoS, incluidos los mecanismos de encolamiento apropiados.
- Tecnología de compresión de nivel de capa 2, compresión de cabecera TCP, RTP (compresión de cabecera RTP), etc.

2.17.2 Jitter.

El termino Jitter define a la variación de tiempo que ocurre entre los paquetes que llegan a un sistema. En la terminología común en el uso de aplicaciones en tiempo real estos retrasos se los denominan "lag". Cuanto mayor sea el "retraso" o mayor el "jitter", menor será la calidad de la comunicación. Estas fluctuaciones pueden deberse a la congestión de la red, las variaciones en los tiempos de métricas o el simple cambio de ruta.

Esto es normal incluso en redes de máxima capacidad gestionadas de forma óptima. Por lo tanto, se utilizan los búferes de jitter en videoconferencias para contrarrestar la fluctuación causada por latencias y demoras, caídas de paquetes de datos y colas de actividad de procesos. Estos búferes almacenan los paquetes entrantes para minimizar el jitter y descartan los paquetes que llegan tarde (Dinecom, 2022).

2.17.3 Latencia.

La latencia o tiempo de respuesta, indica el intervalo de tiempo en el que un paquete de datos se demora en viajar desde su origen hasta su destino, se mide en milisegundos (ms). En la

latencia influye factores como la distancia entre el emisor y receptor, el medio por el que se transfiere la información, el tamaño del paquete transmitido, el tipo de procesamiento de datos, el bandwidth (ancho de banda disponible), la cola de espera en el servidor (Ionos, 2022).

2.17.4 Optimización de ancho de banda.

La optimización de la red se realiza por los proveedores de servicio o administradores en redes en las que se han reutilizando la infraestructura, destacando la necesidad que un ISP tiene para ofrecer variados tipos de QoS a sus clientes, siendo el MIR/CIR parámetros llamados Tasa Máxima de Información (MIR, Maximum Information Rate) y Tasa Comprometida de Información (CIR, Commited Information Rate) que ayudan a la reutilización del ancho de banda en una misma infraestructura.

Cuando es asignado un parámetro de QoS se garantiza que haya un ancho de banda referente a la capacidad de ancho de banda a utilizar, un ISP debe asegurar al usuario que si este se encuentra conectado con un ancho de banda compartido el bandwidth será dedicado cuando necesite, esto se realiza de forma independientemente de los usuarios que compartan el mismo ancho de banda. En caso de que un ancho de banda no esté en uso en un momento en específico este puede ser utilizado por otro usuario con su limitación de tráfico respectivo (Medina, 2007).

CAPÍTULO 3

3.1 Diseño de la propuesta de la red

Tomando los datos teóricos explicados en consideración, se procede a la simulación de los casos de uso. Se procederá a simular una topología de red con conmutadores Open vSwitch los mismos que soportan el protocolo OpenFlow, la posición del controlador tendrá un enfoque centralizado sobre la que se probará el tráfico en estudio. Un tráfico de video streaming y tráfico de videoconferencia, mismos en los que se hará fundamental la transmisión y recepción en la topología.

Los flujos de video se pondrán en marcha a través de las máquinas virtuales que actuaran como host donde la fuente será video, y así consecuentemente realizar las pruebas requeridas, para realizar throughput se utilizará el software Iperf, el mismo que es muy útil en pruebas de rendimiento de la red.

Debido a la restricción del hardware y a la necesidad del uso de máquinas virtuales durante el uso del simulador GNS3, así como para mayor comodidad durante la simulación se decidió utilizar una sola maquina como host de toda la simulación.

3.1.1 Componentes funcionales de la red propuesta

En la simulación tenemos los siguientes componentes:

- **Red Virtual:** Las red interna o subredes que se comunican entre sí, estas funcionan de forma que los dispositivos se manejen con las mismas prestaciones que si se conectaran de la forma tradicional.
- **Máquinas Virtuales:** Los archivos informáticos que se ejecutan en el ordenador físico gracias al hipervisor que depende de los recursos físicos de la computadora y a los recursos lógicos.
- **Hipervisor:** Es el software que permite crear y ejecutar máquinas virtuales asignándole recursos, este aísla del sistema operativo huésped las máquinas y gestiona sus recursos.
- **Controlador SDN Opendaylight:** Se instalo en sistemas operativos Linux directamente a la máquina virtual, siendo utilizada una versión que permite su simulación, también puede utilizarse imágenes Docker que se han sido preparadas para simulaciones en GNS3.

- **Open vSwitch con interfaz de administración (OVS):** Es una imagen diseñada para su uso en GNS3 mediante la instalación de una apliance, la misma que poseen las configuraciones necesarias para su uso en GNS3. OpenvSwitch permite la automatización de la red y el uso de protocolos e interfaces de administración estándar.
- **GNS3 VM:** Es la máquina virtual diseñada con el sistema operativo Ubuntu Linux que posee las características de GNS3 necesarias instaladas, esta permite el uso de contenedores que utilizan los servicios proporcionados por el kernel de GNS3 VM. El enfoque dado a contenedores en GNS3 VM es de máquinas virtuales ligeras más que para producción.
- **Firewall FortiGate:** Es una imagen diseñada para su uso en GNS3, los dispositivos FortiGate ofrecen funciones avanzadas de prevención de amenazas ya sea en la implementación de nube pública, híbrida y privada.

3.1.2 Característica de la maquina huésped utilizada

La Tabla 3.1. muestra las características que fueron utilizadas por la máquina para su uso en el entorno de simulación en GNS3, el objetivo de este es el de aprovechar el máximo posible los recursos limitados.

Tabla 3.1 Características de la maquina huésped

ITEM	**REQUERIMIENTOS**
S.O. (Sistema Operativo)	Windows 10 (64 bit)
Procesador	Intel(R) Core (TM) i7-6500U CPU
Virtualización	Es necesario de extensiones para virtualización. Se debe habilitar, mediante del BIOS de la maquina
Memoria	16 GB RAM
Espacio en disco	Disco SSD (estado sólido) de 480 Gb

La máquina para la simulación utiliza un disco SSD y 16 GB de RAM

3.1.3 Diagrama De Implementación

Figura 3.1 Diagrama de Implementación de la simulación

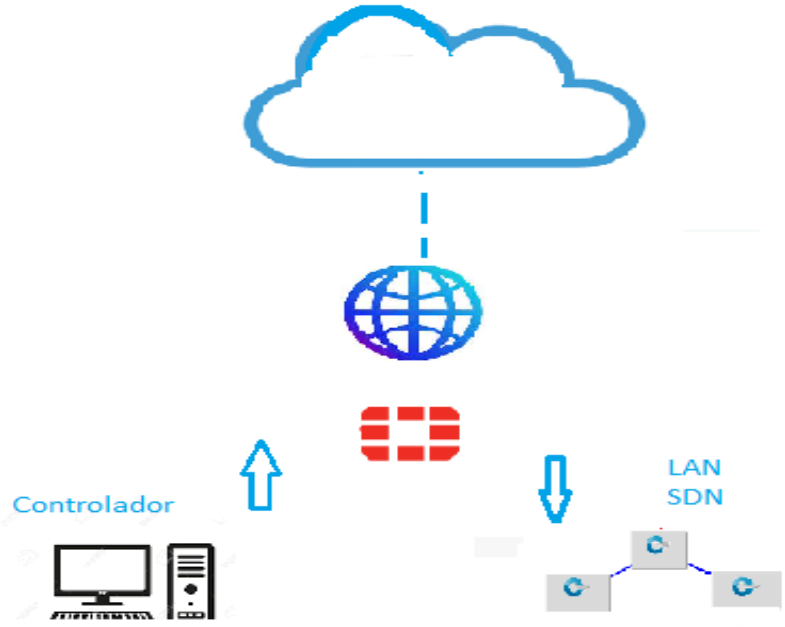


Diagrama de Implementación donde se denota el Controlador, FortiGate y la Red SDN

3.1.4 Controlador OpenDaylight

Se utilizó Ubuntu Server 20.04 para instalar OpenDaylight en su versión karaf-0.3.4-Lithium-SR4. Es necesario que previamente se haya instalado JAVA en la máquina virtual.

Existe en la web varias versiones de OpenDaylight, pero no todas las versiones se pueden utilizar para simulaciones por lo cual es necesario utilizar una versiona que permita su uso en el laboratorio.

Para poder utilizarse es necesario instalar las características necesarias para su funcionamiento con OVS, siendo las siguientes características:

- feature:install odl-restconf-all
- feature:install odl-openflowplugin-all
- feature:install odl-l2switch-all
- feature:install odl-mdsal-all
- feature:install odl-yangtools-common
- feature:install odl-dlux-all

La instalación de ODL se describe más a detalle en el ANEXO 1

3.1.5 Hosts Virtuales

Estas máquinas virtuales se han instalado en VirtualBox siguiendo los pasos detallados en la categoría de anexos y son utilizados en el desarrollo de la topología de red simulada.

Se necesita acondicionar las máquinas para poder realizar pruebas de tráfico dentro de cada una:

1. Instalación y Clonación de máquinas virtuales que utilizan el sistema operativo "Linux Mint 20.03 cinnamon.64 bit" estableciendo los parámetros de diseño como son memoria RAM, espacio en el disco a utilizar, interfaces habilitadas, unidades de red, etc.

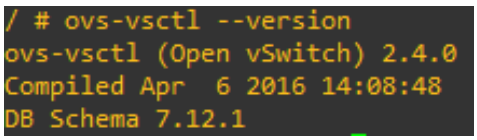
2. La máquina necesita realizar una actualización mediante comandos en terminal "apt-get upgrade" y "apt-get update", para instalar las herramientas Wireshark, Iperf, Curl, necesarios para la prueba en la red

3.1.6 Switch OpenFlow

Para poder utilizar Open vSwitch es necesario descargar su **Apliance** desde la página web de GNS3 en su versión con interfaz de manejo, posteriormente se cargar en el software de simulación, de esa manera el contenedor se descarga y se puede usar en el entorno de laboratorio.

Por defecto cada Open vSwitch posee 16 puertos que se pueden manejar, con las capacidades que brinda el protocolo OpenFlow en la versión de Open vSwitch 2.4.0 la cual se puede ver mediante el comando **ovs-vsctl—version.**

Figura 3.2 Versión OpenvSwitch



Ingreso de un comando para ver la versión utilizada en GNS3

La figura 3.3 muestra las versiones de OpenvSwitch respecto al protocolo OpenFlow, es necesario saber la versión de Openflow que se puede utilizar porque de eso depende lo que se pueda implementar.

Figura 3.3 Versiones de Openflow y OpenvSwitch

Open vSwitch	OF1.0	OF1.1	OF1.2	OF1.3	OF1.4	OF1.5
1.9 and earlier	yes	—	—	—	—	—
1.10, 1.11	yes	—	(*)	(*)	—	—
2.0, 2.1	yes	(*)	(*)	(*)	—	—
2.2	yes	(*)	(*)	(*)	(%)	(*)
2.3, 2.4	yes	yes	yes	yes	(*)	(*)
2.5, 2.6, 2.7	yes	yes	yes	yes	(*)	(*)
2.8, 2.9, 2.10, 2.11	yes	yes	yes	yes	yes	(*)
2.12	yes	yes	yes	yes	yes	yes

—Not supported. yes Supported and enabled by default (*) Supported, but missing features, and must be enabled by user. (%) Experimental, unsafe implementation.

Versión de OpenvSwitch y sus versiones de Openflow (Linux Foundation, 2022)

Al hacer uso de un controlador como OpenDaylight Lithium es necesario ver la versión de OpenFlow que soporta Open vSwitch y las características que posee ODL, es muy necesario saber la compatibilidad de OpenFlow con los switches para no tener problemas al momento de utilizar aplicaciones desarrolladas por el usuario. En la figura 3.4 se observa las versiones de OpenFlow y lo que puede soportar.

Figura 3.4 Versión de OpenFlow

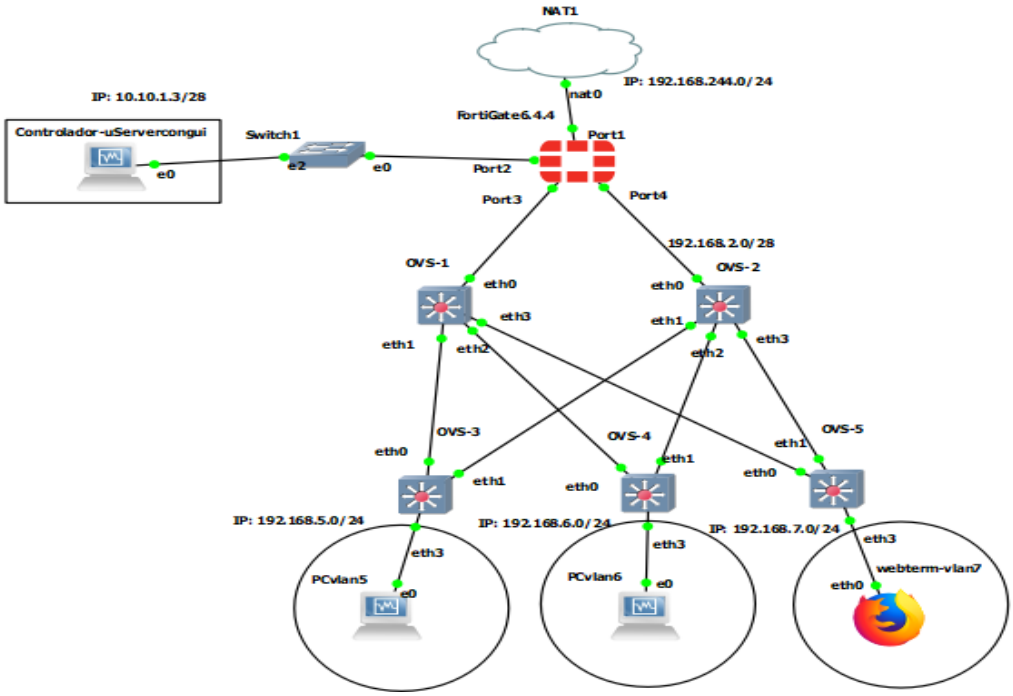
	OF 1.0	OF 1.1	OF 1.2	OF 1.3 Y OF 1.4
Puerto de entrada	X	X	X	X
Metadatos		X	X	X
Ethernet: src, dst, type	X	X	X	X
IPv4: src, dst, proto, ToS	X	X	X	X
TCP/UDP: src port, dst port	X	X	X	X
MPLS: label, traffic class		X	X	X
OpenFlow Extensible Match (OXM)			X	X
IPv6: src, dst, flow label, ICMPv6			X	X
IPv6 Extension Headers				X

Las diferentes versiones de Openflow varian en lo que es posible utilizar, siendo que versiones como OF 1.0 no soportan ciertas caracteristicas (B. Valencia, Junio 2015)

3.1.7 Topología en GNS3

Utilizando el simulador GNS3 se procede a colocar los dispositivos de red detallados anteriormente, es necesario esperar a que este activo el servidor GNS3 VM para poder utilizar el simulador sin inconvenientes. Se agrega una nube NAT para obtener conexión a la internet que utilizara Open vSwitch, asignando direcciones IP y añadiéndose también un dispositivo FortiGate versión 6.4.4 como firewall.

Figura 3.5 Topología en GNS3



Topología en GNS3 que se simulo, se utilizaron de host Máquinas Virtuales y Webterm, así como los OVS con sus respectivas características

Para saber la topología que se utilizara se ha tomado en cuenta las restricciones y las capacidades que poseen los switch OVS en la versión de contenedor de GNS3 y la capacidad del simulador, para proceder a la simulación del proyecto en escenarios que permitan el uso de aplicaciones de video.

Utilizando el emulador GNS3 el cual mediante el uso de GNS3 VM (una máquina virtual creada para poder simular características de Linux como lo es Docker), permite en un entorno de Windows la simulación de OpenvSwitch; GNS3 a diferencia de Mininet consigue una simulación de host más completa debido a sus características de virtualización y el uso de más aplicaciones de manera sencilla, esto es necesario para los escenarios previstos porque la simulación requiere de varias máquinas virtuales.

Se ha elegido el modelo de topología de núcleo contraído tomando en cuenta las restricciones que presenta el simulador, utilizándose STP y puentes de OpenvSwitch para organizar y permitir un correcto funcionamiento de la topología. GNS3 demostró que funciona de forma estable al utilizar varios OVS en la red simulada, pero va perdiendo velocidad de conexión y los switches no se muestran en el controlador al colocar muchos elementos en cascada, causando que la red no logre comunicarse con todos los elementos simulados.

La topología debido a que utiliza Open VSwitch y STP es incompatible con VXLAN, razón por la cual no se desarrolló la topología de tipo Spine and Leaf que utiliza protocolos como ECMP y se utilizó la de núcleo contraído; si se desea utilizar la topología tipo Spine and Leaf sería necesario programar flujos para que eviten bucles en la red, si se programa STP los flujos se realizan de forma automática por el controlador haciendo más sencillo que si se elimina un switch inmediatamente se compute un nuevo camino entre switches fácilmente, además Open vSwitch en esta versión presenta un error al utilizar STP junto a LACP lo cual vuelve difícil su uso.

Para realizar las pruebas de red es necesario utilizar herramientas como Wireshark o Iperf, para analizar y visualizar el tráfico de los paquetes que se están transmitiendo en la red.

3.1.8 Firewall FortiGate

FortiGate funciona de forma que se puede configurar tanto por medio de la consola como por su interfaz gráfica mediante una página web, en el modo trial es necesario tener en cuenta que algunas funciones se ven restringidas.

Para acceder a un puerto y modificar sus valores se colocan los comandos a continuación:

Figura 3.6 Comandos FortiGate puerto estático

- config system interface
- edit port1
- set mode static
- set ip 192.168.244.3/24
- set allowaccess http https ssh ping

Colocar el puerto en modo estático para el acceso mediante la interface web

Se puede ver los resultados al colocar el comando **get system interface**, además es necesario colocar el comando **set allowaccess http https ssh ping** el cual permitirá acceder a la interfaz de http, en la versión de prueba no es posible usar https.

Figura 3.7 Consola FortiGate

Consola de FortiGate con valores por defecto, se puede observar el modo estático.

3.1.9 Integración del controlador con OpenvSwitch

La configuración básica utilizada para configurar el switch OVS se encuentra en el anexo 2.

Figura 3.8 Tabla direccionamiento IP

DISPOSITIVO	INTERFAZ	DIRECCION IP	MASCARA DE SUBRED	GATEWAY PREDETERMINADO
FortiGate	port1	192.168.244.3	255.255.255.0	N/D
	port2	10.10.1.3	255.255.255.240	N/D
	port3 (Software Switch)	192.168.2.1	255.255.255.240	N/D
	port4 (Software Switch)	192.168.2.1	255.255.255.240	N/D
	vlan 5	192.168.5.1	255.255.255.240	N/D
	vlan 6	192.168.6.1	255.255.255.240	N/D
	vlan 7	192.168.7.1	255.255.255.240	N/D
OVS-1	br3	192.168.2.2	255.255.255.240	192.168.2.1
OVS-2	br3	192.168.2.3	255.255.255.240	192.168.2.1
OVS-3	br3	192.168.2.4	255.255.255.240	192.168.2.1
OVS-4	br3	192.168.2.5	255.255.255.240	192.168.2.1
OVS-5	br3	192.168.2.6	255.255.255.0	192.168.2.1
PC1	vlan 5	192.168.5.0	255.255.255.0	192.168.5.1
PC2	vlan 6	192.168.6.0	255.255.255.0	192.168.6.1
PC3	vlan 7	192.168.7.0	255.255.255.0	192.168.7.1

Lista de las direcciones IP que se utilizaron en la simulación.

3.1.10 Requerimiento de ancho de banda

La máquina en la que se realizó la simulación posee una NIC con una velocidad máxima de 100 Mbps teóricos. La medición de velocidad obtenida mediante la página web fast.com es de 89 Mbps. Dejando un intervalo de 10 Mbps que se utilizara en conexiones como la del controlador a los OVS.

Para hacer a la red más eficiente para una pequeña o mediana empresa (SOHO) se procedió a calcular parámetros de Tasa Máxima de Información (MIR) y Tasa Comprometida de Información CIR. Garantizando el bandwidth dedicado de forma adicional a la capacidad calculada del bandwidth compartido. El CIR será la velocidad de datos que se garantizará y el MIR la velocidad de datos máximo.

Figura 3.9 Ancho de Banda Meet

Ajuste de vídeo de Meet	Ancho de banda entrante mínimo	Notas
720p	2,6 Mbps	Se trata de la opción predeterminada de alta calidad, que ofrece la mejor experiencia de usuario.
480p	1,5 Mbps	
360p	1,0 Mbps	
240p	0,5 Mbps	Esta opción ofrece una mala experiencia de usuario, por lo que no se recomienda.

Ancho de banda medio por participante		
Tipo de reunión	Saliente	Entrante
Vídeo de alta definición	2,2 Mbps	1,6 Mbps
Solo audio	12 kbps	18 kbps

Ancho de banda ideal por participante		
Tipo de reunión	Saliente	Entrante
Videollamadas en HD con dos participantes	1,7 Mbps	1,7 Mbps
Videollamadas en grupo	0,7 Mbps	2,0 Mbps

Valores de ancho de banda requerido para videoconferencias en Meet (Google Meet, 2022)

Para comprimir el video es necesario el uso de un codec, en el caso de estudio se ha utilizado un codec h.264, para calcular el tamaño del video standard de 360*640 a 30fps sin comprimir se calcula utilizando los frames, la cantidad de pixeles, los colores por píxel y los bytes por color como se ve en la formula a continuación.

$$\left(360 * 640\frac{pixeles}{frames}\right)\left(30\frac{frames}{sec}\right)\left(3\frac{colores}{pixel}\right)\left(1\frac{bytes}{color}\right) = BW \qquad Ec.(3.1)$$

$$BW = 19.77Mbps$$

Estos 19.77 Mbps son necesarios por pantalla no comprimidos, para el uso del codec h.264 se ha referido a la siguiente imagen:

Figura 3.10 Ancho de banda codec H.264

Name	Resolution	Link (Mbps)	Bitrate (Mbps)	Video (kbps)	Audio (kbps)
240p	424x240	1.0	0.64	576	64
360p	640x360	1.5	0.96	896	64
432p	768x432	1.8	1.15	1088	64
480p	848x480	2.0	1.28	1216	64
480p HQ	848x480	2.5	1.60	1536	64
576p	1024x576	3.0	1.92	1856	64
576p HQ	1024x576	3.5	2.24	2176	64
720p	1280x720	4.0	2.56	2496	64
720p HQ	1280x720	5.0	3.20	3072	128
1080p	1920x1080	8.0	5.12	4992	128
1080p HQ	1920x1080	12.0	7.68	7552	128
1080p Superbit	1920x1080	N/A	20.32	20000	320

Se describen los valores de ancho de banda requerido para códec h.264 (Lighterra, 2012)

Se ha tomado como valor de ancho de banda 1.5 Mbps necesario para el codec h.264, el bandwidth máximo disponible es de 15 Mbps, de este total se tomará el 75% para el uso del simulador GNS3 dando un total de 11.25 Mbps para utilizar, a los cuales se le quitaran 2 Mbps reservados para la conexión de la red con el controlador dando 9.25 Mbps de uso para la red LAN. Se desea garantizar un bandwidth del 80% requerido para el codec h.264 para que no se presenten pérdidas durante la transmisión.

$$MIR = \ Total\ de\ Ancho\ de\ banda\ disponible = \ 9.25\ Mbps \qquad Ec.(3.2)$$

$$CIR = BWrequerido * porcentaje\ que\ se\ garantiza \qquad Ec.(3.3)$$

$$CIR = 1.5\ Mbps * 80\%$$

$$CIR = 1.2\ Mbps$$

$$BWusuario = \frac{BW\ disponible}{Numero\ usuarios\ simultaneos} \qquad Ec.(3.4)$$

$$Numero\ de\ ususarios = \ 9.25\ Mbps/1.5Mbps = 6{,}16\ usuarios$$

Se resumen los resultados en la siguiente tabla:

Tabla 3.2 Resultados cálculos ancho de banda

Nombre	**Usuarios**	**Mbps codec h.264**	**% BW CIR (Mbps)**	**CIR/ Usuario**	**CIR/vlan (Mbps)**	**MIR/vlan (Mbps)**
vlan 5	3	1.5	80%	1.2 Mbps	3.6	4.5
vlan 6	2	1.5	80%	1.2 Mbps	2.4	3
vlan 7	1	1.5	80%	1.2 Mbps	1.2	1.5

Tabla acerca del ancho de banda calculados en la simulación

3.1.11 Latencia, Jitter, perdida de paquetes

Según (Guamán & Condoy, 2021) los resultados del rendimiento de una red SDN está relacionado con la funcionalidad del controlador. En su red propuesta se caracterizan diferentes resultados en el rendimiento, siendo OpenDaylight el que posee un mayor desempeño en las mediciones de QoS, con un promedio de 0,936 ms de latencia, con variación de 0,72 ms y 1,97 % de fluctuación con pérdida de paquetes, para la red ipv4 su latencia fue de 14,629 ms enviando un archivo de tamaño de 65507 bytes y un jitter de 0.010.

Para el manejo de estas pérdidas, así como otras configuraciones requeridas se han hecho sus respectivos ajustes requeridos que se detallan en los anexos 2 y 3.

CAPÍTULO 4

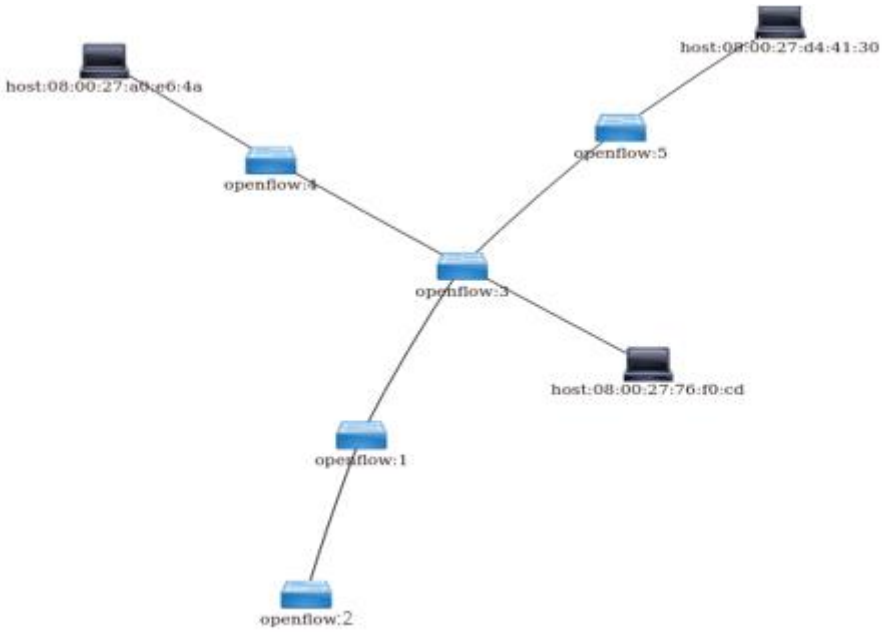
4.1 SIMULACIÓN DE LA RED

4.1.1 Escenario 1

Utilizando VLC Media Player transmitiendo entre los clientes host 1 y host 2 mediante el protocolo http en el puerto 8080, estos reciben su dirección IP dependiendo de que vlan se encuentren mediante DHCP, gestionado por el firewall FortiGate como se ve en el anexo 3.

Una vez iniciado el controlador se generan automáticamente las reglas de flujos necesarias para indicar el camino de los flujos. Se conectan el servidor de video 192.168.5.2 al cliente en 192.168.6.2.

Figura 4.1 Topología en Opendaylight



El controlador Opendaylight detecta a los switches que utilizan OpenFlow y los muestra en su interface gráfica.

La forma en la que se presenta la red es debido al protocolo STP que evita los bucles de red, una vez obtenida la conexión de los hosts hacia internet y entre ellos procedemos a ver los flujos de la red pertenecientes al OVS.

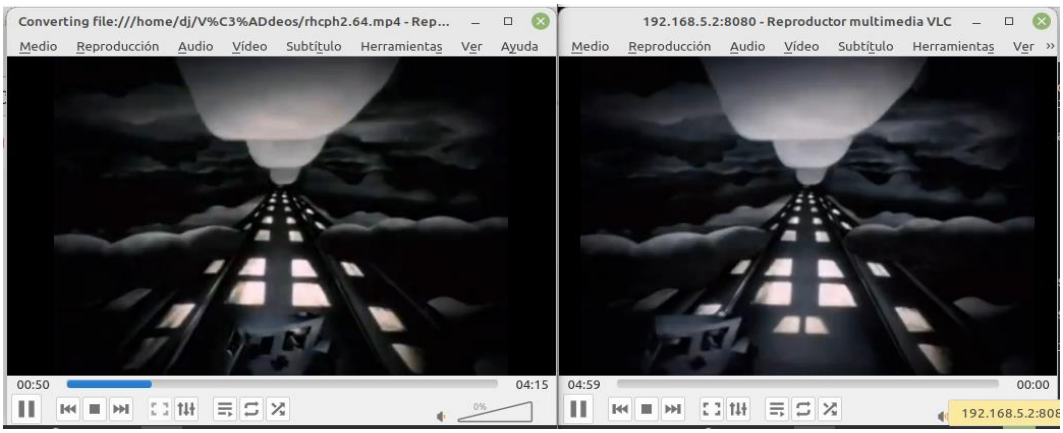
Figura 4.2 Flujos de OpenvSwitch

```
/ # ovs-ofctl dump-flows br3
NXST_FLOW reply (xid=0x4):
 cookie=0x2b00000000000005, duration=11986.202s, table=0, n_packets=7182, n_bytes=610470, idle_age=1, priority=100,dl_type=0
x88cc actions=CONTROLLER:65535
 cookie=0x102, duration=11986.189s, table=0, n_packets=81924, n_bytes=79097417, idle_age=98, priority=100,in_port=1 actions=
output:2,output:3
 cookie=0x11, duration=11986.180s, table=0, n_packets=54258, n_bytes=39340288, idle_age=186, priority=100,in_port=3 actions=
output:1
 cookie=0x2b00000000000098, duration=940.767s, table=0, n_packets=37, n_bytes=2590, idle_age=44, priority=2,in_port=2 action
s=output:1,output:3,CONTROLLER:65535
 cookie=0x2b00000000000099, duration=940.767s, table=0, n_packets=0, n_bytes=0, idle_age=11980, priority=2,in_port=1 actions
=output:2,output:3
 cookie=0x2b0000000000009a, duration=940.767s, table=0, n_packets=0, n_bytes=0, idle_age=11980, priority=2,in_port=3 actions
=output:2,output:1,CONTROLLER:65535
 cookie=0x2b00000000000005, duration=11986.200s, table=0, n_packets=9, n_bytes=630, idle_age=568, priority=0 actions=drop
/ #
```

Se pueden observar flujos que posee el puente br3 en el OpenvSwitch.

Para la prueba ser utilizo un video comprimido con codec H.264, el cual se pasó sin ningún tipo de interferencia para poder ver el retraso en la emisión en condiciones sin ningún retraso.

Figura 4.3 Video Transmisor y Receptor.



Se realizo una prueba para ver la transmisión de un video, el emisor está en la izquierda y el receptor a la derecha.

Utilizando iperf3 procedemos a generar tráfico UDP de forma que la red se vea congestionada al valor de ancho de banda calculado y poder observar el comportamiento. Para esto se utilizaron los comandos:

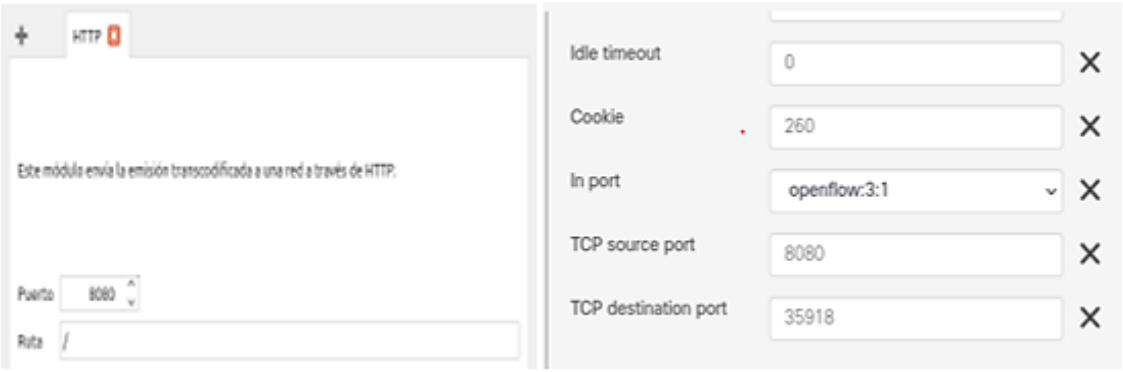
En el Servidor **iperf3 -s**.

En el Cliente **iperf3 -c 192.168.5.2 -V -u -b4m -t180 -i5 port 5201.**

Se generará 4 Mbps en UDP, un total de 180 segundos con un intervalo de presentación de datos cada 5 segundos a través del puerto 5201. También se ha configurado prioridad alta al

tráfico procedente desde el puerto 8080 perteneciente al puerto http hacia el puerto 35918, esto con el fin de que se tenga la menor perdida de paquetes.

Figura 4.4 Instalación flujo con OpenFlow Manager



Se coloca una regla para dar prioridad a un puerto TCP 8080 en el que se realiza una transmisión en VLC

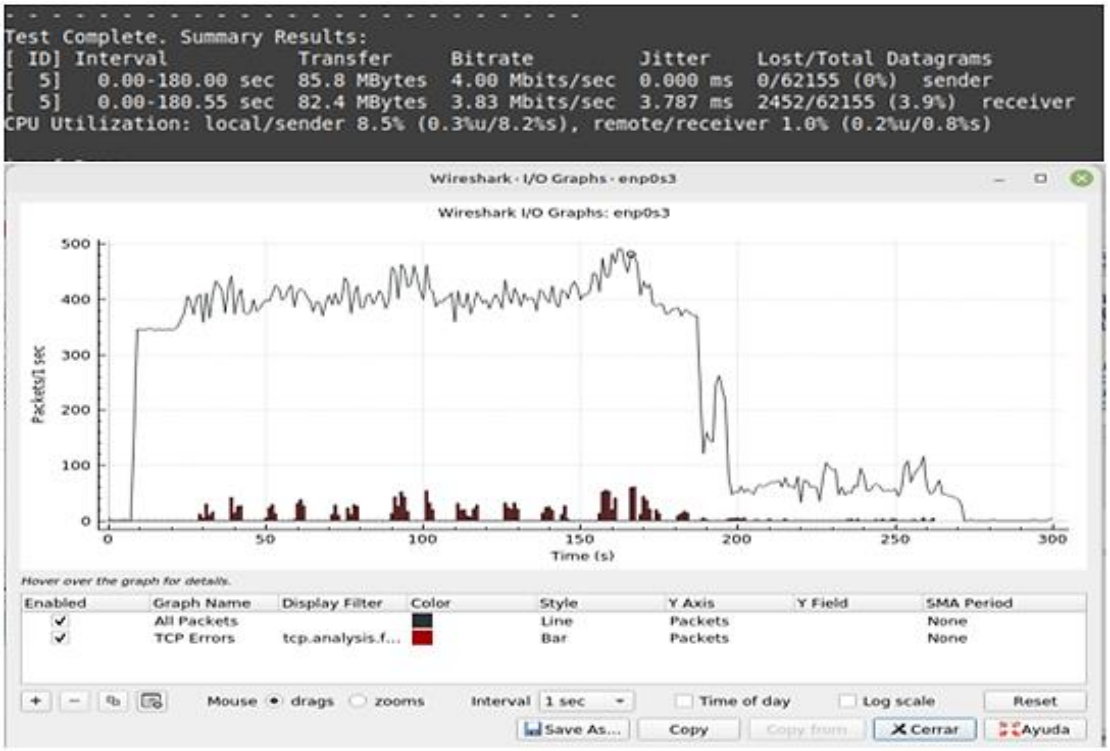
El protocolo http usa TCP razón por la cual se producen retardos en el video, pero no se ve perdidas de pixeles a comparación con UDP, pero si se aplica un filtro que favorezca al puerto UDP la perdida de pixeles disminuye también, en ambos casos se puede observar un retraso en la transmisión del video de unos segundos.

Figura 4.5 Video Emisor y Receptor

Se puede observar el retraso producido en la red al realizar throughput, el emisor está en la izquierda y el receptor a la derecha.

Mediante el uso de Iperf3 y Wireshark se puede observar el retraso, jitter y la perdida de paquetes producida durante la transmisión observándose una pérdida de 3.9% en el receptor como muestra en la figura 4.6.

Figura 4.6 Captura en Wireshark e Iperf



En la imagen se puede ver en Wireshark e Iperf la prueba de stress de la red durante la transmisión de un video.

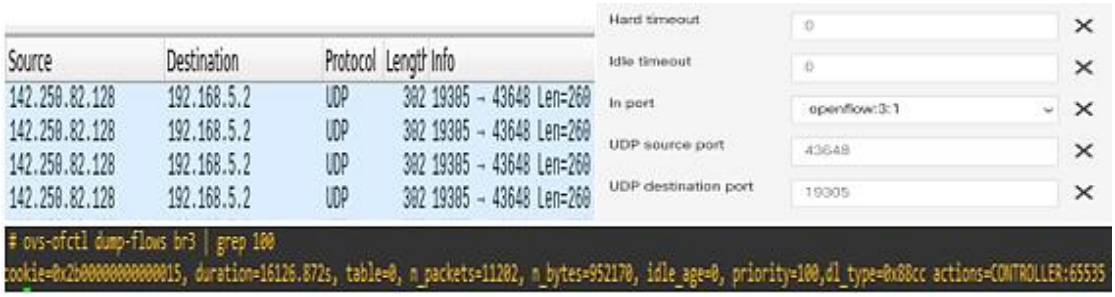
4.1.2 Escenario 2:

Para el segundo escenario se utilizó Google Meet, para videoconferencias Google sugiere que se debe tener un mínimo de 3.2 Mbps de ancho de banda en cualquier situación.

Google Meet usa las direcciones disponibles de la red IP 142.250.82.0/24 para el consumidor, sugiriendo tener una latencia de 50 ms respecto a la dirección DNS de Google 8.8.8.8 para tener una llamada con calidad HD y 100 ms para una calidad SD.

Con el uso de Wireshark se obtuvo el puerto UDP que se utiliza siendo el puerto 43648, mediante el controlador Opendaylight y Openflow Manager se colocó con prioridad al flujo proveniente de ese puerto UDP, el flujo se puede visualizar al buscar el switch con una prioridad de 100.

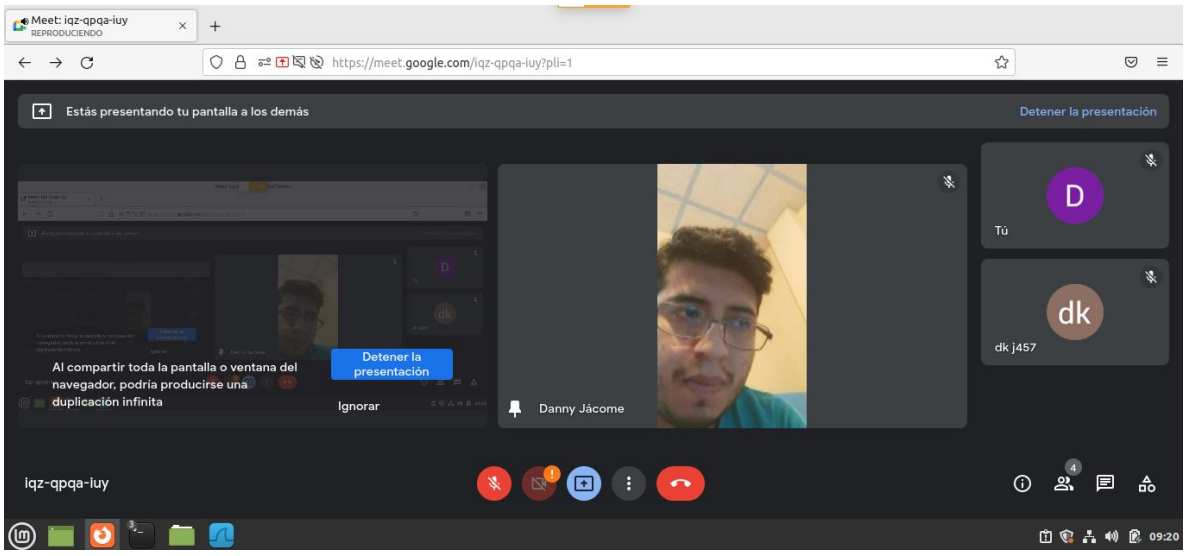
Figura 4.7 Flujos UDP en Open VSwitch



Se puede ver la colocación de un nuevo flujo en un Switch OVS que da prioridad a los paquetes UDP que provienen del puerto 43648 con destino al puerto 19385.

Para la prueba de funcionamiento inicial se utilizaron host, uno externo a la red y 2 host pertenecientes a la red para poder visualizar el desarrollo de la llamada, un host hacía de cámara, mientras otro presentaba pantalla.

Figura 4.8 Prueba utilizando Google Meet



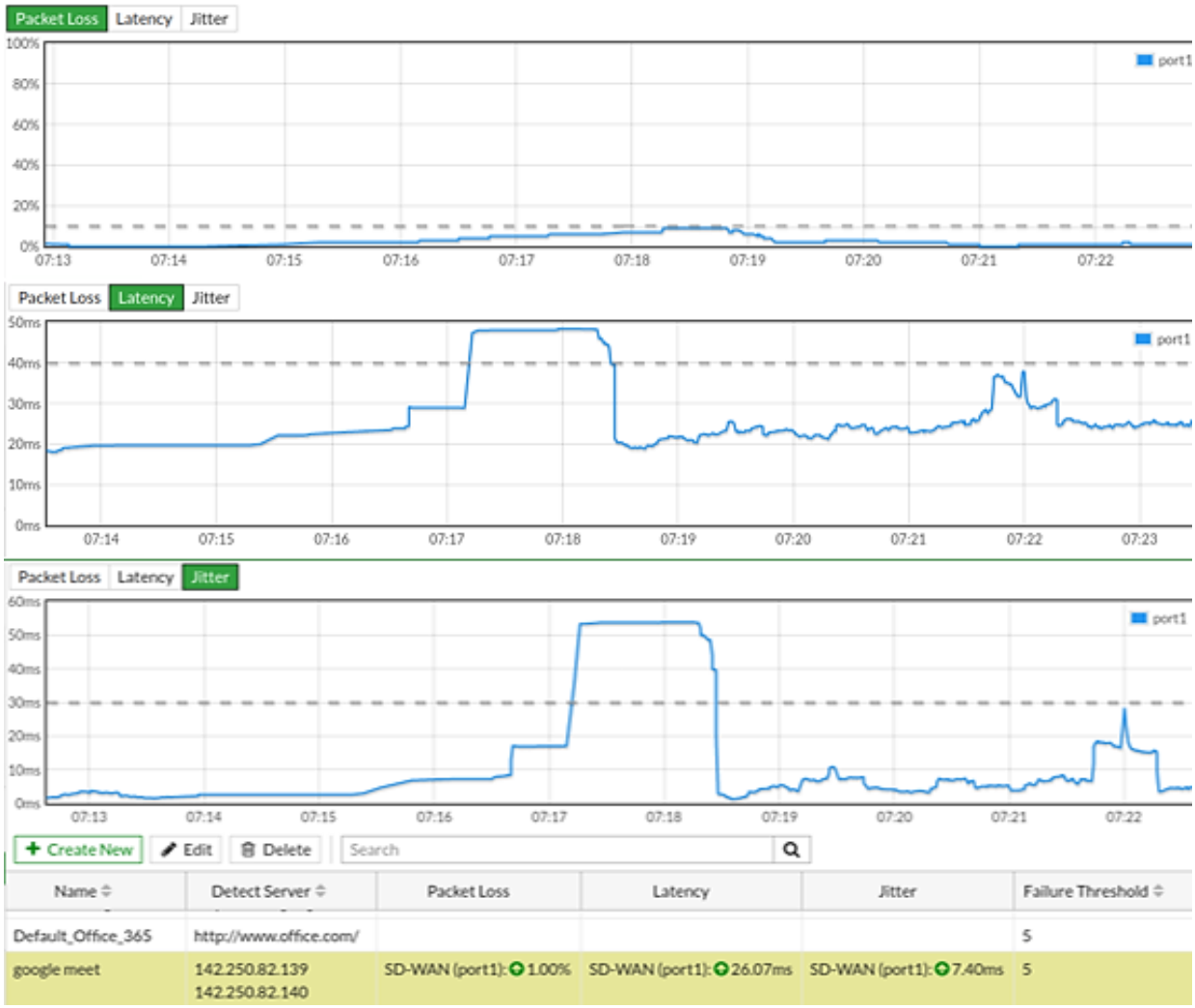
Se realiza una prueba en Google Meet realizando una presentación en un host mientras otros hosts se encuentran en la reunión.

La llamada presento un nivel un jitter y una pérdida de paquetes altos hasta que se insertó la dirección IP provista por Google Meet, momento en el que se ajustó a los valores establecidos de Garantía de Nivel de Servicio (SLA) que son 10% de perdida de paquetes, 40 ms de latencia y 30 ms de jitter.

A continuación, La Figura 4.9 se presenta la gráfica respectiva del desarrollo de la llamada. Los valores de los parámetros de perdida obtenidos, corresponden a un momento antes y

después de aplicar las políticas de FortiGate, mejorando la llamada y como se puede apreciar en el grafico los valores de latencia, jitter y perdida de paquetes disminuyeron debajo de lo programado.

Figura 4.9 Jitter, Perdida de Paquetes y Latencia



FortiGate antes y después de aplicar performance SLA con un límite de perdida de paquetes del 10%, Latencia de 40 ms y Jitter de 30 ms.

Se volvió a realizar una prueba más, pero esta vez se procedió a inundar el canal InterVLAN con paquetes UDP mediante Iperf3 con 2 Mbps, de esta forma se estreso el canal y se obtuvo los siguientes valores.

Figura 4.10 Throughput en la topología

```
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-30.00  sec  7.15 MBytes  2.00 Mbits/sec  0.000 ms  0/5180 (0%)  sender
[  5]   0.00-31.05  sec  6.76 MBytes  1.83 Mbits/sec  7.592 ms  283/5180 (5.5%)  receiver
CPU Utilization: local/sender 5.9% (0.2%u/5.7%s), remote/receiver 0.0% (0.0%u/0.0%s)
```

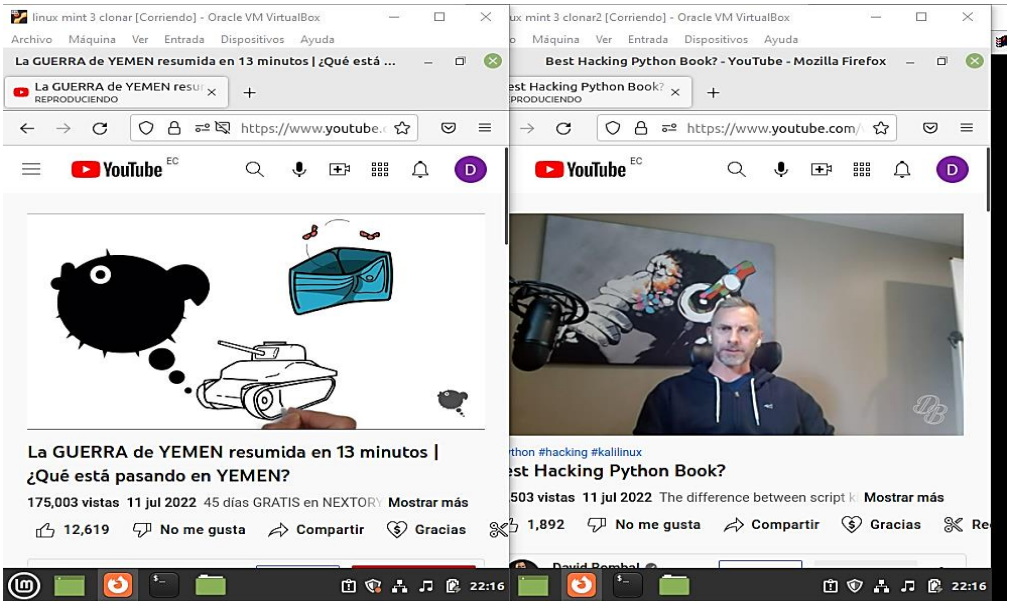
Name	Detect Server	Packet Loss	Latency	Jitter
google meet	142.250.82.139 142.250.82.140	SD-WAN (port1): 2.00%	SD-WAN (port1): 24.59ms	SD-WAN (port1): 6.18ms

Se puede observar que los valores de perdida de paquetes no pasaron el 10% y el jitter se mantuvo debajo del límite.

4.1.3 Escenario 3

Para un tercer escenario se procedió acceder a YouTube una plataforma que es conocida por su servicio de compartición de videos para poder ver el comportamiento. Se coloco una política para que se dé prioridad a los paquetes de YouTube como se ve en el anexo 3, de forma que se exista una conexión rápida y se tomó como un valor referente el video a 360p.

Figura 4.11 Prueba en YouTube

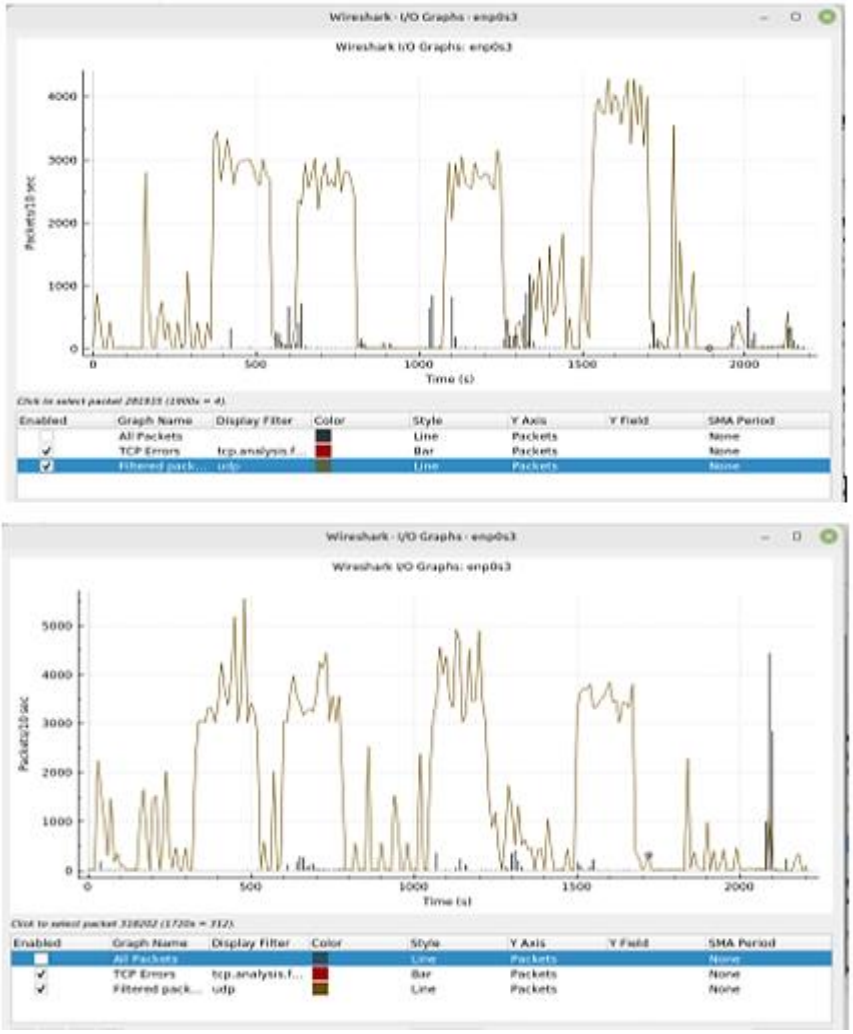


Se comprobó el correcto funcionamiento en la red de YouTube y no se vio que el video se deba cargar para reproducirse.

Con Wireshark se filtraron los flujos de datos UDP que se presentan en la figura 4.12, se puede ver los momentos en los que se realizó el throughput la cantidad de paquetes observables aumenta. Los picos que se observan son los paquetes que YouTube necesita para poder seguir la carga de video, también en color negro se muestran los errores TCP que

se ven en la red, estos errores se generaron más en los momentos en los que se inició el estrés, pero no son muchos.

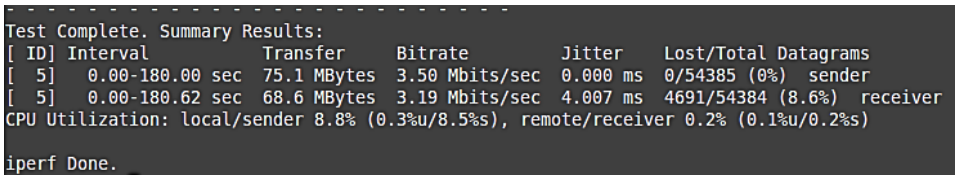
Figura 4.12 Aplicar Wireshark entre host 1 y host 2



Captura de paquetes de YouTube aplicando un stress en la red, se utilizó Wireshark en un host 1 y host 2 que se reprodujeron al mismo tiempo.

La prueba de estrés de la red se realizó con Iperf, la respuesta de Iperf se muestra en la figura 4.13, cabe denotar que hubo un jitter de 4.007 ms y una pérdida de paquetes de 8.6%.

Figura 4.13 Estrés en la red usando Iperf



Se utilizó Iperf al mismo tiempo que se utilizaba YouTube con el fin de observar si existen cortes en la carga del video o si bajaba la calidad de la reproducción.

CAPÍTULO 5

5.1 ANÁLISIS DE RESULTADOS

5.1.1 Escenario 1

En las pruebas entre hosts de diferente VLAN se observó que para una transferencia de un archivo de 16 MB sometido a throughput en el canal transmitiendo 85.8 MBytes, obtuvo una pérdida de 3.9%, a un Jitter de 3,9 ms en el receptor y el Bitrate de 3.83 Mbits/s.

Este resultado se obtuvo al enviar paquetes UDP en un ancho de banda de 4 Mbps, un total de 180 segundos.

5.1.2 Escenario 2

En la llamada mediante Google Meet sin las políticas de SLA se obtuvieron valores de perdida de paquetes del 10%, latencia máxima de 48 ms, y un jitter de 53 ms. Posterior a colocar la política se obtuvo un valor de perdida de paquetes de 1%, latencia de 38 ms y un jitter de 29 ms como se observa en la figura 4.9

Al realizar una prueba de throughput en el canal se obtuvieron al transferir 7.15 MBytes, el receptor obtuvo un Bitrate de 1.83 Mbits/s un jitter de 7.592 ms y una pérdida de paquetes de 5.5 %, la prueba se hizo con paquetes UDP de 2 Mbps. En el firewall se observó una pérdida de paquetes de 2 %, una latencia de 24.59 ms y un jitter de 6.18 ms como se observa en la figura 4.10.

5.1.3 Escenario 3

Se realizaron varias pruebas de throughput para videos a una calidad de 360 p, los análisis de Wireshark se observan en la figura 4.12 utilizando 2 host, el estrés del ancho de banda con paquetes UDP se realizó con 3Mbps y 4 Mbps.

Se observan picos de máximo 4.5 Mbps en el host 1 y en el host 2 de 3.5 Mbps en la prueba con 3Mbps de estrés en el canal. Con la prueba de 4 Mbps de estrés se observa un valor máximo de 4.2 Mbps en el host 1 y de 3.7 Mbps en el host 2.

En la prueba con Iperf3 se encontró un valor de jitter de 4.007 ms y una pérdida de paquetes de 8.6% en el receptor para una transferencia de 75.1 MBytes durante la carga de video.

CONCLUSIONES

La arquitectura de las redes SDN al separar el plano de datos del plano control, permite manejar la red de una forma centralizada y simple, permitiendo que el despliegue y la gestión de la misma sea más sencilla; siendo un aspecto importante de esta red simulada en GNS3 la separación de cada contenedor y máquina virtual haciendo posible un escenario más real. La tecnología SDN permite un mejor manejo de los datos debido a sus características, lo cual aplicados servicios como el videostreaming hacen posible que estos puedan tener una mayor calidad.

Durante el desarrollo de la topología de la red FortiGate resulto ser una herramienta muy útil debido a las políticas de seguridad que puede implementar en todo tipo de redes, su capacidad de utilizar SD-WAN junto a SLA y otros servicios han brindado soluciones que Open vSwitch solamente no las puede realizar en esta versión de simulación en GNS3. GNS3-VM ha resultado ser muy útil al momento de simular contenedores necesarios para son los Open vSwitch o Webterm, que son elementos indispensables al momento de probar esta red definida por software (SDN).

El controlador OpenDaylight en la simulación trabaja versátilmente, debido a que posee una amplia documentación a la que es posible acceder fácilmente, siendo una ventaja también el uso de aplicaciones como es OpenFlow Manager. El colocar flujos que tengan prioridades a ciertos puertos ya sean UDP o TCP mediante el controlador, el firewall FortiGate, o comandos en la consola de OpenvSwitch han permitido que la red tenga menos perdidas al momento de utilizar servicios de aplicaciones video.

En el capítulo de análisis de resultados se realizó pruebas en tres escenarios donde se evaluó el desempeño de la red. La conclusión de cada escenario se presenta a continuación:

- En el primer escenario, luego de las respectivas evaluaciones, se pudo obtener que los parámetros analizados se mantienen en un intervalo que permite observar la transmisión sin muchos cortes.
- En el segundo escenario al evaluar la aplicación con los valores de SLA de FortiGate se observó una mejora en la videoconferencia dado que mantuvo la llamada incluso con la prueba de estrés de red, la misma que se mantenía, aunque la calidad de la llamada decayera.

- En el escenario tres al realizar la prueba se puede apreciar que, aunque hubo picos en la red no hubo un corte en la carga del video, y la calidad se mantuvo en 360p que fueron fijados para YouTube.

RECOMENDACIONES

Para realizar la simulación fueron necesarios recursos como el tamaño de la memoria RAM, un disco de estado sólido (SSD), un procesador Intel Core i7, esto debido a que la simulación se vuelve muy lenta al utilizar computadores de menos recursos haciendo muy difícil que la simulación se vea afectada.

Se utilizaron imágenes disponibles para simulación de GNS3, los cuales pueden estar desfasados de las versiones actuales que utilizan controladores más actualizados que la versión Opendaylight Lithium o FortiGate 4.4. Para poder utilizar estas versiones es mejor utilizar directamente en máquinas que permitan su uso.

Los programas de OpenvSwitch y FortiGate no son versiones completas puesto que se utilizan para simulación, por lo que presentan fallas por errores o porque no soportan ciertas configuraciones por lo que es necesario siempre referenciarse en la documentación disponible.

BIBLIOGRAFÍA

Quanti . (15 de Febrero de 2022). *Que es FortiGate: Conociendo el Firewall*. Obtenido de https://quanti.com.mx/articulos/conociendo-el-firewall-fortigate/

2022 TECNASA INC. (21 de Octubre de 2022). *Fortinet lider en el cuadrado magico de Gartner*. Obtenido de Comunicado de prensa de Fortinet.: https://www.tecnasa.com/es/fortinet-es-nuevamente-nombrado-lider-en-el-cuadrante-magico-de-gartner-2021-para-infraestructura-wan-edge-situandose-en-el-puesto-mas-alto-en-capacidad-de-ejecucion/

A. Crouch, H. K. (2010, Octubre). *Forwarding and Control Element Separation (ForCES) RFC 6041*. Retrieved from https://pike.lysator.liu.se/docs/ietf/rfc/60/rfc6041.xml#:~:text=Introduction-,The%20Forwarding%20and%20Control%20Element%20Separation%20(ForCES)%20protocol%20defines%20a,IP%20routers%20and%20similar%20devices.

Alexander La rosa, P. F. (Julio de 2021). *Pandora FMS*. Obtenido de ¿Cómo las redes definidas por Software cambian nuestra visión sobre las redes?: https://pandorafms.com/blog/es/redes-definidas-por-software/

ARÉVALO, J. J. (2020). EMULACIÓN DE UNA RED SD-WAN (SOFTWARE-DEFINED WIDE AREA NETWORK) UTILIZANDO TECNOLOGÍA FORTINET Y EL SOFTWARE GNS3. *TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES.* ESCUELA POLITÉCNICA NACIONAL, Quito.

Ariganello, R. C. (2017). Guía de estudio para la certificación CCNA Routing y Switching. . . 4º Edición actualizada,). Obtenido de https://books.google.com.ec/books?id=tpBFDwAAQBAJ&lpg=PT287&ots=k6P5x9WN2K&dq=El%20mejor%20ancho%20de%20banda%20disponible%20en%20la%20ruta%20es%20el%20del%20enlace%20que%20tiene%20menor%20ancho%20de%20banda.&pg=PA1#v=onepage&q=El%20mejor%20ancho%20de%20b

aruba. (2022). *aruba a hewlett packard enterprise company*. Retrieved from ¿Qué es SD-WAN?: https://www.arubanetworks.com/es/faq/que-es-sd-wan/

Azodolmolky, S. (2013). Software Defined Networking with Openflow. Birmingham: Packt Publishing Ltd.

B. Valencia, S. S. (Junio 2015). Mininet: una herramienta versátil para emulación y prototipado de redes definidas por Software. *Entre Ciencia e Ingenieria*. Obtenido de Mininet: una herramienta versátil para emulación y prototipado de redes definidas por Software.

Black Box . (2022). *¿Qué es el formato de codificación de vídeo H.264 y porqué se está convirtiendo en el estándar de la industria en compresión de vídeo?* Obtenido de https://www.blackbox.com.mx/mx-mx/page/40830/Recursos/Technical/black-box-explica/Multimedia/Compresion-de-video-H264#:~:text=264%3F-,H.,Picture%20Experts%20Group%20(MPEG).

Blanco Pérez, R. (2019). Avanzando hacia una red auto-adaptativa: simulación de redes definidas por software (SDN) mediante el simulador GNS3. *Tesis de grado de Ingenieria de Tecnología de la Información.* Universidad de Valladolid, Valladolid.

Cisco. (2017). *Tutorial sobre calidad de servicio (QOS) de vídeo.* Obtenido de https://www.cisco.com/c/es_mx/support/docs/quality-of-service-qos/qos-video/212134-Video-Quality-of-Service-QOS-Tutorial.html

Cisco Systems, Inc. (4 de junio de 2009). *QoS preguntas*. Obtenido de Preguntas más frecuentes (FAQ) : https://www.cisco.com/c/es_mx/support/docs/quality-of-service-qos/qos-policing/22833-qos-faq.html

Cisco Systems, Inc. (2022). *What Is SD-WAN?* Retrieved from https://www.cisco.com/c/en/us/solutions/enterprise-networks/sd-wan/what-is-sd-wan.html

Craven, C. (November de 2020). Obtenido de https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/

Dinecom. (2022). *¿Cómo se explica el uso de un jitter buffer, y cuáles son las tecnologías de Voz sobre IP –VoIP- que normalmente lo incluyen?* Obtenido de Que se supone que mide el jitter buffer: https://dinecom.cl/2020/videoconferencia-que-se-supone-que-mide-el-jitter-buffer/

Erickson, D. (2013, August 16). The Beacon OpenFlow Controller. (A. f. Machinery, Ed.) doi:10.1145/2491185.2491189

Fortinet, Inc. (2022). *About us*. Obtenido de About Us: https://www.fortinet.com/corporate/about-us/about-us

Fortinet, Inc. (2022). *Secure SD-WAN*. Obtenido de Fortinet: https://www.fortinet.com/lat/products/sd-wan

Galaxy Technologies LLC. (2021). *Getting Started with GNS3*. Retrieved from GNS3: https://docs.gns3.com/docs/

Google Meet. (2022). *Elige la configuración, la tasa de bits y la resolución del codificador en vivo*. Obtenido de https://support.google.com/youtube/answer/2853702?hl=es-419

Guamán, D. J., & Condoy, J. R. (2021). EVALUACIÓN DEL RENDIMIENTO DE UN PROTOTIPO SDN (SOFTWARE DEFINED NETWORKING) BAJO EL PROTOCOLO OPENFLOW UTILIZANDO HERRAMIENTAS OPEN SOURCE EN UN ENTORNO VIRTUALIZADO. *Trabajo de titulación previo a la obtención del título de: Ingenieros de Sistemas.* UNIVERSIDAD POLITÉCNICA SALESIANA, Quito.

Ionos. (11 de 04 de 2022). *¿Qué es la latencia?* Obtenido de https://www.ionos.es/digitalguide/servidores/know-how/latencia/

Jiménez, D. C. (2020). SIMULACIÓN DE UNA RED SDN DE VIDEOVIGILANCIA IP. *Trabajo de Fin de grado.* Universitat Politecnica de Valencia, Valencia.

Juan A. Michell Martín, G. A. (2015). *Compresión de Video.* Retrieved from https://ocw.unican.es/pluginfile.php/171/course/section/75/tema_2.1.pdf

Lighterra. (2012). *Video Encoding Settings for H.264*. Retrieved from https://www.lighterra.com/papers/videoencodingh264/

Linux Foundation. (2016). *What Is Open vSwitch?* Retrieved from Open vSwitch: https://docs.openvswitch.org/en/latest/intro/what-is-ovs/

Linux Foundation. (2022, Julio). *Using Openflow*. Retrieved from https://docs.openvswitch.org/en/latest/faq/openflow/

Medina, P. A. (2007). Diseño de un ISP considernado criterios de calidad de servicio para la transmisión de voz, datos y vídeo utilizando el estándar IEEE 802.16 (WIMAX) para cubrir el área norte de la ciudad de Quito. 60.

NetApp. (2022). *¿Qué son los contenedores?* Obtenido de https://www.netapp.com/es/devops-solutions/what-are-containers/

Open Networking Foundation. (2015). *OpenFlow Switch Specification Version 1.5.1 (Protocol version 0x06).* Retrieved from https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

Open Networking Foundation. (2022). *An Innovative Combination of Standards and Open Source Software*. Retrieved from https://opennetworking.org/software-defined-standards/overview/

OpenDaylight Project. (2021). *Opendaylight The Linux Foundation Project*. Retrieved from https://www.opendaylight.org/

Ortiz, J. (2017). *¿Qué es el codec VP9?* Obtenido de https://javierortiz.mx/glosario/codec-de-video/codec-vp9/

Pinilla, R. Á. (2015). Estudio de las Redes definidas por Software mediante el desarrollo de escenarios virtuales basados en el controlador Opendaylight. *Trabajo fin de Master.* Universidad Politécnica de Madrid, Madrid.

Pozuelo, A. G. (2016). Despliegue de una Infraestructura de Red Definida por Software. *GRADO EN INGENIERÍA TELEMÁTICA.* UNIVERSIDAD CARLOS III DE MADRID - ESCUELA POLITÉCNICA SUPERIOR, Madrid.

Rapp, V. J. (2022). *Redes definidas por software*. Obtenido de ¿Qué son las redes definidas por software (SDN)?: https://www.vmware.com/es/topics/glossary/content/software-defined-networking.html

Telectrónika. (29 de abril de 2018). *GNS3 Guía Introductoria: Características y Requerimientos Mínimos*. Obtenido de https://www.telectronika.com/articulos/ti/que-es-gns3/

Vizard, M. (2014, Octubre). *SDN Will Transform the Delivery of IT Services*. Retrieved from channel insider: https://www.channelinsider.com/managed-services/sdn-will-transform-the-delivery-of-it-services/

Winchester School of Art. (2012). *TechShop: Digital Video Basics*. Retrieved from http://wsa.wikidot.com/tbm:video-basics#toc9

Zhu, A. (2018, July 22). *OpenFlow Switch: What Is It and How Does it Work?* Retrieved from https://medium.com/@AriaZhu/openflow-switch-what-is-it-and-how-does-it-work-7589ea7ea29c

ANEXOS

9.1 ANEXO 1: INSTALACIÓN DE SOFTWARE REQUERIDO

GNS3VM y GNS3

Una vez se ha instalado GNS3 se debe instalar GNS3VM para poder utilizar OpenvSwitch en Windows debido a que fue creado para Linux o realizar toda la simulación en Linux, la máquina virtual de GNS3VM se diseñó especialmente para VMware por lo que es necesario instalar preferiblemente este.

Los adaptadores de red que se utilizaran se han cambiado por VMnet2(Bridge) y VMnet8(NAT) debido a problemas que presentaba durante la simulación.

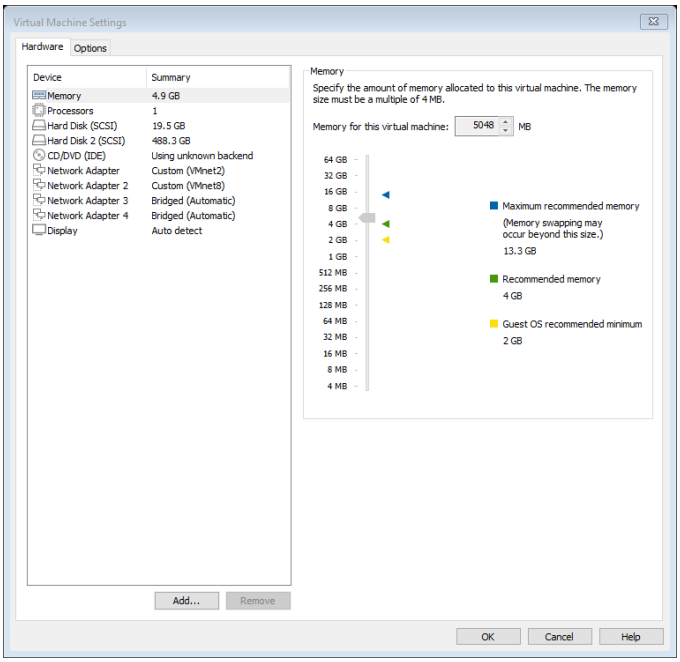


Figura 9.1 Característica de la Máquina Virtual GNS3 VM

Nota: Debido a problemas con la simulación de OVS se utilizó VMware Workstation Pro 16.2.0 build-18760230, junto a la versión de GNS3 2.2.26 y la versión de OpenvSwitch 2.4.0.

INSTALACIÓN DE OPENDAYLIGHT

Se procede a actualizar el S.O. Ubuntu 20 server mediante un gestor de paquetes de apt. Mediante el comando apt-get procedemos a actualizar el S.O.

- **sudo apt-get upgrade && sudo apt-get update -y**

Al utilizar la interfaz gráfica de Ubuntu Server es necesario instalar un entorno de escritorio, esta interfaz permitirá utilizarlo como si tuviera instalando la versión de escritorio.

Para poder instalar la interfaz es necesario incluir los paquetes de tasksel, con el siguiente comando:

- **sudo apt install tasksel**

Una vez instalado se puede usar tasksel para instalar un entorno de escritorio. Algunos entornos de escritorio necesitan más recursos del sistema (como GNOME), mientras que otros utilizan menos recursos del sistema (como Xfce, MATE, etc.).

- **sudo apt-get install lightdm**
- **Para poder visualizar la interfaz es necesario el comando**
- **sudo service lightdm start**

Es necesario para el funcionamiento de esta versión de ODL el tener instalado los paquetes de java.

- **sudo apt-get -y install openjdk-8-jre**

En caso de tener más de una versión de java la correcta versión a utilizar es la 8.

- **sudo update-alternatives --config java**

Procedemos a descargar la versión de Opendaylight Lithium que permite la simulación de la red

- **curl -XGET – O**

Y esperamos a que se termine de descargar el archivo necesario.

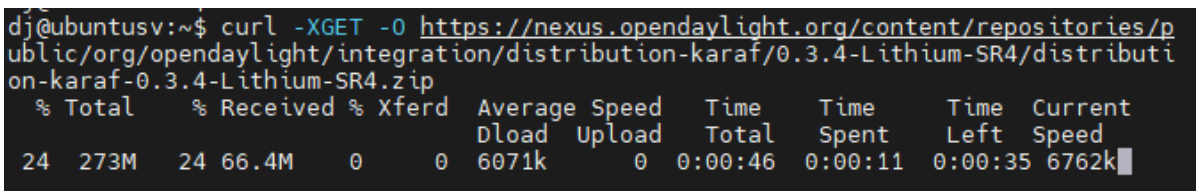


Figura 9.2 Descarga de Opendaylight Lithium

Una vez descargado el archivo .zip procedemos a colocar el comando **unzip** el cual permitirá descomprimir el archivo.

Para iniciar OpenDaylight es necesario entrar en el archivo descomprimido y buscar la ubicación del archivo karaf usamos el comando **sudo ./bin/karaf**

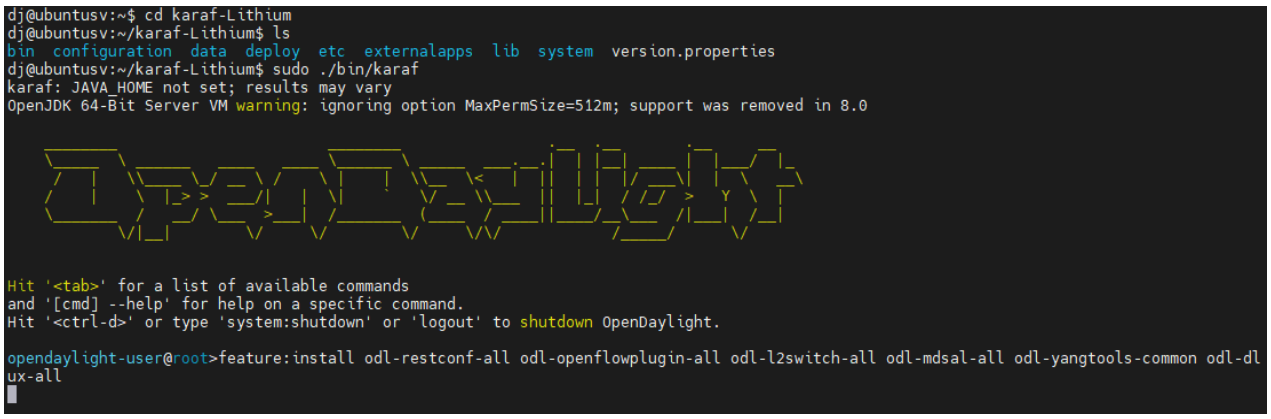


Figura 9.3 Instalación de las características de Opendaylight

Es necesario instalar las propiedades necesarias para la versión de OpenDaylight que se utilizará, las mismas que pueden diferir respecto a la versión que se utilice, colocamos el comando:

Feature:install odl-restconf-all odl-openflowplugin-all odl-l2switch-all odl-mdsal-all odl-yangtools-common odl-dlux-all

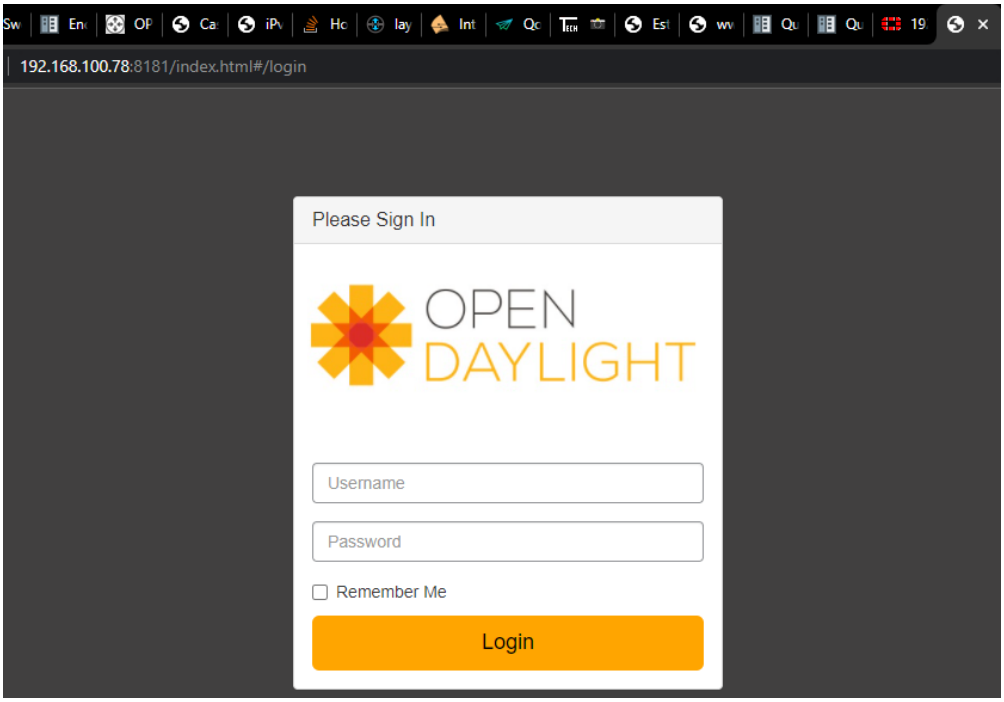


Figura 9.4 Página Web Opendaylight

Al acceder a la interfaz gráfica (FrontEnd) de Opendaylight se necesita ingresar en un browser la dirección URL de la interfaz http.

Las credenciales de acceso por defecto son **Username:** admin **Password:** admin.

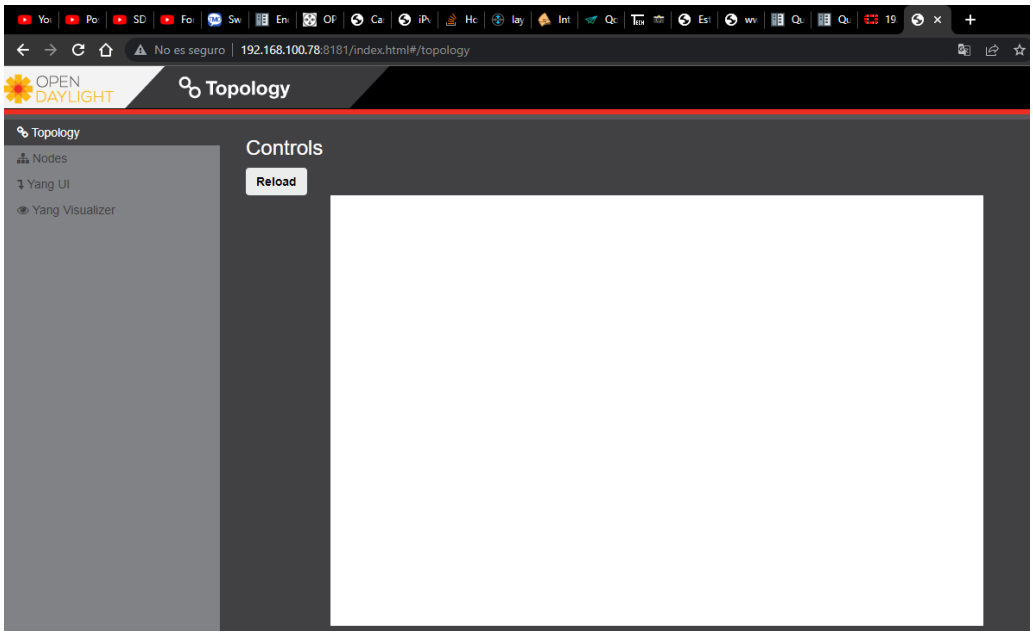


Figura 9.5 Espacio para la Topología dado por Opendaylight

NOTA: Puede ser necesario en caso de problemas de carga de la página de inicio de OpenDaylight un reset, por lo que se requiere usar el comando **sudo ./karaf clean,** posterior a eso se de volver a iniciar OpenDaylight y colocar nuevamente el comando **feature:install.**

INSTALACIÓN DE OPENFLOW MANAGER

Para poder integrar Openflow Manager es necesario tener instalado las features en Opendaylight que son: **odl-restconf-all, odl-openflowplugin-all, odl-l2switch-all,**

Para copiar el programa desde GitHub se utiliza el comando:

Sudo apt install git -y

Git clone https://github.com/CiscoDevNet/OpenDaylight-Openflow-App

También es necesario la instalación de nodejs, npm y grunt:

- **Sudo apt -y install nodejs**
- **Sudo apt-get install -y npm**
- **Sudo npm install -g grunt-cli**

Para poner en funcionamiento Openflow Manager se debe usar **sudo grunt**.

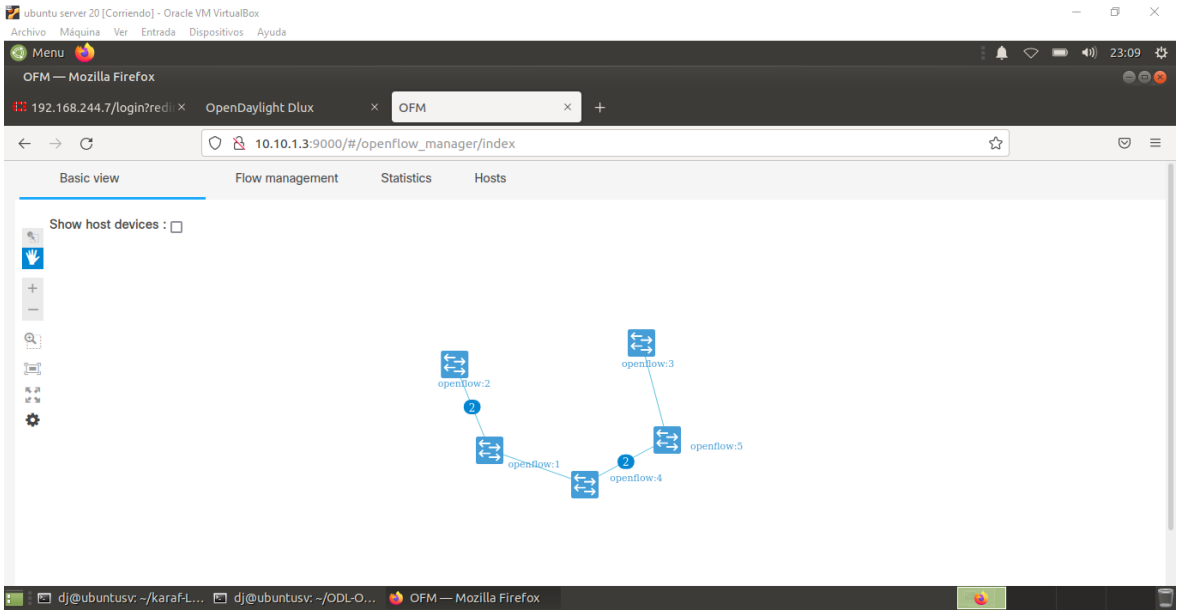
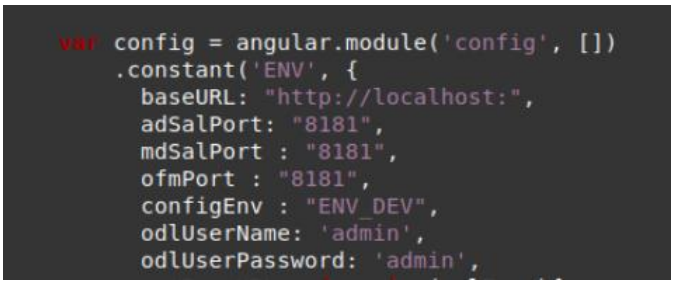


Figura 9.6 Presentación de una Topología en Openflow Manager

Nota: Si se utiliza Openflow Manager desde una dirección diferente a la de Opendaylight es necesario modificar el archivo con el nombre **/ofm/src/common/config/env.module.js.**

Utilizando un editor como nano si es necesario se modifica **http://localhost:** por la dirección IP del servidor OpenDaylight.



```
var config = angular.module('config', [])
    .constant('ENV', {
      baseURL: "http://localhost:",
      adSalPort: "8181",
      mdSalPort : "8181",
      ofmPort : "8181",
      configEnv : "ENV_DEV",
      odlUserName: 'admin',
      odlUserPassword: 'admin',
```

Figura 9.7 Características de OpenFlow Manager

Los elementos que se muestran en Opendaylight se pueden ver en Openflow Manager, de forma que se indica en la Figura 9.8.

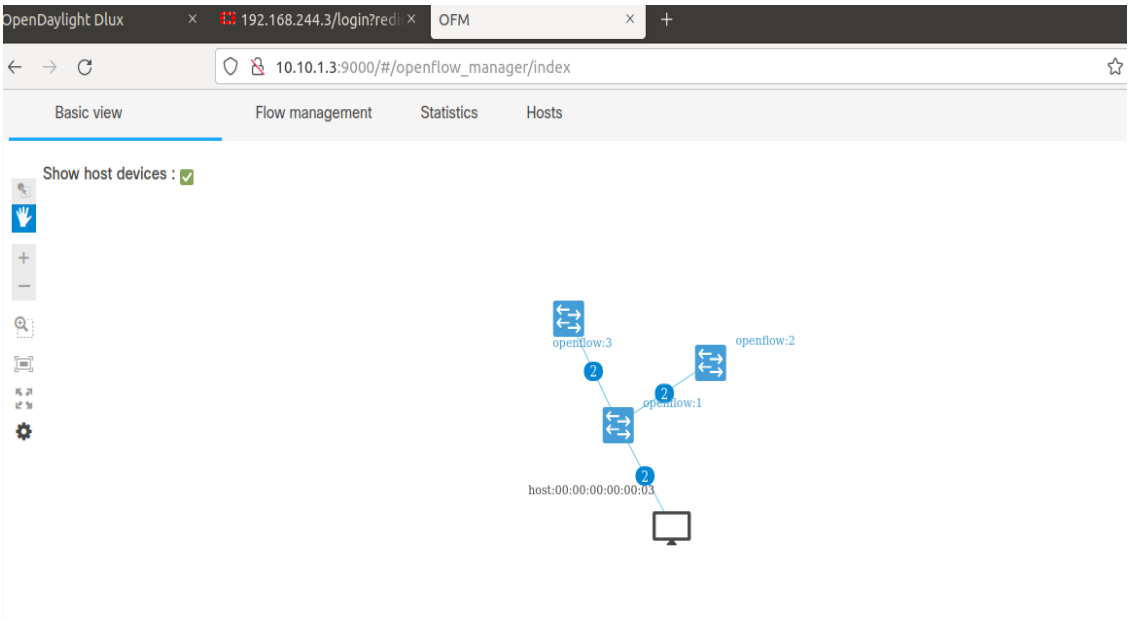


Figura 9.8 Elementos utilizados presentados en OpenFlow Manager

Openflow Management permite agregar una entrada de flujo, se debe elegir un OpenvSwitch y procedemos a añadir o eliminar sus reglas de flujos:

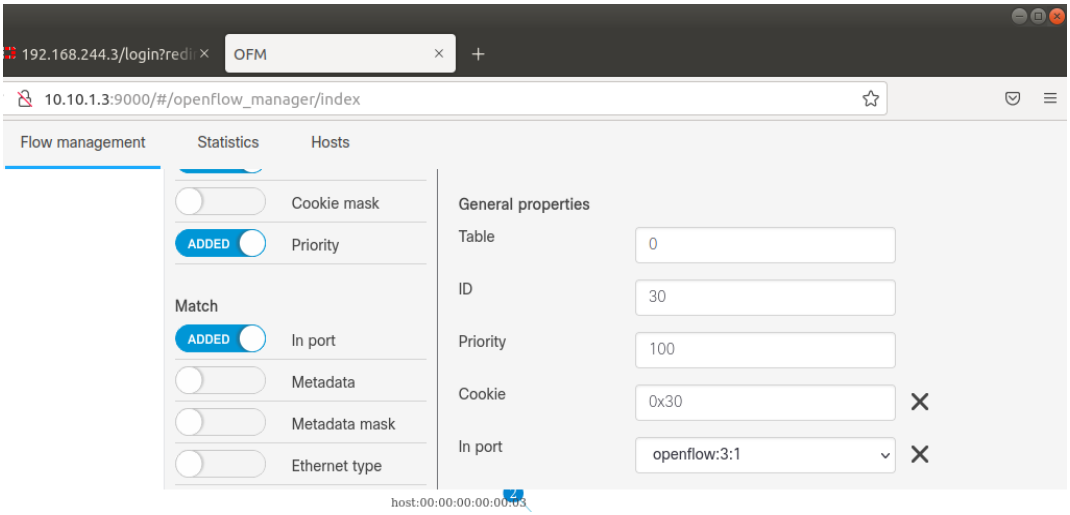


Figura 9.9 Flujo creado con OpenFlow Manager

En General Properties se debe colocar los datos apropiados para los flujos como son Table, ID, Priority, que deben configurarse de la forma correcta para el funcionamiento del switch OVS.

OpenFlow Manager permite crear los flujos que rigen sobre un switch OVS de forma que es más amigable para el usuario, pero también permite ver una vista previa de lo que programara en lenguaje Json.

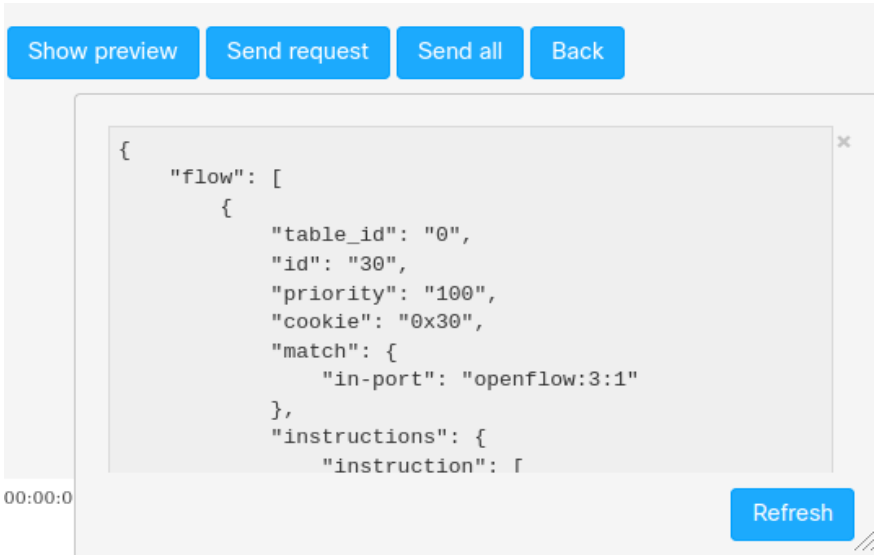


Figura 9.10 Flujo a colocarse en formato Json

Una vez enviado el flujo programado a switch OVS se puede ver que se ha programado en la consola de OVS con los datos colocados en la interfaz de Openflow Manager, e la imagen se puede observar el flujo colocado junto a los que se generaron de Opendaylight.

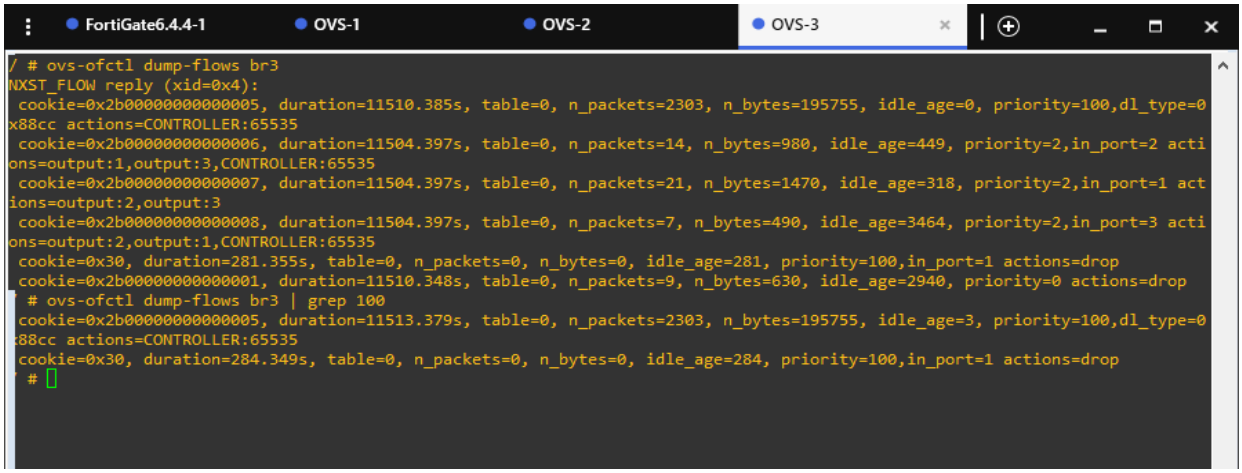


Figura 9.11 Presentación de los flujos instalados en un OpenvSwitch

INSTALACIÓN DE FORTIGATE 6.4.4

Para poder utilizar el appliance de FortiGate es necesario descarga la versión para simulación desde la página web de GNS3 y proceder a instalarlo mediante GNS3

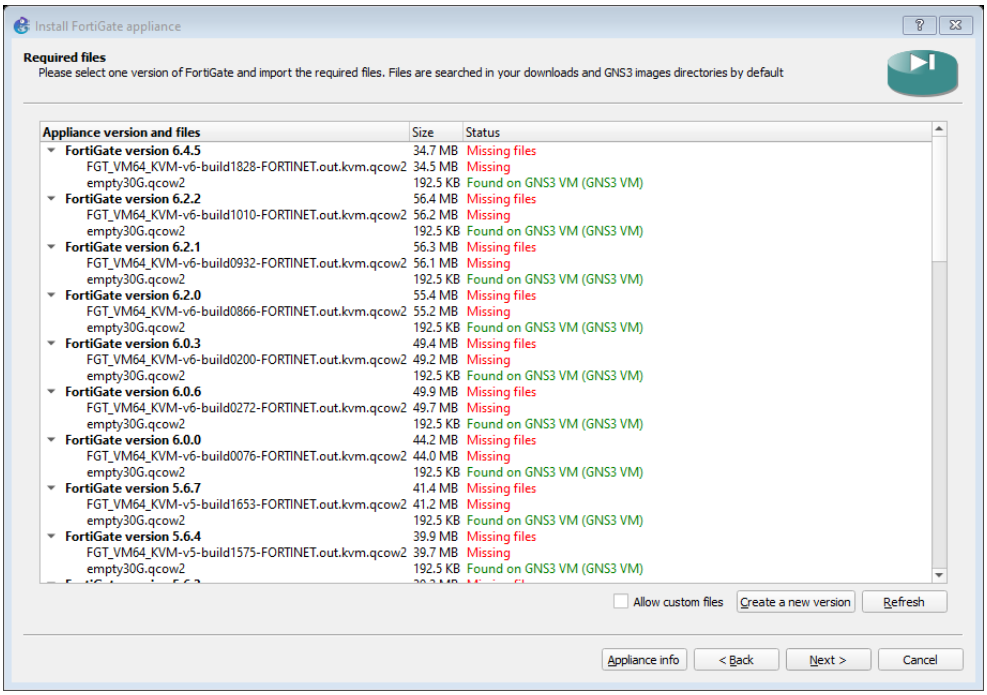


Figura 9.12 Instalación de FortiGate

Una vez instalado se podrá utilizar FortiGate en Gns3 en modo de trial de 15 días.

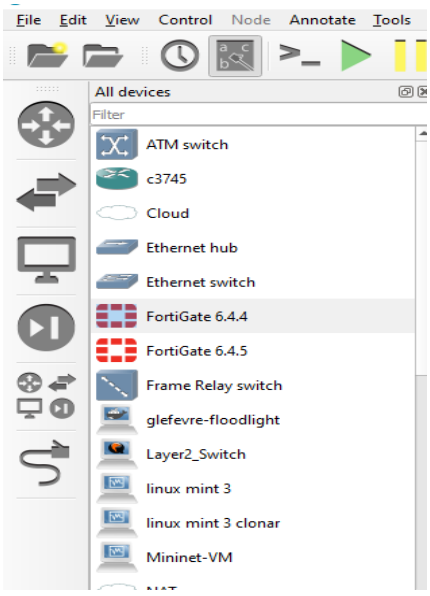


Figura 9.13 Icono de FortiGate en Gns3

INSTALACIÓN DE OPENVSWITCH

Para poder utilizar el appliance de OpenvSwitch es necesario descargar la versión de OVS denominada **Open vSwitch with management interface** desde la página web de GNS3, una vez descargado se importa a GNS3

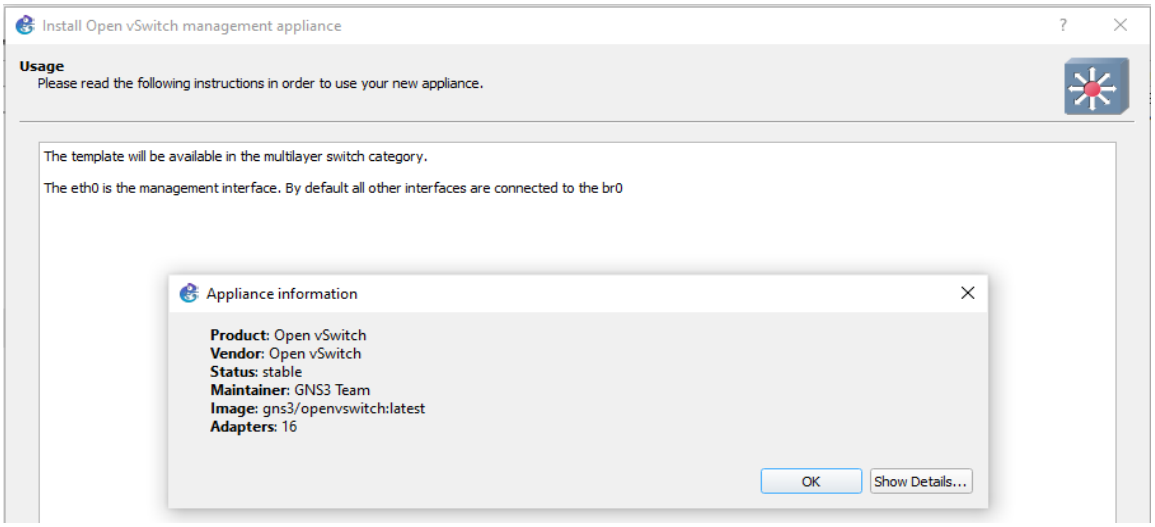


Figura 9.14 Características del appliance OpenvSwitch

Una vez descargado se debe arrastrar al sitio de trabajo para que se configure la imagen que se colocará en GNS3 VM, esta es una imagen Docker permitirá simular un OVS como se ve en la figura 9.15.

En caso de una mala instalación es necesario eliminar la imagen desde GNS3 VM para lo cual es necesario ingresar en el modo terminal de la máquina virtual, con el comando **sudo Docker Images** se puede ver las **appliance** que se han descargado de Docker Hub, para poder saber que contenedor está en funcionamiento o su información básica se utiliza el comando **sudo Docker ps**. Para eliminar la imagen se utiliza el comando **sudo Docker rmi imageID.**

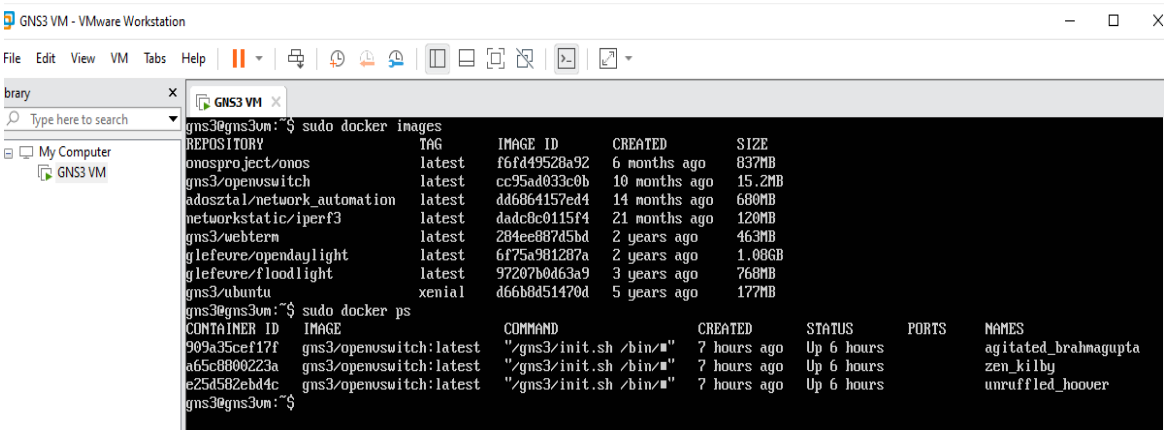


Figura 9.15 Imágenes Docker de uso de Gns3

9.2 ANEXO 2: CONFIGURACIONES EN OPENVSWITCH

CONFIGURACIÓN DE OVS

Para el funcionamiento de los switches OVS se colocó los puertos en los denominados puentes de OVS, colocándoles un **Datapath** y una dirección MAC para poder identificar de que switch se trata, se ha colocado el controlador para una subred diferente a la de los switches por lo que es necesario colocarlo en modo o**ut-of-band**, el modo de fallo está colocado en **Secure** de forma que no permita ningún tráfico en caso de desconexión del controlador.

- ovs-vsctl del-port br0 eth1
- ovs-vsctl del-port br0 eth2
- ovs-vsctl del-port br0 eth3
- ovs-vsctl add-port br3 eth0
- ovs-vsctl add-port br2 eth1
- ovs-vsctl add-port br2 eth2
- ovs-vsctl add-port br2 eth3
- ovs-vsctl set bridge br3 other-config:datapath-id=0000000000000001
- ovs-vsctl set bridge br3 other-config:hwaddr=00:00:00:00:00:01
- ovs-vsctl del-controller br3
- ovs-vsctl set-controller br3 tcp:10.10.1.3:6633
- ip addr add 192.168.2.2/28 dev br3
- route add default gw 192.168.2.1 br3
- ovs-vsctl add-port br3 patch1
- ovs-vsctl set interface patch1 type=patch
- ovs-vsctl set interface patch1 options:peer=patch2
- ovs-vsctl add-port br2 patch2
- ovs-vsctl set interface patch2 type=patch
- ovs-vsctl set interface patch2 options:peer=patch1
- ovs-vsctl set Bridge br2 stp_enable=true
- ovs-vsctl set controller br3 connection-mode=out-of-band
- ovs-vsctl set-fail-mode br3 secure

Figura 9.16 Comandos colocados en un OVS

Para la restricción del ancho de banda se utiliza el comando en kbps

- ovs-vsctl set Interface eth3 ingress_policing_rate=3500

Para los puertos vlan se utilizó los siguientes comandos referidos a los puertos:

- ovs-vsctl set port eth0 vlan_mode=trunk
- ovs-vsctl set port eth0 tag=6
- ovs-vsctl set port eth3 trunk=5
- ovs-vsctl set port eth3 vlan_mode=native-untagged
- ovs-vsctl set port eth3 tag=6

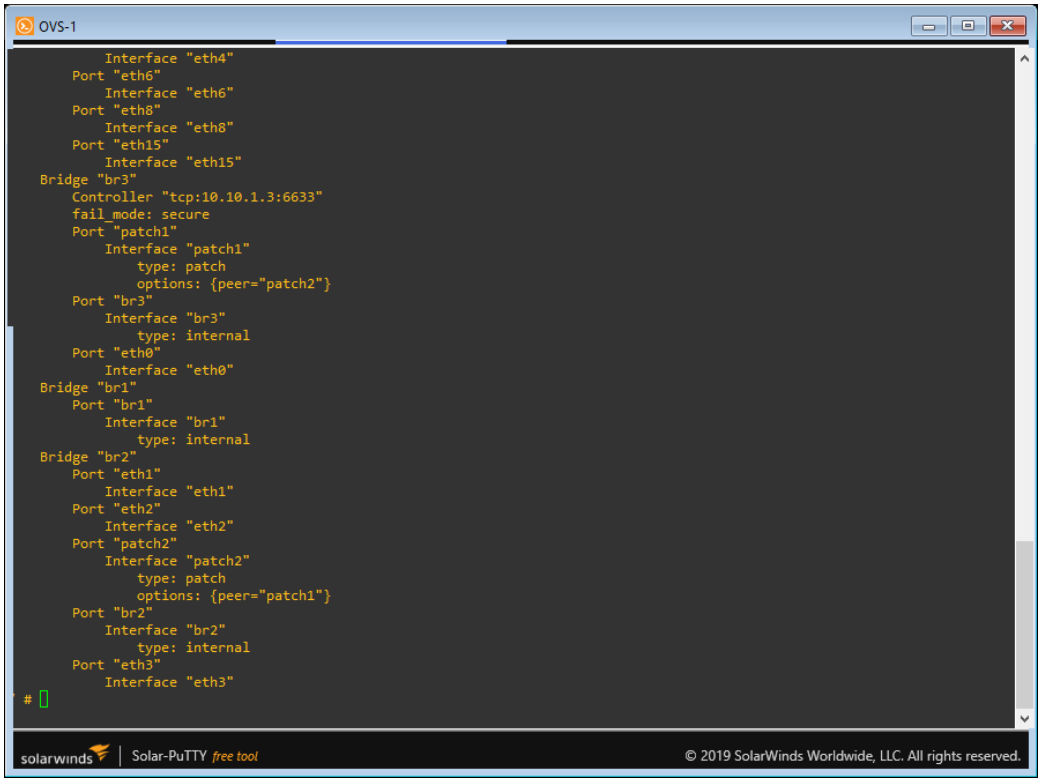


Figura 9.17 Instalación de comandos de configuración en OpenvSwitch

9.3 ANEXO 3: CONFIGURACIONES EN FORTIGATE

CONFIGURACIÓN DE UNA INTERFAZ FÍSICA COMO SD-WAN

Se procede a configurar la interfaz de entrada port1 como SD-WAN de forma que se pueda aplicar políticas de FortiGate como el análisis de Jitter, Latency.

Es necesario en la pestaña de Interfaces en el apartado Network elegir la opción LAN con una interfaz de tipo físico; la dirección IP en modo manual es la 192.168.244.3/255.255.255.0, debido a que de esta forma si es necesario un reinicio de FortiGate la dirección IP que por default se obtiene no utilice DHCP sino por una dirección fija para poder ingresar siempre a la interfaz gráfica por la misma dirección.

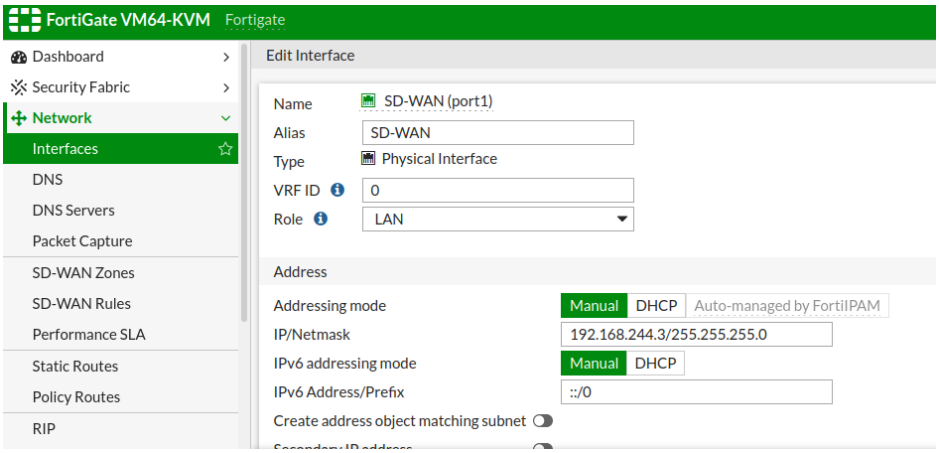


Figura 9.18 Configuración de un puerto SD-WAN

Los permisos de acceso se pueden modificar en el mismo panel en la parte inferior.

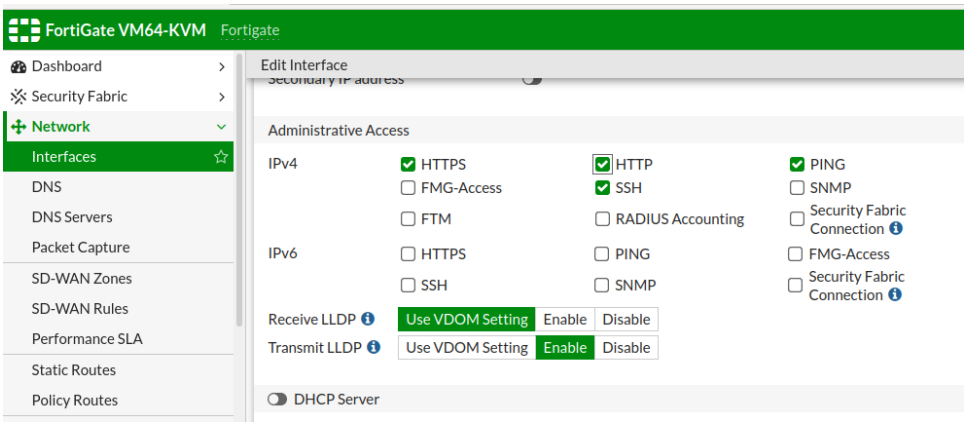


Figura 9.19 Configuración de accesos administrativos

Para seleccionar que la interfaz será SD-WAN escogemos en la pestaña Network- SD-WAN Zone el puerto que se utilizará en este caso port1.

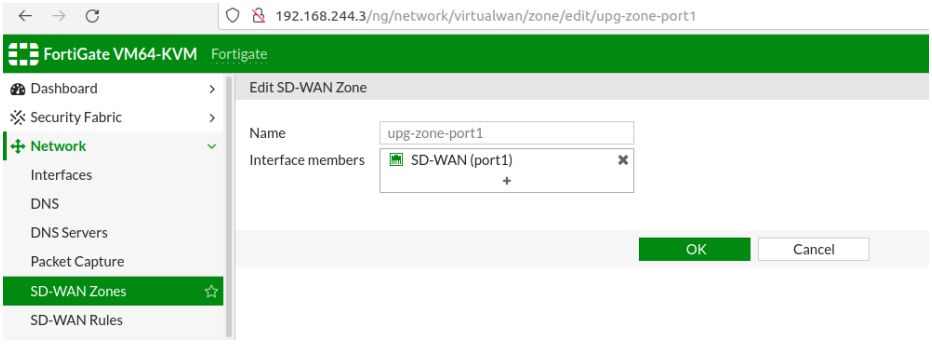


Figura 9.20 Configuración de una zona SD-WAN

Se colocan las reglas que se cumplirá, en **Source address** se customiza las direcciones que se permite; en **Destination** se elige las subredes que tendrán permitido acceder a las direcciones web.

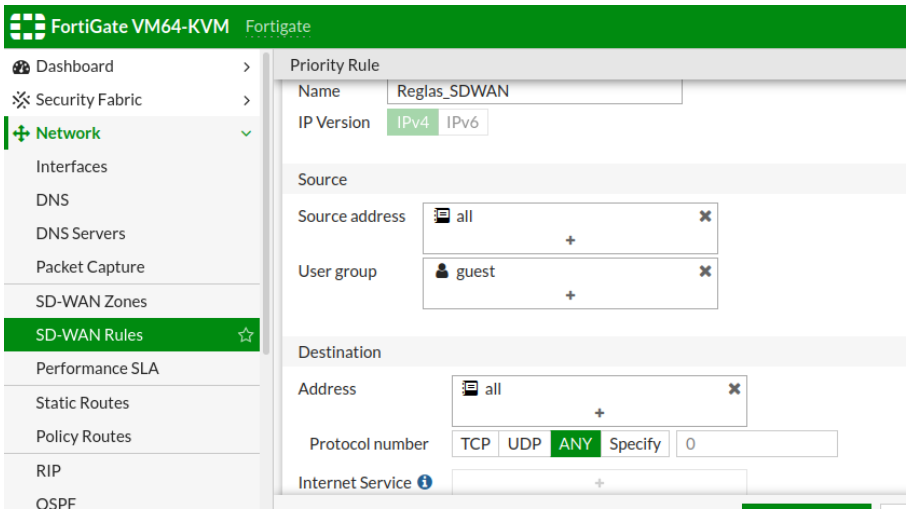


Figura 9.21 Configuración Reglas SD-WAN

Para poder utilizar las reglas SD-WAN y que las interfaces de salida utilicen el mayor ancho de banda se seleccionó la opción Maximize Bandwith (SLA), en caso de tener más enlaces SD-WAN esta configuración realizara un balanceo de carga entre interfaces que coincidan con los blancos SLA.

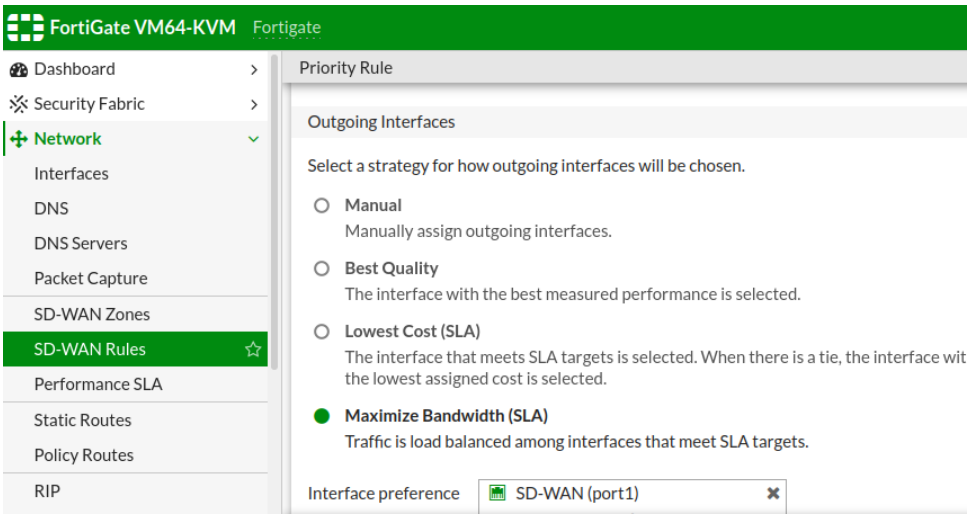


Figura 9.22 Configuración Reglas para el ancho de banda en SD-WAN

Para poder utilizar SLA en una página objetivo en específico es necesario colocarla en la pestaña Required SLA Target de esta forma se podrá ver el efecto que tiene en la red en una serie de gráficas que son ping, latencia, jitter de la red; es necesario colocar las reglas de muestreo de la página web a la que se desee utilizar.

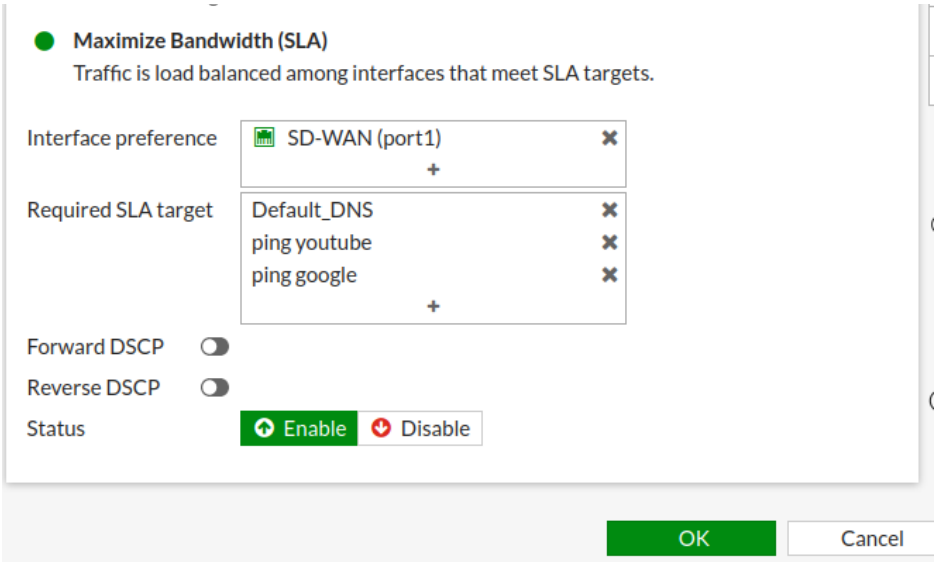


Figura 9.23 Configuración reglas de SLA en una interfaz SD-WAN

En la pestaña Performance SLA se puede observar los requerimientos que se han programado en la gráfica de tiempo vs Latencia, Perdida de Paquetes, Jitter de la red.

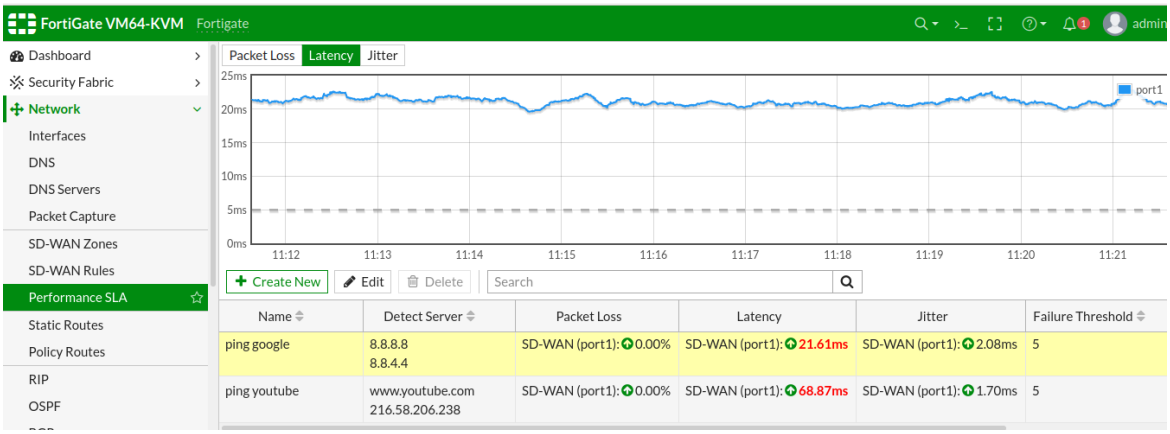


Figura 9.24 Performance SLA

Para poder colocar valores de eficiencia a cumplir es necesario ir a la pestaña Performance SLA, de ahí se procede a configurar el nombre, el protocolo que se utilizara, la dirección IP, y valores como Latency Theshold, Jitter Theshold, Packet Loss Theshold.

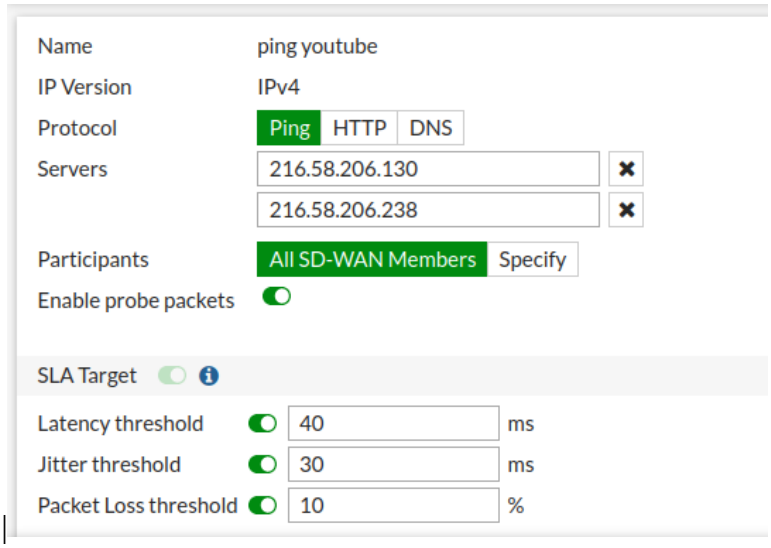


Figura 9.25 Configuración de SLA en una dirección IP

TRAFFIC SHAPING

Para poder utilizar Traffic Shaping en un puerto de FortiGate es necesario crear una política en la que se configurara las interfaces en las que se realizara, el tráfico de fuente desde la que llegara al firewall, así como nombre y estatus de la política de modelado de tráfico.

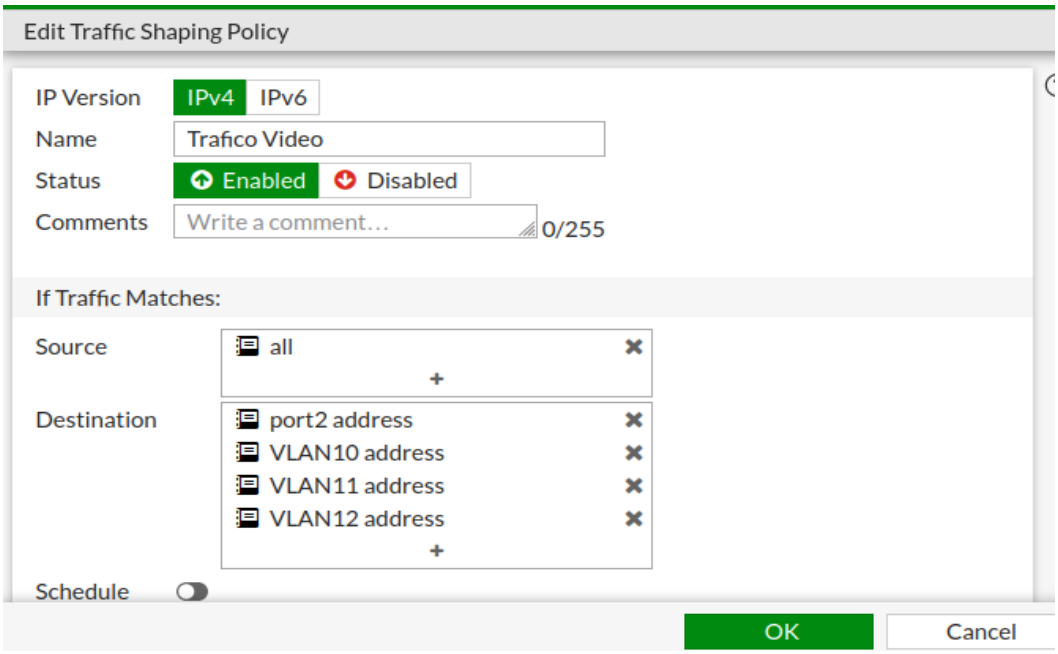


Figura 9.26 Políticas de Modelado de Tráfico

La política de Traffic Shaping de FortiGate permite poder aplicarla en una aplicación que ya ha sido colocada en FortiGate de fábrica, en la imagen se puede observar algunos ejemplos de políticas en uso.

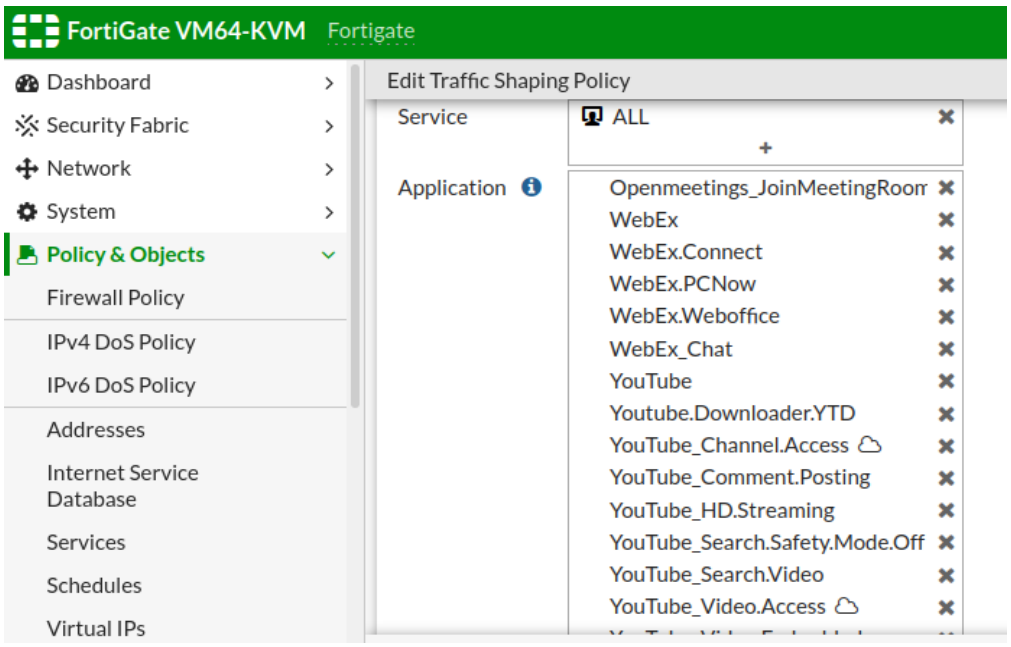


Figura 9.27 Aplicación de política configurada para un programa

También es necesario elegir la interfaz de salida en la que se realizará, así como la política previamente programada que se realizará.

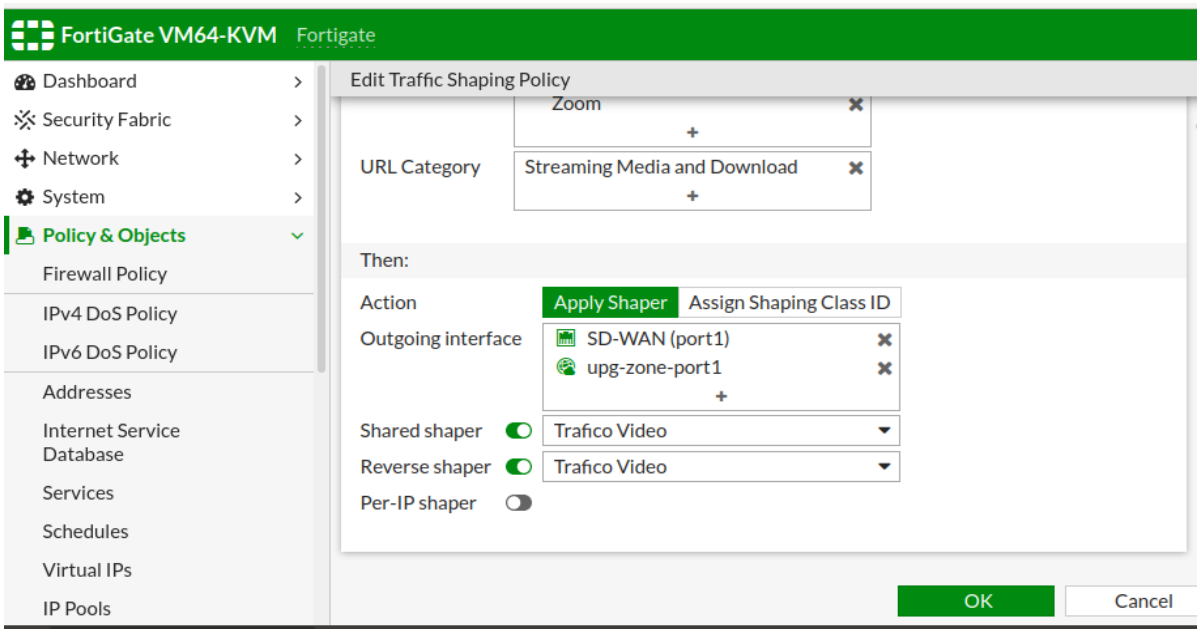


Figura 9.28 Elección de puertos para Traffic Shaping

SERVIDOR DNS

Es posible programar un servidor de DNS, el cual se puede utilizar con las direcciones DNS, en este caso se ha utilizado las de Google que son 8.8.8.8 o.8 8.8.4.4.

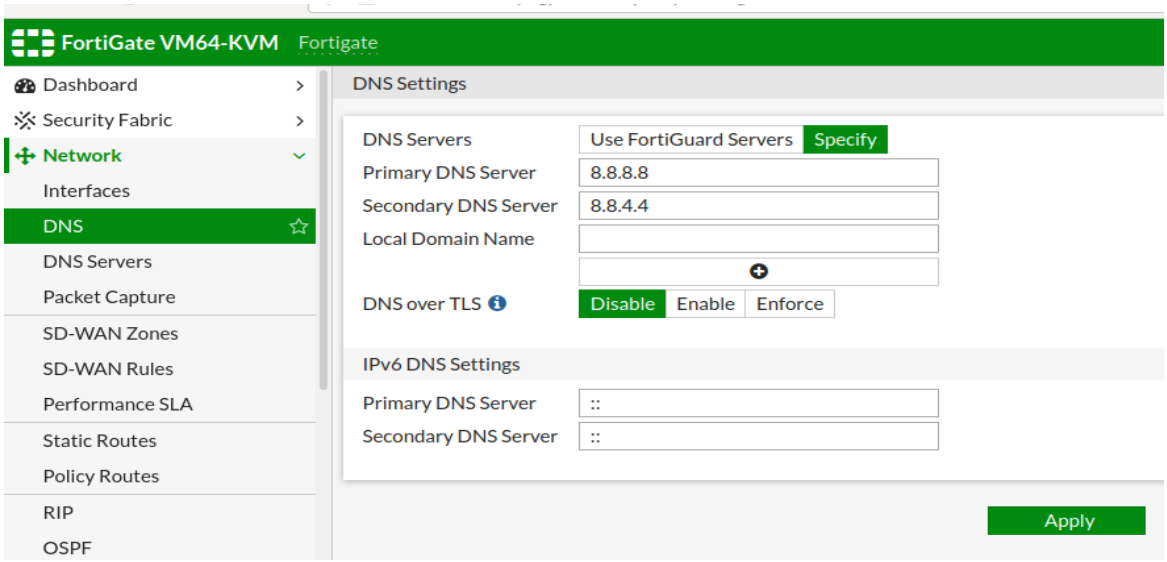


Figura 9.29 Opciones de DNS primario y Secundario

INTERFACES

FortiGate permite utilizar las interfaces mediante DHCP o de manera manual, en el caso de estudio se ha utilizado de manera manual. Colocando la dirección 10.10.1.0/28 para la LAN del controlador.

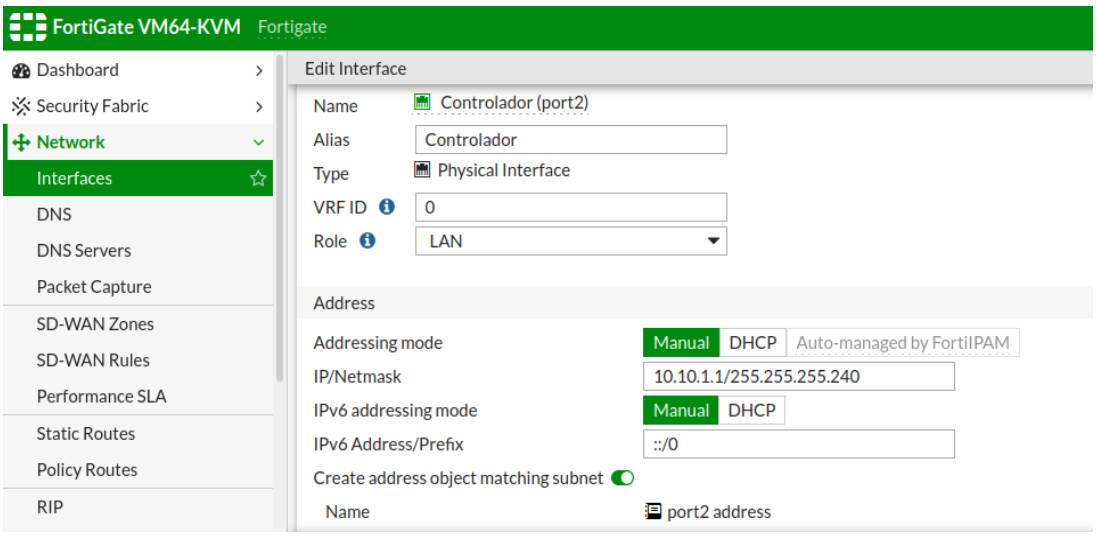


Figura 9.30 Interfaz de tipo LAN|

Se configura en esta opción los parámetros de acceso de los que se eligió HTTPS, PING, SSH, necesarios para un acceso remoto desde las interfaces permitidas y para probar conectividad, se ha utilizado también un servidor DHCP para los nuevos aparatos que se deseen conectar a esta subred.

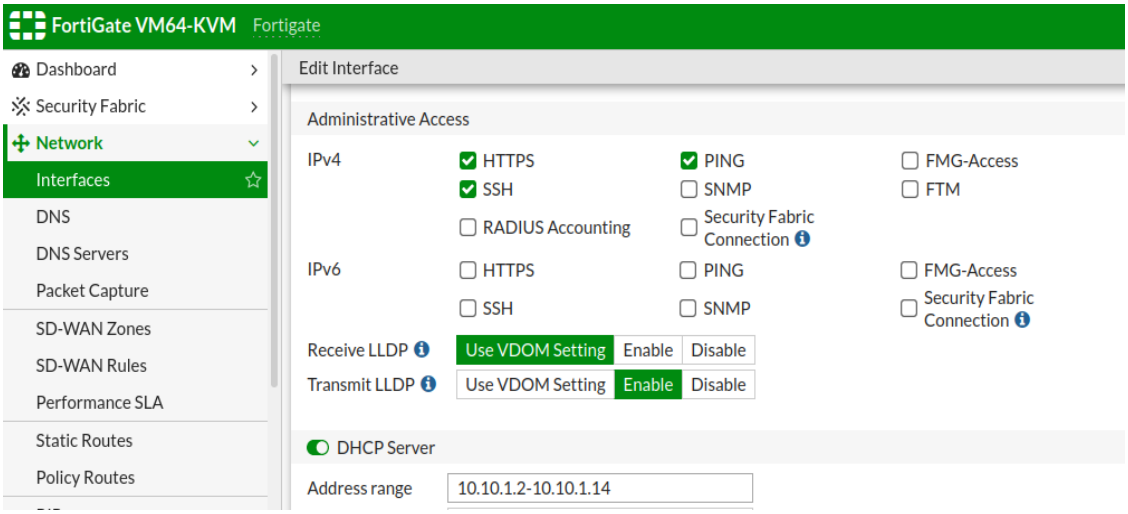


Figura 9.31 Configuración de acceso administrativo y servidor DHCP

Interfaz Software Switch

Para la interfaz que conecta a los OVS se eligió el modo de uso Software Switch la cual permite elegir varios puertos y separarlos para poder usarlos como puertos que pertenecen a la misma subred.

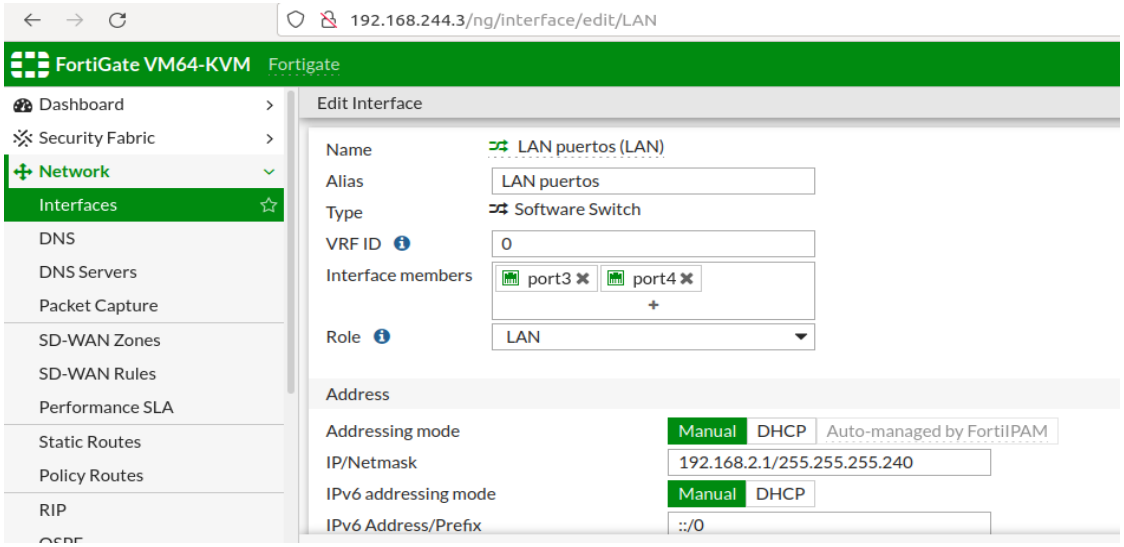


Figura 9.32 Interfaz de tipo Software Switch

VLAN

Para la creación de este tipo de configuración, en el panel de interfaces se elige que sea de tipo VLAN y se coloca un Nombre, un rol de LAN y una dirección IP.

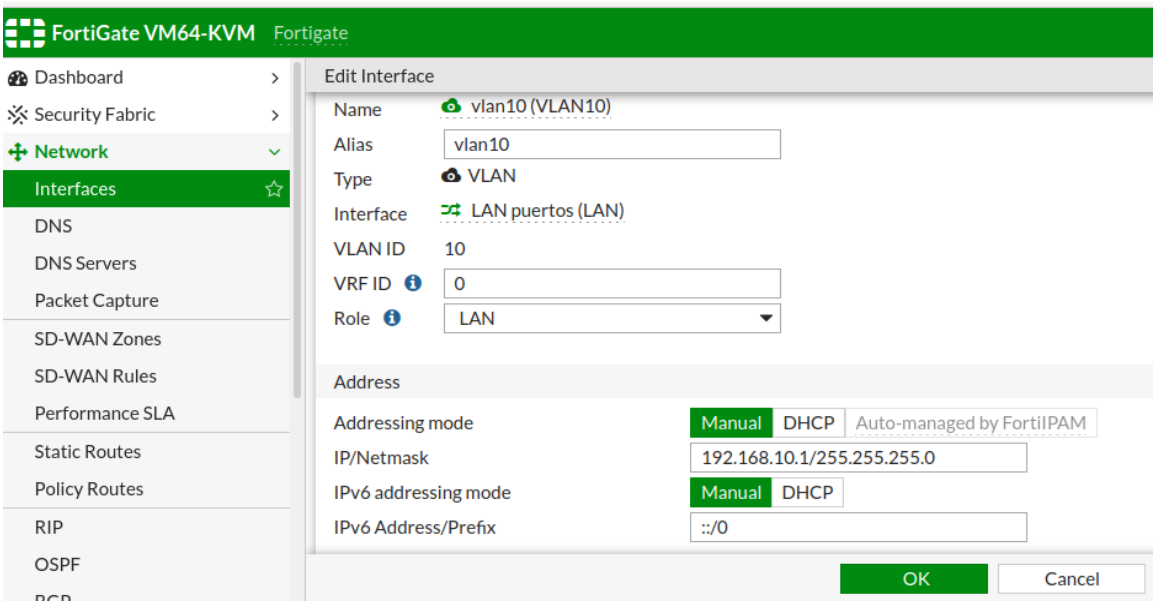


Figura 9.33 Configuración de una VLAN

Como con otras interfaces se elige que se permitirá en este caso solo se ha elegido ping para comprobar conexiones y DHCP para que los hosts obtengan rápidamente su dirección IP

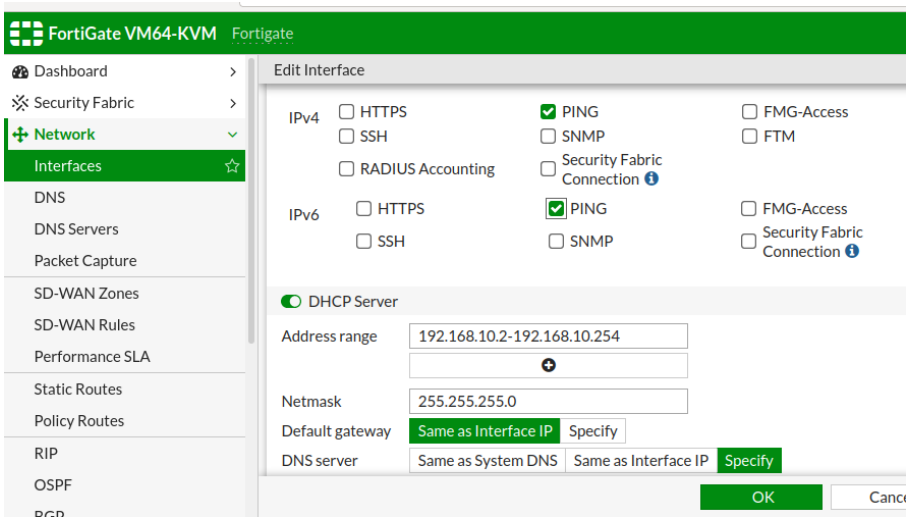


Figura 9.34 Creación de servidor DHCP en una VLAN

Se creo una política de Traffic Shaping la misma que da garantía acerca del ancho de banda calculado y el máximo ancho de banda posible colocándolo en alta prioridad.

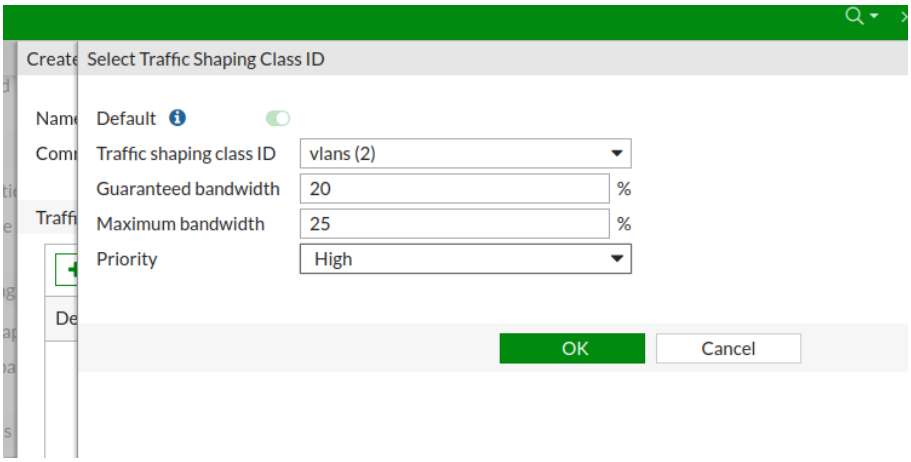


Figura 9.35 Política de Traffic Shaping para una VLAN

FIREWALL

Para poder permitir las conexiones a las diferentes redes este paso es muy importante puesto que FortiGate bloquea las conexiones que no se han configurado, siendo necesario configurar conexiones InterVLAN, del controlador hacia la red de OVS, desde las VLAN hacia internet, desde internet hacia las VLAN, del controlador hacia internet. Para crear una política de firewall se añade colocando un nombre, el puerto de entrada, el puerto de salida, las conexiones que se permitirán y el tipo de paquetes que se permitirá UDP, TCP, etc.

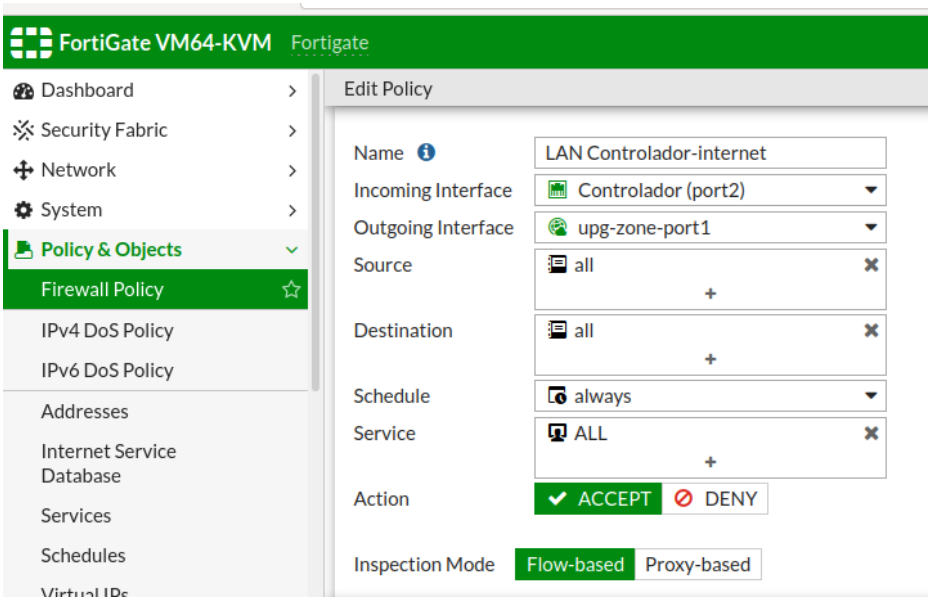


Figura 9.36 Creación de una política de Firewall

Es necesario colocar la traducción de nombres de NAT y colocar las políticas de seguridad, estas políticas se las puede editar, pero para este caso se han utilizado las políticas por defecto de FortiGate. Se pueden activar políticas de Antivirus, Filtrado de Páginas Web, Filtrado de DNS, Control de Aplicaciones, IPS, se puede activar o desactivar estos filtros para que sean utilizados.

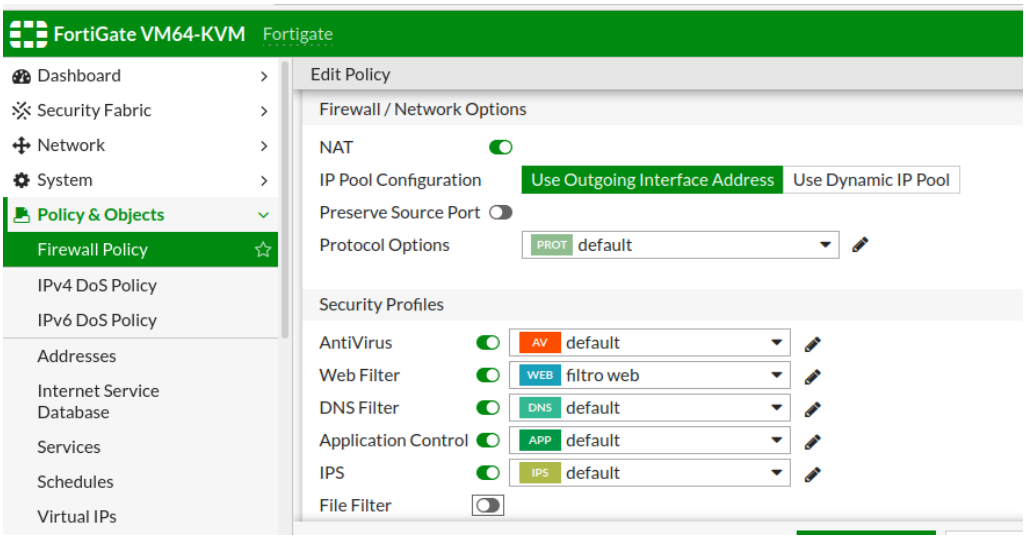


Figura 9.37 Características a implementar en una política Firewall

Para la creación de esas políticas se puede dar clic en el lápiz de manera que se podrá crear una nueva política en la que elegirá el nombre, un perfil, un grupo al que se aplica, opciones de aplicación.

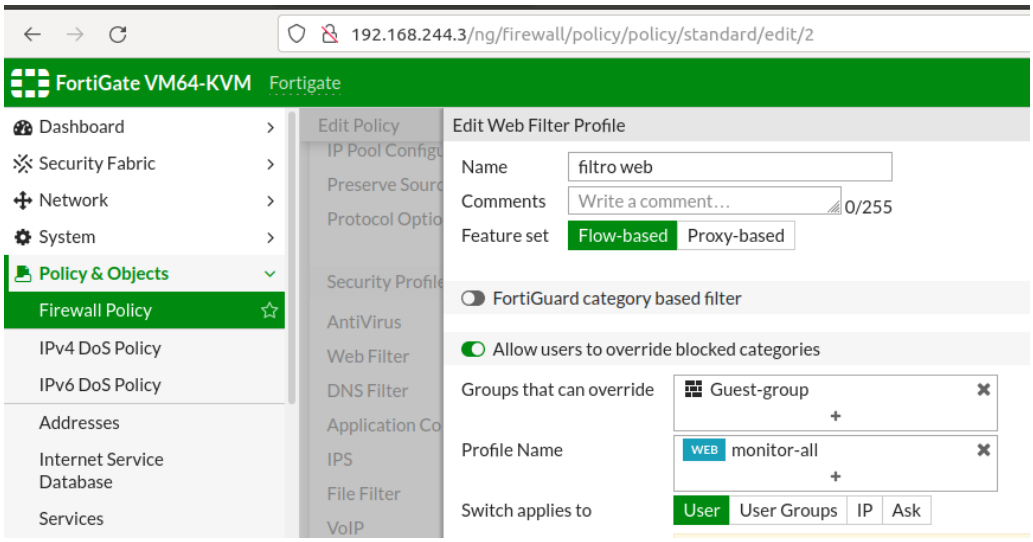


Figura 9.38 Perfil de una Filtro Web

Firewall Del Controlador

En el caso del controlador se ha elegido que solo se permita a la red de OVS un servicio de TCP, evitando otros tipos de conexiones hacia esa red.

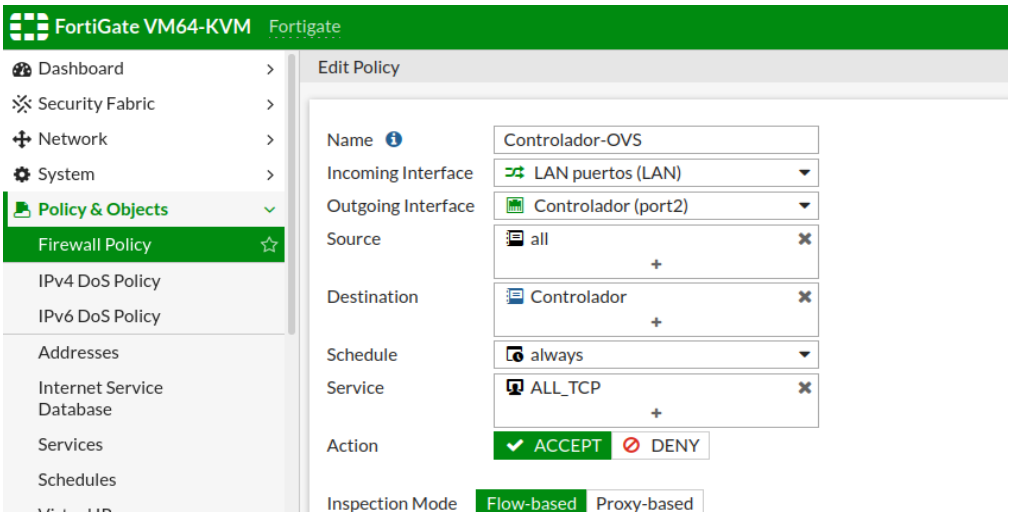


Figura 9.39 Firewall del Controlador Opendaylight

Firewall InterVLAN y NAT

Para crear políticas entre VLAN es necesario elegir la interfaz de entrada de una VLAN y la de salida de la que se desea conectar, elegir el tipo de servicio permitir su conexión a internet elegir el tipo de servicio. Esto se debe repetir para todas las políticas de Firewall que se permitirá.

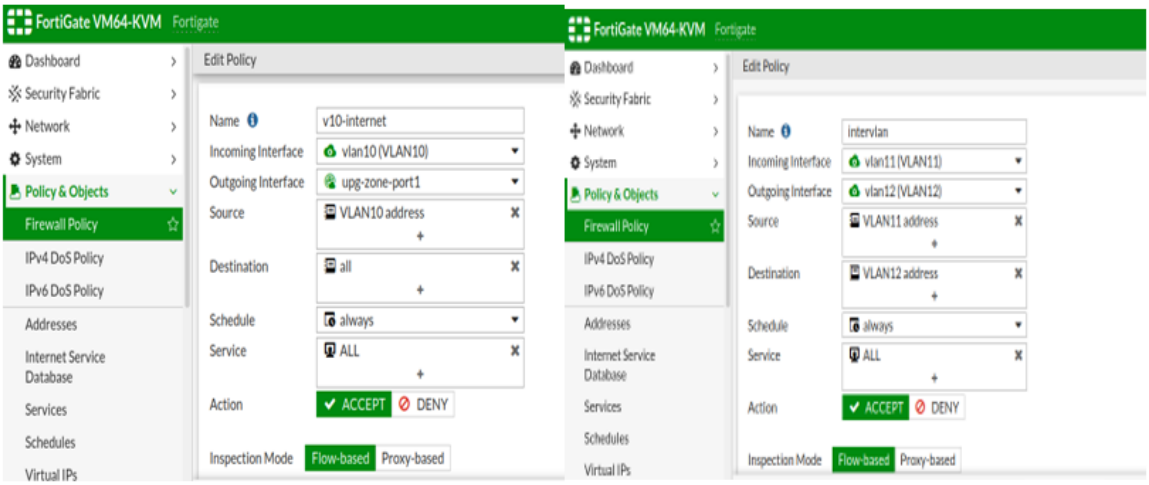


Figura 9.40 Políticas de acceso controlador ODL-Internet-Vlan

Para la conexión a internet se ha colocado una ruta estática predeterminada de IPv4 en la interfaz de ingreso conectada a la nube NAT perteneciente a la red creada en GNS3

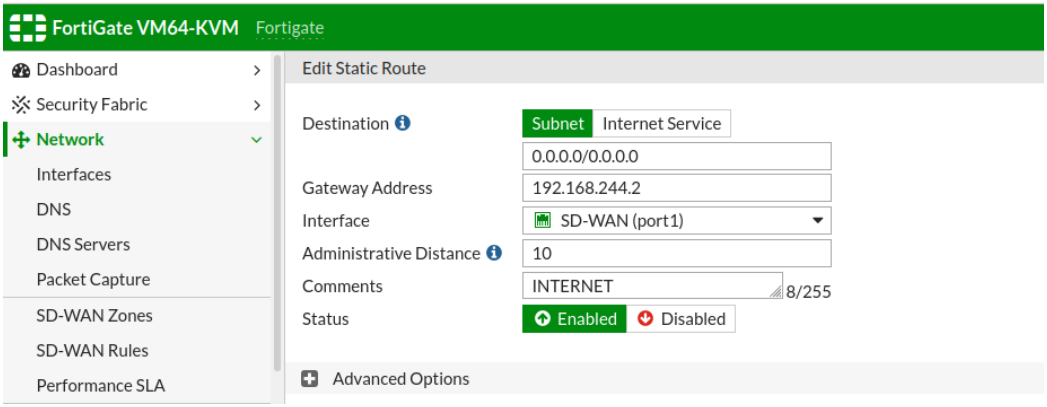


Figura 9.41 Configuración de una ruta estática.