UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE TELECOMUNICACIONES

INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR, PROYECTO DE INVESTIGACIÓN

TEMA:

"TESTBED DE DETECCIÓN Y MITIGACIÓN DE ATAQUES DE INUNDACIÓN DE DENEGACIÓN DE SERVICIO (DoS) EN REDES SDN/MPLS DE SERVICIOS DIFERENCIADOS"

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

Línea de investigación: Desarrollo, aplicación de software y cibersecurity (seguridad cibernética)

AUTOR:

Juma Guamán Keneth Santiago

DIRECTOR:

Ing. Domínguez Limaico Hernán Mauricio, MSc

Ibarra, 2025

UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003847231		
APELLIDOS Y NOMBRES:	Juma Guamán Keneth Santiago		
DIRECCIÓN:	Ibarra, Barrio San José de Chorlavi y vía Chorlavi casa 2-43		
EMAIL:	ksjumag@utn.edu.ec		
TELÉFONO FIJO:		**TELF. MOVIL**	0992875929

DATOS DE LA OBRA	
TÍTULO:	Testbed de Detección y Mitigación de Ataques de Inundación de Denegación de Servicio (DoS) en Redes SDN/MPLS de Servicios Diferenciados
AUTOR (ES):	Juma Guamán Keneth Santiago
FECHA:	23/01/2025
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
CARRERA/PROGRAMA:	■ **GRADO** □ **POSGRADO**
TITULO POR EL QUE OPTA:	Ingeniero en Telecomunicaciones
DIRECTOR:	Msc. Hernán Mauricio Domínguez Limaico
	Msc. Fabián Geovanny Cuzme Rodríguez

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Juma Guamán Keneth Santiago, con cédula de identidad Nro. 1003847231 en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 23 días del mes de enero de 2025

EL AUTOR:

.....

Keneth Santiago Juma Guamán

CONSTANCIAS

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

Ibarra, a los 23 días, del mes de enero de 2025

EL AUTOR:

.....
Keneth Santiago Juma Guamán

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 23 de enero de 2025

Ing. Hernán Mauricio Domínguez Limaico Msc.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:
Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

.....

Ing. Hernán Mauricio Domínguez Limaico. MSc
C.C.: 1002379301

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular "Testbed de detección y mitigación de ataques de inundación de denegación de servicio (DOS) en redes SDN/MPLS de servicios diferenciados" elaborado por Juma Guamán Keneth Santiago, previo a la obtención del título de Ingeniero en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

.....

Ing. Hernán Mauricio Domínguez Limaico. MSc
C.C.: 1002379301

.....

Ing. Fabián Geovanny Cuzme Rodríguez MSc
C.C.: 1311527012

DEDICATORIA

El presente trabajo va dedicado con todo el amor que tengo a las personas que me han apoyado en todo el trayecto de mi vida; en primera instancia a mi madre María Elizabeth Guamán Guerra, quien ha sido un modelo a seguir para mí yaqué gracias a su perseverancia, apoyo y amor incondicional en todos los proyectos y etapas de mi vida, me han dotado de la capacidad de superar cualquier desafío que se me presente y a la vez formar mi carácter para siempre ser una persona humilde de valores y que ayude a los demás cuando está en mis capacidades.

De igual manera, quiero dedicar esta investigación a mi padre Miguel Ramiro Juma Potosí quien ha sido un padre amoroso, preocupado del bienestar de su familia y que siempre ha estado orgulloso de todo lo que he hago y haré en mí vida. Además, su motivación de alcanzar mis metas con trabajo duro en base a un buen desempeño y esfuerzo sin nunca rendirse ante cualquier adversidad han marcado el perfil humano, académico y profesional que tengo que seguir; de modo que esto no sería posible si no fuese por él.

A mi hermana Nicole Elizabeth Juma Guamán, quien ha sido la persona que más ha creído en mí y que con su cariño sin límites en conjunto con sus ocurrencias siempre han dejado una huella en mi vida del amor que una hermana y en ocasiones se sacrifica por ayudarte a alcanzar tus metas. También, quiero mencionar a mi hermano Cristian Daniel Castro Guamán y toda su familia, los cuales me han motivado para nunca decaer en cualquier proceso de mi vida y siempre ser mejor cada vez.

Finalmente, quiero dedicar la culminación de este trabajo a mí abuelita Blanca Potosí y a mis ángeles que están en el cielo y que desde allí me brindan su bendición en todos los aspectos de mí vida, a mis abuelitos: Miguel Juma, Tadeo Guamán y María Guerra.

Juma Guamán Keneth Santiago

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por darme la vida, salud y la oportunidad de culminar este proyecto de la mejor forma, que ha sido el trabajo en conjunto de varias personas que han aportado su tiempo y dedicación.

Agradezco de todo corazón a mis padres, mis hermanos, a mis abuelitos y a mi familia por el apoyo incondicional en cada una de las fases que ha implicado una etapa universitaria, sin su amor y enseñanzas no habría podido superar los retos que me presentó este camino.

De igual manera, agradezco de a mis amigos con los cuales he compartido un sin número de experiencias en la universidad, como fuera de ella. En especial quiero agradecer a mis amigas Anita, Kathy y Anahí, que desde que las conocí siempre se han preocupado por mí bienestar y hemos compartido varias experiencias muy especiales en mi vida. También, quiero agradecer a mis amigos Ariel, Ismael, Marco y Karlita por formar parte de este proceso universitario desde sus inicios y poder contar siempre con su amistad para cualquier ámbito que se me presente.

De manera especial quiero agradecer a mi director el Ing. Mauricio Domínguez, quien ha sido un pilar fundamental para este trabajo de investigación, debido a la gran apertura de tiempo y dedicación de su parte que brindo desde los inicios del desarrollo del proyecto, que además su experiencia, conocimiento y consejos me han dejado grandes enseñanzas en el ámbito académico y profesional.

Finalmente, quiero agradecer de igual manera a mi asesor el Ing. Fabián Cuzme por su guía durante el desarrollo de toda la investigación brindándome sus consejos y sugerencias para la culminación exitosa de la tesis.

Juma Guamán Keneth Santiago

RESUMEN EJECUTIVO

Este trabajo presenta un análisis y aplicación de técnicas de mitigación de ataques de denegación de servicio (DoS) en redes definidas por software (SDN) mediante la operación en conjunto de un sistema de detección y prevención de intrusiones (IDS/IPS) de código abierto y el controlador SDN Ryu. La flexibilidad y capacidad de gestion centralizada que ofrecen las redes SDN, enfrentan desafíos de seguridad significativos, especialmente ante ataques DoS de tipo inundación, que afectan la operatividad de la red al saturar los recursos de red como la capacidad de procesamiento del controlador en conjunto con las reglas de flujo de los conmutadores, que al soportar altos volumenes de tráfico en los enlaces saturan su ancho de banda, generando en casos críticos, la interrupción total de los servicios que funcionan sobre la red. Esta investigación aplicada combina análisis teórico, escenarios controlados de ataques y la mitigación mediante el despliegue de Snort como IDS/IPS en conjunto con las capacidades de programación del controlador, de modo que los resultados evidencian la efectividad de la mitigación ante este tipo de ataques, validando la integración del IDS/IPS como una solución de seguridad factible y centralizada para este tipo de redes

Palabras clave: SDN, MPLS, QoS, IDS/IPS, controlador SDN, conmutador SDN, ataques DoS, saturación de ancho de banda, mitigación de ataques, reglas de flujo.

ABSTRACT

This work presents an analysis and application of techniques to mitigate denial-of-service (DoS) attacks in software-defined networks (SDN) through the combined operation of an open-source intrusion detection and prevention system (IDS/IPS) and the Ryu SDN controller. The flexibility and centralized management offered by SDN face significant security challenges, especially against flood-type DoS attacks. These attacks impact network operations by overloading network resources, such as the controller's processing capacity and the flow rules of switches. High traffic volumes on links can saturate their bandwidth, causing critical cases where all network services are completely interrupted. This applied research combines theoretical analysis, controlled attack scenarios, and mitigation through the deployment of Snort as an IDS/IPS together with the programming capabilities of the controller. The results show the effectiveness of the mitigation against these types of attacks, validating the integration of the IDS/IPS as a feasible and centralized security solution for this type of network.

Keywords: SDN, MPLS, QoS, IDS/IPS, SDN controller, SDN switch, DoS attacks, bandwidth saturation, attack mitigation, flow rules.

LISTA DE SIGLAS

SDN. Redes Definidas por Software (Software Defined Networking).

MPLS. Conmutación de Etiquetas Multiprotocolo (Multiprotocol Label Switching).

IDS. Sistema de Detección de Intrusiones (Intrusion Detection System).

IPS. Sistema de Prevención de Intrusiones (Intrusion Prevention System).

DoS. Denegación de Servicio (Denial of Service).

DDoS. Denegación de Servicio Distribuida (Distributed Denial of Service).

QoS. Calidad de Servicio (Quality of Service).

API. Interfaz de Programación de Aplicaciones (Application Programming Interface).

RFC. Solicitud de Comentarios (Request for Comments).

TTL. Tiempo de Vida (Time To Live).

TCAM. Memoria Ternaria con Dirección por Contenido (Ternary Content Addressable Memory).

OFPT. Tipo de Paquete OpenFlow (OpenFlow Packet Type).

ONF. Fundación de Redes Abiertas (Open Networking Foundation).

IRTF. Grupo de Trabajo de Investigación de Internet (Internet Research Task Force).

ACL. Lista de Control de Acceso (Access Control List).

SPAN. Analizador de Puertos Conmutados (Switched Port Analyzer).

TAP. Punto de Acceso de Prueba (Test Access Point).

CIA. Triada de Confidencialidad, Integridad y Disponibilidad.

SSL. Capa de Conexión Segura (Secure Sockets Layer).

TLS. Seguridad de la Capa de Transporte (Transport Layer Security).

STRIDE. Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege.

IPv4. Protocolo de Internet versión 4 (Internet Protocol version 4).

OFPT_FLOW_MOD. Modificación de Flujo OpenFlow.

OFPT_PACKET_IN. Paquete de Entrada OpenFlow**.**

OFPT_PACKET_OUT. Paquete de Salida OpenFlow**.**

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO I: Antecedentes

1.1 Problema de investigación

La evolución de la interconectividad en las redes de comunicación ha brindado mayor flexibilidad en la implementación de arquitecturas basadas en software, lo que permite una escalabilidad futura con la adopción de nuevos mecanismos de red (Cuenca Pérez & Flores Marín, 2015). Sin embargo, el enfoque de desarrollo de nuevas formas de arquitecturas de red se basa en un principio a las limitaciones de las redes tradicionales que requerían una infraestructura de hardware estático que permita su funcionamiento mediante equipos de conmutación, enrutamiento y middleboxes de forma centralizada. Esto ha llevado a que el tráfico de datos en este tipo de redes sea administrado y controlado de forma única mediante reglas y políticas de control alojadas en el hardware de los dispositivos de red, que en muchos casos son definidos por los mismos fabricantes, lo que ha generado un deterioro en el rendimiento y eficiencia de estas redes debido al alto nivel de tráfico de datos que pueden llegar a manejar, especialmente en la distribución de los servicios que se alojan en estas infraestructuras (Mejía, 2018).

A partir de eso, se origina una solución denominada Redes Definidas por Software (SDN), las cuales otorgan una serie de características que permiten satisfacer las necesidades de capacidad volumétrica, altas velocidades de transmisión de datos y facilidad de implementación que carecían las infraestructuras de red tradicionales. Esto se debe principalmente a la arquitectura innovadora de las SDN, que separa el plano de datos encargado de la conmutación de los paquetes con el plano de control, el cual realiza las funciones de gestión de la red (Bujedo et al., 2022). Lo que ha originado que toda la inteligencia del sistema de intercomunicación sea centralizada en un controlador que posee la capacidad de configuración, control y administración de la red con el propósito

de identificar y controlar todo el flujo de tráfico que se transfiere mediante este tipo infraestructuras (Cuenca Pérez & Flores Marín, 2015).

Las capacidades de operabilidad de las SDN han permitido que puedan funcionar en conjunto con tecnologías que mejoran la eficiencia en la conmutación de paquetes IP, como lo es MPLS, permitiendo el despliegue de redes híbridas que aprovechan las ventajas y características de ambos enfoques, brindado la posibilidad de optimizar el enrutamiento de paquetes con una gestión eficiente del tráfico de datos y a la vez otorgar un control centralizado que facilita la administración, programabilidad e implementación de políticas dinámicas de interconexión (Fernández & Ulloa, 2016). En base a estos conceptos surgen las redes híbridas SDN/MPLS, las cuales poseen como objetivo central obtener un sistema de comunicación eficiente, adaptable y preparado para satisfacer las demandas cambiantes de las aplicaciones y servicios que manejan un alto número de paquetes de datos (Lozada, 2022).

No obstante, las redes SDN/MPLS al ser un avance tecnológico en el área de los sistemas de comunicación basados en software están expuestas a varias fuentes de riesgo de seguridad, debido a su perspectiva innovadora en el diseño de su arquitectura que centraliza todas sus funciones en el controlador SDN, de modo que las vulnerabilidades explotadas pueden llegar a generar repercusiones que compromete a la integridad, disponibilidad y eficiencia de todo el sistema de intercomunicación (Yanko et al., 2016). Las redes híbridas SDN/MPLS enfrentan diversas amenazas en los tres planos que conforman su arquitectura como lo son el plano de datos, el plano de control y el plano de aplicación (Shu et al., 2016) Sin embargo, el plano de control es el principal objetivo de las intrusiones de seguridad ya que al comprometer su funcionamiento genera graves repercusiones en el rendimiento de la red y afecta la capacidad de operación de todos los servicios que dependen de esta infraestructura (Ahmad et al., 2015).

La existencia de vulnerabilidades en el plano de control debido a la centralización de sus funciones en uno solo elemento como lo es el controlador, ha generado que los ataques de denegación de servicio (DoS) de tipo inundación enfoquen sus esfuerzos en explotar los recursos a nivel de hardware y las capacidades de administración a nivel de software de este elemento, en un intento de comprometer su operabilidad en toda la red; siendo este tipo de afecciones los principales riegos de explotación de vulnerabilidades que las SDN/MPLS de distribución de servicios diferenciados poseen, en virtud del gran nivel de impacto que este tipo de acometidas pueden llegar a generar (Dargahi et al., 2017). Los DoS de tipo inundación son los ataques que poseen un elevado índice de riesgo en comprometer la disponibilidad e integridad del plano de control y esto se debe a que son intrusiones que agotan el ancho de banda y los recursos que el controlador SDN y se debe principalmente al masivo número de solicitudes en tiempos cortos generados desde una o múltiples fuentes ya que poseen una rápida expansión y fácil desarrollo (Ahmad et al., 2015).

Las intrusiones de seguridad de este tipo representan una amenaza directa contra dos de los principales pilares de la seguridad en redes definidas por software que es la integridad y disponibilidad de la red, generando una notable disminución en la calidad de los servicios a causa de este tipo de afectaciones de seguridad. Estas complicaciones de seguridad de las redes híbridas SDN/MPLS pueden detectarse y mitigarse mediante el uso de software libre como lo son los Sistemas de Detección de Intrusiones (IDS) y los Sistemas de Prevención de Intrusiones (IPS), en base al flujo de información que este tipo de ataques pueden generar en sus primeras fases al llegar a los conmutadores o a su vez en la asignación de reglas de flujo que genera el controlador (Shirokova, 2019), es posible adaptarlos de tal manera que estos ataques sean mitigados antes de comprometer al plano de control y las funcionalidades de la red SDN/MPLS con la consigan de no

comprometer la disponibilidad e integridad del controlador en conjunto con los servicios que se distribuyen.

1.2 Objetivos

1.3.1 Objetivo General

Implementar mecanismos de seguridad en redes híbridas SDN/MPLS mediante el uso de herramientas de software libre que permitan la prevención y mitigación de los ataques de denegación de servicio (DoS) de tipo inundación.

1.3.2 Objetivos Específicos

- Establecer el estado del arte sobre los ataques de denegación de servicio (DoS) de tipo inundación y como las vulnerabilidades de las redes híbridas SDN/MPLS comprometen con la disponibilidad en la distribución de servicios diferenciados.
- Implementar una red SDN/MPLS de distribución de servicios diferenciados mediante un software libre de diseño de redes que permita el análisis de su funcionamiento ante ataques de denegación de servicio (DoS) de tipo inundación.
- Desplegar una solución de seguridad en redes híbridas SDN/MPLS basada en un IDS/IPS de software libre que permita el control flujos de tráfico de red en la detección y mitigación de ataques de denegación de servicio de tipo inundación.
- Evaluar los resultados del despliegue del IDS/IPS en la red híbrida SDN/MPLS en comparación con la red de características similares que no posee mecanismos de seguridad que resguarden su disponibilidad e integridad.

1.3 Alcance

El enfoque principal del proyecto es mejorar la seguridad en redes híbridas SDN/MPLS, cumpliendo con los requisitos necesarios para fortalecer los componentes de disponibilidad e integridad, establecidos por la triada CIA en redes definidas por software. Por consiguiente, la documentación del trabajo de integración curricular se centra en la investigación aplicada, ya que requiere en primera instancia recopilar la información bibliográfica necesaria referente a redes híbridas SDN/MPLS, vulnerabilidades, ataques de denegación de servicio y mecanismos de contramedida contra este tipo de afectaciones a la red. Esto es realizado con el objetivo de adquirir los conocimientos teóricos necesarios para posteriormente implementar una solución práctica, como lo son los sistemas de detección y prevención de intrusiones (IDS/IPS) basados en el flujo del tráfico de red, que permitan asegurar el funcionamiento de la red híbrida SDN en conjunto con los servicios alojados en la infraestructura de red.

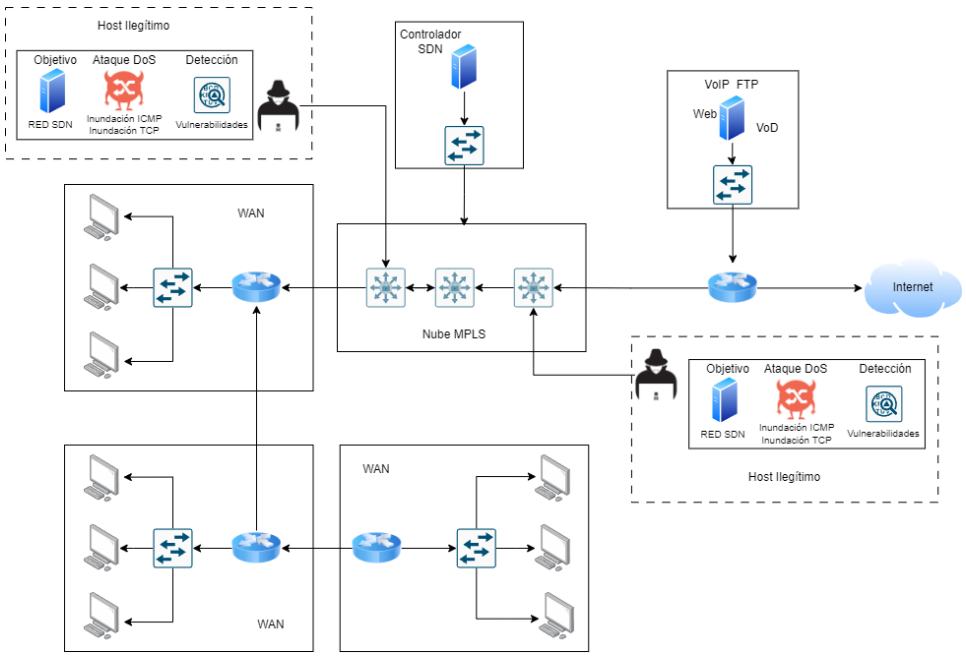
El enfoque metodológico utilizado para abordar todas las fases relacionadas con la seguridad en el desarrollo del testbed se basa en las recomendaciones de la metodología STRIDE. Este enfoque se compone de seis etapas clave correspondientes a la identificación de activos y componentes del sistema, la identificación de amenazas STRIDE, análisis de riesgos, diseño de contramedidas, implementación y pruebas, así como el monitoreo y mantenimiento continuo del mecanismo de seguridad implementado; por consecuente cada una de estas fases desempeña un papel fundamental en el proceso de asegurar y proteger la red híbrida SDN/MPLS, al abordar aspectos clave como la identificación de vulnerabilidades, la evaluación de riesgos y la implementación de medidas de seguridad efectivas, así pues tomando como base a la metodología previamente mencionada las fases que conforman el proyecto de investigación son las siguientes:

La primera sección del desarrollo del proyecto corresponde al análisis de la fundamentación teórica mediante la recopilación en fuentes bibliográficas que permitan identificar y contextualizar la problemática respecto a las vulnerabilidades que poseen las redes híbridas SDN/MPLS y como los ataques de denegación de servicio (DoS) de tipo inundación afectan a su funcionamiento. Esto se lleva a cabo con el propósito de establecer una sólida base de conocimientos teóricos que permita identificar una propuesta de seguridad adecuada de despliegue de un mecanismo de seguridad, como lo son los IDS/IPS, debido a que su funcionamiento tiene como principal finalidad salvaguardar la disponibilidad e integridad de la red y los servicios alojados, evitando afectaciones a la operabilidad de la red ante posibles ataques de denegación de servicio (DoS) de tipo inundación.

La segunda fase corresponde al uso de una plataforma de diseño de redes de código abierto que permita el despliegue de una red híbrida SDN/MPLS de distribución de servicios diferenciados en base a un controlador SDN que centraliza sus funciones y permite la conectividad de todas las redes, con la finalidad de establecer el primer escenario de pruebas. Una vez que la red esté operativa, el primer entorno de pruebas del testbed puede comenzar con la explotación de las vulnerabilidades previamente identificadas en la fase de fundamentación teórica. En el escenario (ver Figura 1), un grupo de hosts maliciosos utiliza herramientas de software y exploits con el objetivo de afectar a los servidores que brindan los servicios mediante ataques de denegación de servicio (DoS) de tipo inundación. El enfoque principal de los ataques es impactar en la disponibilidad e integridad del controlador SDN y las aplicaciones DiffServ que se encuentran alojadas en la infraestructura híbrida de intercomunicación SDN.

Figura 1

Topología de red correspondiente al primer escenario del testbed.

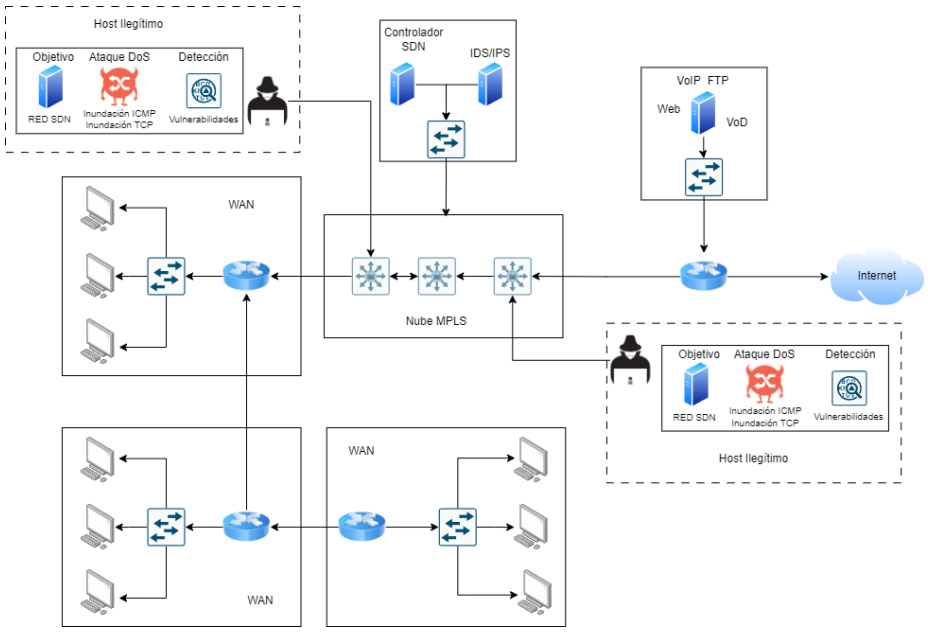


La tercera fase del proyecto analiza las repercusiones de los ataques de inundación DoS en la red SDN/MPLS del primer escenario del testbed, en base a las tres primeras etapas de la metodología STRIDE. Las primeras secciones se desarrollan con el propósito de establecer los criterios necesarios para fundamentar el diseño de contramedidas en la cuarta fase, que está enfocada al diseño de estrategias de mitigación ante las repercusiones generadas por los ataques DoS a la red híbrida SDN, permitiendo así respaldar la propuesta de implementación del mecanismo de seguridad IDS/IPS basado en flujos, que tiene como objetivo principal garantizar la integridad y disponibilidad del controlador SDN. Además, la solución de seguridad propuesta busca minimizar el impacto negativo en la latencia y pérdida de paquetes de los servicios diferenciados alojados en la red ante este tipo de ataques. En la quinta fase de la metodología STRIDE, enfocada en implementación y pruebas, se enfoca en el despliegue de las medidas de seguridad diseñadas específicamente como contramedidas (ver Figura 2). Posteriormente, se realizan pruebas que permitan evaluar su efectividad frente a los ataques de denegación

de servicio (DoS) de tipo inundación con el objetivo de asegurar la protección del controlador SDN y los servicios alojados en la red, evitando que se vean afectados por estos ataques.

Figura 2

Topología de red correspondiente al segundo escenario del testbed.



Finalmente, la última etapa de la metodología STRIDE se centra en el monitoreo y mantenimiento, que comienza con la evaluación de los dos escenarios del testbed. El objetivo es determinar cómo el mecanismo de seguridad implementado en la red SDN/MPLS cumple su función de mitigar los efectos de los ataques de inundación DoS en el controlador SDN y los servicios de red. Además, se realiza un análisis comparativo con la finalidad de identificar las características del mecanismo de seguridad implementado, que garantiza la disponibilidad e integridad de los elementos de la red SDN y los servicios distribuidos, en comparación con una red similar que carece de las medidas de seguridad necesarias para hacer frente a los ataques de denegación de servicio (DoS). Finalmente, se establecen mecanismos de monitoreo y mantenimiento continuo en conjunto con medidas de seguridad adicionales para asegurar que la línea de defensa de la red SDN/MPLS implementada se mantengan actualizada y efectiva.

1.4 Justificación

El proyecto de investigación surge a partir de la necesidad de establecer criterios y mecanismos de seguridad en las redes híbridas SDN/MPLS, puesto que las redes definidas por software son una tecnología en constante evolución y han sido ampliamente aceptadas en la actualidad, debido a las facilidades que ofrecen en base a su arquitectura centralizada, mejorando en gran medida la velocidad y flexibilidad en la gestión de la infraestructura de red de forma conjunta con mejores prestaciones de conmutación de paquetes de datos, permitiendo la modificación de la red y las reglas de flujo en tiempo real para hacer frente a todo tipo de demanda que requieran los proveedores de servicios respecto a introducir nuevas aplicaciones de forma inmediata. Por consiguiente, el estudio de los efectos que pueden generar los ataques de denegación de servicio (DoS) en SDN/MPLS permite generar información bibliográfica de gran relevancia que permita exponer como la implementación de mecanismos de seguridad permiten que este tipo soluciones tecnológica en el campo de las redes de comunicación cumpla con los requerimientos necesarios de seguridad respecto a las infraestructuras de red tradicionales (Bone et al., 2021).

Uno de los Objetivos de Desarrollo Sostenible propuesto por la Organización de las Naciones Unidas consiste en "Establecer infraestructuras resilientes, fomentar la industrialización y promover la innovación". Este objetivo cobra gran importancia en el ámbito de las redes híbridas SDN/MPLS, ya que conforman un entorno donde los recursos de red son centralizados y controlados, de modo que los riesgos y amenazas a la seguridad se orientan de forma directa a los elementos que componen la operabilidad de la infraestructura de red (Dargahi et al., 2017). Por lo tanto, es crucial desarrollar redes resilientes capaces de resistir y recuperarse de eventos adversos, como lo son los ataques cibernéticos, los cuales pueden tener consecuencias devastadoras, incluyendo la

interrupción de servicios críticos, la pérdida o robo de datos sensibles y los impactos hacia la integridad y disponibilidad de la red.

Al tomar en cuenta todas estas consideraciones, el trabajo de investigación tiene como finalidad cumplir los objetivos establecidos en los pilares 1 y 19 de la Agenda de Transformación Digital del Ecuador 2022-2025 correspondiente a promover el despliegue de infraestructuras de redes de banda ancha, fomentar la innovación del sector de las telecomunicaciones y mejorar la gestión de la seguridad de la información a través de la formulación de programas y proyectos (MINTEL, 2022). De esta manera la implementación de mecanismos de seguridad como lo son los IDS/IPS cumplen con el objetivo de asegurar la disponibilidad e integridad de las redes híbridas SDN/MPLS ante posibles intrusiones como lo son los ataques de denegación de servicio (DoS) de tipo inundación, además este tipo de despliegues cumplen con el artículo 44 de la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP) que implica en garantizar la seguridad de las infraestructura de comunicaciones y los datos que se transfieren en estos sistemas de intercomunicación definidos por software.

CAPÍTULO II: Fundamentación Teórica

El presente capítulo propone un análisis bibliográfico referente a estudios previos relacionados con el tema de investigación, de modo que se proporcionará una descripción sobre las vulnerabilidades existentes que poseen las redes híbridas SDN/MPLS que hace uso de los conceptos teóricos correspondientes a Redes Definidas por Software (SDN) en conjunto con protocolos de conmutación de etiquetas como lo es MPLS, con el objetivo de identificar las principales afectaciones que los ataques de denegación DoS de tipo inundación pueden causar en su funcionamiento y como mecanismos de seguridad como lo son los IDS/IPS pueden minimizar sus consecuencias.

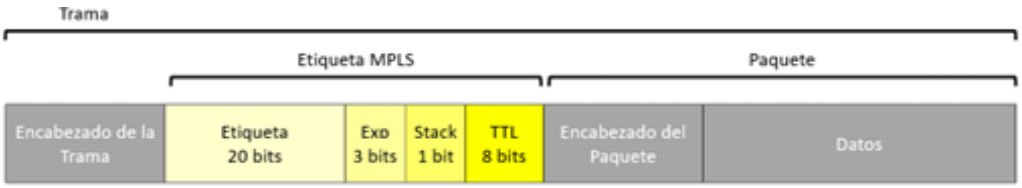
2.1 MultiProtocol Label Switching

Conforme a lo establecido por la (IETF, 2001) en el RFC 3031 referente al Conmutación de Etiquetas Multiprotocolo identificada por las siglas MPLS se caracteriza principalmente como una tecnología que posee las capacidades de potenciar el desempeño de las redes IP mediante el uso de etiquetas de conmutación. Esta tecnología es situada en la capa 2.5 tomando como referencia al modelo ISO/OSI[1], lo que indica que se encuentra ubicada en la sección intermedia del encabezado del paquete IP y el encabezado de la trama de la capa enlace de datos (IETF, 2001) como puede ser identificado en la Figura 3.

Debido a que MPLS se encuentra entre estas dos capas es dotado con el nombre de “Multi Protocol” ya que su posicionamiento le permite obtener como ventaja la capacidad de usar sin restricciones las características de los protocolos de capas adyacentes (Lores, 2011).

[1]ISO/OSI: International Standard Organization / Open System Interconnection.

Figura 3
Posicionamiento y campos del encabezado MPLS



Nota. Obtenido de (Okokpujie et al., 2018).

2.1.1 Encabezado MPLS

El encabezado MPLS establecido en el RFC 3032 con el nombre de pila de etiquetas define un tamaño total de cabecera de 4 octetos correspondientes a 32 bits (IETF, 2001), La Figura 3 proporciona una visualización de cómo el encabezado MPLS establece 4 elementos que conforman la pila de etiquetas. Según lo establecido por el RFC 3032, cada uno de estos elementos corresponde a campos exclusivos de un número específico de bits, como se detalla en la Tabla 1. Estos campos, incluyendo la Etiqueta, los bits de Exp, el bit S y el TTL, desempeñan funciones críticas en el enrutamiento y la gestión de paquetes en redes MPLS

Tabla 1
Estructura y tamaño de los campos que conforman el encabezado MPLS

Campo	Función	Tamaño
Etiqueta	Identificador de ruta en la red MPLS.	20 bits
Experimental	Bits de experimentación para calidad de servicio.	3 bits
Stack	Bit de apilamiento de etiquetas posteriores.	1 bit
Time to Live	Tiempo de vida de la etiqueta para evitar bucles.	8 bits

Nota. Adaptado de (IETF, 2001).

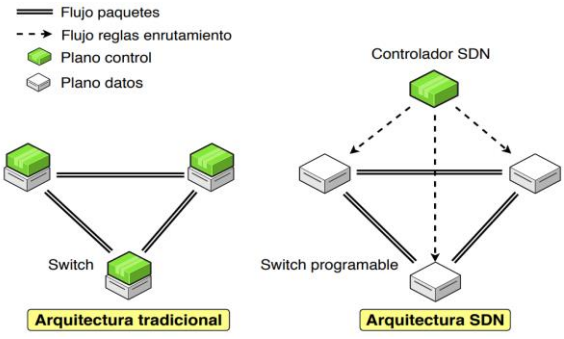
2.2 Redes Definidas por Software

Las Redes Definidas por Software, conocidas por sus siglas como SDN, representan un enfoque innovador en la gestión de redes. El RFC 7426 resalta que los administradores y operadores de red con esta tecnología poseen las capacidades de controlar, cambiar y gestionar dinámicamente el comportamiento de la red a través de un dispositivo central denominado controlador. Este elemento permite la asociación y configuración dinámica de los dispositivos SDN de forma individual, para posteriormente controlar su funcionamiento en la red de manera conjunta, mediante técnicas que permiten su diseño, administración y operación (IRTF, 2015).

El controlador SDN ocupa un lugar fundamental en la operabilidad de la red, debido a que desempeña un papel esencial en la orquestación y coordinación de operaciones en un entorno SDN. Además, el controlador se basa en el principio del desacoplamiento entre el plano de control, responsable de determinar las rutas de los paquetes, y el plano de datos, encargado de ejecutar estas decisiones y gestionar el flujo de tráfico; permitiendo obtener así un control centralizado y una visión consolidada de toda la infraestructura, simplificando la gestión y acelerando el despliegue de servicios de manera flexible y rápida respecto a una red tradicional, identificado sus diferencias en la Figura 4.

Figura 4

Diferencia entre una red tradicional y una red SDN

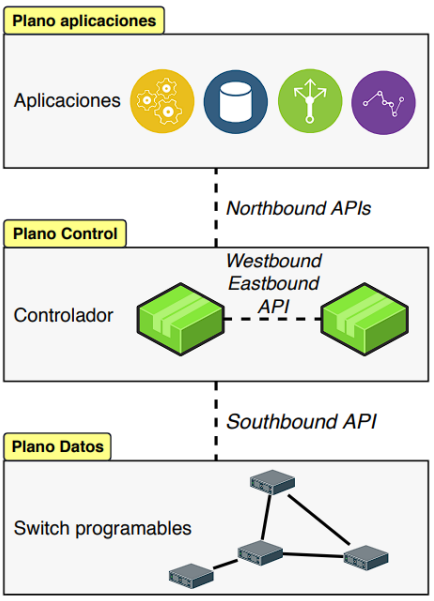


Nota. Obtenido de (Kivachuk, 2018).

2.2.1 Arquitectura SDN

La gestión centralizada que caracteriza a las SDN fundamentada en la separación de los planos de datos y control permite definir las características de funcionamiento de los dispositivos de red en función de las aplicaciones y servicios que se ejecutan en base a su arquitectura de tres capas. Cada capa de la arquitectura realiza su comunicación en base al uso de APIs[2], las cuales ofrecen una serie de protocolos que permiten la operación en conjunto de los elementos que conforman cada capa (Kivachuk, 2018). La arquitectura de red SDN mostrada en la Figura 5 permite evidenciar la constancia de las 3 capas correspondientes a los planos de aplicaciones, control y datos establecidas en el RFC 7426 (IRTF, 2015).

Figura 5
Elementos de la arquitectura SDN



Nota. Obtenido de (Kivachuk, 2018).

El plano de aplicaciones alberga los servicios y aplicaciones que desempeñan un papel fundamental en la definición del comportamiento de la red. Estas aplicaciones

[2] APIs: Interfaces de Programación de Aplicaciones.

ofrecen una vista integral de la infraestructura de SDN, incluyendo capacidades de balanceo de carga y administración. Esta visión global permite la gestión eficaz de la red y sus recursos a través del plano de control mediante el uso de una interfaz que permite su comunicación conocida como Northbound, basando su uso en un conjunto de APIs utilizadas por cada controlador acorde a las aplicaciones requeridas por el administrador (López, 2020).

La segunda capa en la arquitectura de una red definida por software es el plano de control, que se aloja en el software del controlador SDN. Este componente central es el encargado de programar y gestionar todas las operaciones dentro de la estructura de comunicación. Utilizando la información proporcionada por el plano de datos, el controlador realiza el proceso denominado como orquestación, el cual se basa en definir el funcionamiento de los elementos de la SDN, facilitando la coordinación en el intercambio de información y enrutamiento en la red (IRTF, 2015). Además, la comunicación entre los planos de control y datos es mediante el uso de la interfaz southbound, la cual el RFC 7426 indica que el protocolo OpenFlow es el principal utilizado en conmutadores SDN.

La tercera capa en la arquitectura SDN se denomina plano de datos, también conocido como capa de infraestructura. En esta capa, se encuentran todos los dispositivos que conforman la red, y su funcionamiento se basa en reglas definidas en las tablas de flujo proporcionadas por el controlador SDN. Estas reglas guían la forma en que los dispositivos manejan los paquetes entrantes, su procesamiento y enrutamiento (IRTF, 2015). Además, el plano de datos permite recolectar información que permite conocer el estado de la red, que se almacenada en los dispositivos locales y se envían al controlador para ser mostradas en forma estadística en el plano de aplicación (Kivachuk, 2018).

2.2.2 Protocolo OpenFlow

El protocolo OpenFlow es el primer estándar utilizado para la comunicación del controlador con los dispositivos de red como switches y routers físicos como elementos virtuales alojados en un hipervisor, que en conjunto conforman una red SDN con el propósito de ajustar la operabilidad del plano de datos de acuerdo con los requerimientos de la red (Moscoso, 2016). La arquitectura del protocolo OpenFlow identifica al controlador, tablas de flujo y dispositivos de red como componentes principales de una red SDN que operan en base a este protocolo (ONF, 2013). Los elementos que conforma la arquitectura son:

1. Controlador SDN: Software encargado de la conexión, configuración y administración de los conmutadores SDN en la red.
2. Tablas de Flujo: Instaladas en cada conmutador con el propósito de determinar las acciones a realizar con el tráfico de red.
3. Conmutadores SDN: Dispositivos físicos o virtuales encargados de realizar el reenvió de los paquetes dentro de la red SDN.

2.2.3 Mensajes OpenFlow

Los mensajes OpenFlow son unidades de comunicación en las redes SDN que permite la interacción entre el controlador y los conmutadores de la red. La (ONF, 2013) establece en su publicación tres categorías correspondientes a mensajes asincrónicos, mensajes simétricos y mensajes de controlador a conmutador.

En la dinámica de comunicación de los elementos de una SDN, tanto los controladores como los conmutadores tienen la capacidad de enviar mensajes simétricos que son utilizados comúnmente por el protocolo OpenFlow con el propósito de intercambio de información del funcionamiento de la red (ONF, 2013), siendo los mensajes más reconocidos:

- OFPT_HELLO: Consta de un encabezado OpenFlow más un conjunto de elementos de saludo que contienen datos opcionales para informar el protocolo de enlace inicial de la conexión.
- Echo Request: Consta de un encabezado OpenFlow más un campo de datos que contiene una marca de tiempo del mensaje para comprobar la latencia, diversas longitudes para medir el ancho de banda o de tamaño cero para verificar la continuidad entre el conmutador y el controlador
- Echo Replay: Consta de una cabecera de OpenFlow más el campo de datos no modificado de un mensaje de solicitud de eco.
- OFPT_ERROR_MSG: Son utilizados por el conmutador o el controlador para notificar problemas de conexión y son principalmente utilizados por el conmutador para indicar un fallo en una solicitud iniciada por el controlador.

Los mensajes de controlador a conmutador son iniciados por el controlador y son utilizados para solicitar funciones específicas o enviar paquetes por la ruta de datos del conmutador (ONF, 2013), siendo los mensajes más destacados:

- OFPT_FEATURES_REQUEST and OFPT_FEATURES_REPLY: Los mensajes de solicitud/respuesta intercambia la información principal sobre las identidades de los conmutadores y las capacidades de los conmutadores. Estos mensajes son enviados cuando un controlador solicita funciones al establecerse una nueva conexión al canal de control.
- OFPT_FLOW_MOD: Este mensaje indica una modificación a las entradas de la tabla de flujo cuando se agrega, modifica o elimina los flujos de tablas.
- OFMP_PORT_STATS: El controlador envía este mensaje para solicitar estadísticas sobre uno o varios puertos del conmutador. Las estadísticas pueden ser sobre paquetes y bytes recibidos o transmitidos, etc.

- OFPT_PACKET_OUT: Un controlador utiliza este tipo de mensaje para enviar un paquete a través de la ruta de datos al conmutador.

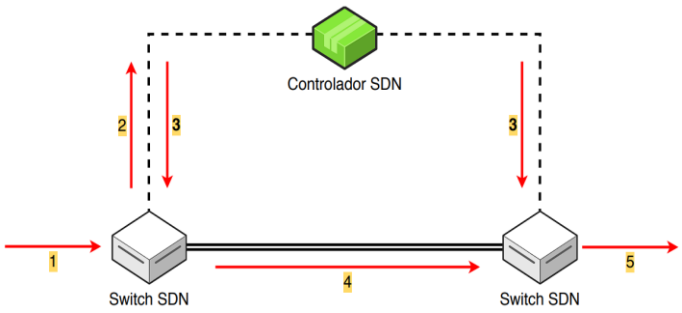
Por último, los mensajes asincrónicos son utilizados por los conmutadores para informar eventos de red a los controladores o cambios en sus estados (ONF, 2013), siendo los utilizados:

- OFPT_PACKET_IN: Los conmutadores envían mensajes OFPT_PACKET_IN a los controladores para transferir el control del paquete. Se activan mediante una entrada de flujo (una regla que especifica enviar mensajes OFPT_PACKET_IN) o por un error de tabla (cuando no se puede encontrar ninguna regla y el conmutador envía mensajes OFPT_PACKET_IN como comportamiento predeterminado).
- OFPT_FLOW_REMOVED: El controlador ha solicitado ser notificado cuando las entradas de flujo expiran o se eliminan de las tablas.
- OFPT_PORT_STATUS: El mensaje es enviado al controlador si se agregan, modifican y eliminan puertos de la ruta de datos.

2.2.4 Funcionamiento de las SDN

El funcionamiento de una red SDN respecto al proceso de conmutación de los paquetes que llegan a la red puede ser identificado en la Figura 6.

Figura 6
Escenario de funcionamiento de una red SDN



Nota. Obtenido de (Kivachuk, 2018).

1. Al recibir un paquete en la red SDN, el switch evalúa la acción que debe llevar a cabo según las reglas configuradas para diferentes tipos de tráfico. En función de esta evaluación, el switch optará por uno de los siguientes casos:
    - Enviar el paquete al controlador con el objetivo de definir la acción a realizar.
    - Realizar la conmutación del paquete hacia otro switch.
    - Descartar y finalizar el procesamiento del paquete en la red.
2. Cuando el switch recibe un paquete sin una política de gestión específica para ese tipo de tráfico, interrumpe el flujo y dirige el paquete al controlador.
3. El controlador al recibir el paquete establece una serie de parámetros a un número determinado de switches que considere adecuados para permitir el flujo del tráfico de este tipo de paquetes.
4. Los switches al recibir la política de gestión correspondiente para ese tipo de tráfico pueden iniciar con el proceso de conmutación hacia el siguiente switch, de modo que, para los paquetes futuros de ese mismo tipo, ya no será necesario consultar nuevamente al controlador para determinar la acción a seguir.
5. En el siguiente switch al que llega el paquete, ya se han establecido las políticas correspondientes para ese tipo de tráfico mediante el controlador. Por lo tanto, este switch procederá a conmutar el paquete sin necesidad de realizar consultas adicionales al controlador SDN.

2.2.5 Redes híbridas SDN/MPLS

Las redes híbridas SDN/MPLS son entornos de comunicación que combinan las ventajas de las Redes Definidas por Software y la tecnología de Conmutación de Etiquetas Multiprotocolo (MPLS). En estas redes que combinan las capacidades de estas dos tecnologías, la capa MPLS proporciona una base establecida para el enrutamiento y la

conmutación de paquetes, mientras que la SDN agrega una capa de control programable que permite una gestión más dinámica y centralizada de la red. Esto permite obtener una infraestructura de red adaptable y escalable que facilita la implementación de servicios y aplicaciones dentro de su funcionamiento (Lozada, 2022).

2.3 Seguridad en redes SDN

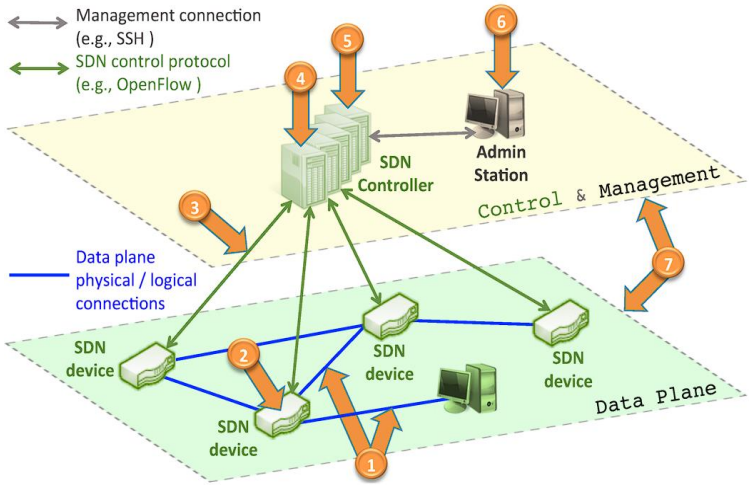
Las redes definidas por software enfrentan tanto amenazas de seguridad tradicionales que afectan a las redes convencionales como nuevos desafíos derivados de su arquitectura, esto se debe al uso de tecnología de virtualización para crear redes virtuales (VN), lo que conlleva la herencia de los problemas de seguridad tradicionales asociados con la virtualización de máquinas virtuales, así como la aparición de nuevos desafíos relacionados con la virtualización de hipervisores de red y funciones de red (Farahmandian & Hoang, 2016).

2.3.1 Vulnerabilidades de seguridad en redes SDN

En una red SDN, una vulnerabilidad representa un punto débil en la seguridad que los atacantes pueden aprovechar para comprometer la integridad, confidencialidad y disponibilidad de la información e infraestructura de la red. Estos riesgos pueden afectar a todos los elementos que conforman los distintos planos de la arquitectura SDN, de modo que tomando en cuenta este escenario en su publicación, (Kreutz et al., 2019) identifican siete vectores principales que representan las vulnerabilidades más significativas dentro de una red SDN operativa como se muestra en la Figura 7.

Figura 7

Vectores de vulnerabilidades de las redes SDN



Nota. Obtenido de (Ahmed et al., 2015).

Vector de vulnerabilidad 1: flujos de tráfico falsificados o manipulados, que pueden ser utilizados para atacar conmutadores y controladores en una red SDN, lo que conlleva a que esta amenaza puede surgir de dispositivos defectuosos o de usuarios malintencionados. Un atacante podría aprovechar elementos como servidores, conmutadores u ordenadores para llevar a cabo un ataque de denegación de servicio (DoS) contra los conmutadores OpenFlow, agotando los recursos del controlador que a su vez afecta directamente con las TCAM[3] saturando la búsqueda y tomar decisiones de reenvío de paquetes (Kreutz et al., 2019).

Vector de vulnerabilidad 2: explotación de funciones de los conmutadores SDN, que generan problemas significativos en la red ya que un único conmutador tiene el potencial de ralentizar o eliminar los paquetes en la red y a su vez pueden ser modificados para copiar o desviar el tráfico con propósitos maliciosos, o incluso sobrecargar el controlador o los conmutadores cercanos mediante la introducción de tráfico o solicitudes falsificadas (Ujcich et al., 2020).

[3] TCAM: Memoria Ternaria con Dirección por Contenido.

Vector de vulnerabilidad 3: ataques contra las comunicaciones del plano de control, que tienen el potencial de desencadenar ataques de denegación de servicios (DoS) y filtración de datos. Las SDN utilizan el modelo TLS/SSL para establecer la comunicación entre controladores y conmutadores, no obstante, (Georgiev et al., 2012) indican que la explotación de las debilidades evidenciadas en la infraestructura de PKI (Infraestructura de Clave Pública), puede llevar a graves consecuencias en la seguridad.

Vector de vulnerabilidad 4: ataques en controladores, que representan las amenazas más graves para las SDN. La razón se debe a que un controlador defectuoso o malicioso tiene el potencial de comprometer por completo la red y esto resulta en dificultades de identificación de las secuencias precisas de eventos que desencadenan comportamientos maliciosos, lo que a su vez prolonga el tiempo necesario para mitigar la amenaza (Kreutz et al., 2019). Además, las aplicaciones maliciosas pueden tener un amplio alcance en la red, ya que los controladores proporcionan abstracciones que se traducen en comandos de configuración para la infraestructura subyacente.

Vector de vulnerabilidad 5: ausencia de mecanismos que aseguren la confianza entre el controlador y las aplicaciones, esto se debe a que existen dificultades en establecer relaciones de confianza sólidas entre estos dos elementos, lo que conlleva a que procesos de certificación de las aplicaciones puedan ser suplantados con el objetivo de exponer a la red a riesgos de ejecuciones de aplicaciones maliciosas (Ruipérez, 2021).

Vector de vulnerabilidad 6: ataques a estaciones administrativas. Esto se debe a que el elemento central como lo es el controlador SDN requieren de control y configuración por parte de los administradores, lo que conlleva a que los servidores que alojan este elemento central sean el objetivo, debido a que comprometerlas significa poner en peligro las configuraciones que permiten el funcionamiento de la red (Dargahi et al., 2017).

Vector de vulnerabilidad 7: falta de recursos confiables para análisis forense y remediación ante ataques. La carencia de poseer los recursos necesarios que posibilitarían entender el origen de un problema identificado y proceder a un modo de recuperación rápido y seguro para esclarecer los detalles de un incidente es esencial ante ataques que comprometen contra la disponibilidad del controlador y los elementos de red que la conforma (Kreutz et al., 2019).

La Tabla 2 resume los siete vectores de vulnerabilidad identificados en las SDN como potenciales riesgos que pueden ser aprovechados por los atacantes con el propósito de interferir en la seguridad y operación de estas redes. Estos vectores representan áreas críticas que requieren una atención cuidadosa y medidas de prevención efectivas para garantizar la integridad, confidencialidad y disponibilidad de las SDN desde las primeras fases de diseño y despliegue.

Tabla 2
Vulnerabilidades y repercusiones generadas en las SDN

Vulnerabilidad	Especifico de las SDN	Repercusiones
Vector 1	No	Pueden ser puerta para ataques DoS
Vector 2	No	Funciones de los swithces SDN interrumpidas
Vector 3	Si	Filtración de datos de la red y el controlador puede ser utilizado para ataques DDoS
Vector 4	Si	Manejo mal intencionado del controlador compromete a toda la red
Vector 5	Si	Aplicaciones maliciosas pueden ser desarrolladas e implementadas en el controlador
Vector 6	No	Acceso no autorizado a elementos de la red SDN comprometen su funcionamiento
Vector 7	No	Respaldos ineficientes no aseguran la recuperación de la red ante fallos

Nota. Adaptado de (Ahmed et al., 2015)

2.3.2 Ataques en redes SDN

Los ataques en redes SDN[4] pueden ser categorizados en función a su relación con los desafíos asociados al flujo de información, amenazas a los componentes de la arquitectura SDN y la interrupción de las funciones de administración y control de la red. Según la investigación de (Ujcich et al., 2020), indica que los principales ataques que afectan a las funciones mencionadas corresponden a los ataques (DoS) que se fundamentan en los principios de los ataques de inundación conocidos como flooding[5].

2.3.3 Ataques de denegación de servicio (DoS) en SDN

Los ataques denegación de servicio (DoS) se basan en el principio de inundar al objetivo con un gran número de solicitudes de semiconexión. Esto provoca una sobrecarga en las funciones de reenvió de paquetes de datos realizadas por los conmutadores. El éxito de un ataque DoS conlleva en cumplir con el objetivo de interrumpir la comunicación entre los conmutadores y el controlador, o incluso entre los propios conmutadores (Klöti et al., 2013). En esta investigación se enfatiza el estudio de dos objetivos de los ataques DoS de tipo inundación que afectan directamente a los planos de datos y control de la arquitectura SDN, los cuales son:

a) Ataques al ancho de banda del plano de control

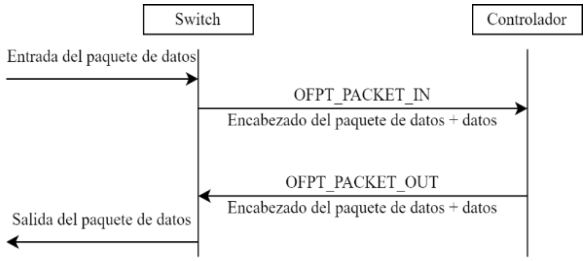
Cuando un conmutador recibe por primera vez un paquete que pertenece a un nuevo flujo, almacena el paquete en un buffer antes de enviar un mensaje OFPT_PACKET_IN al controlador. Este mensaje incluye un máximo de 128 bytes del encabezado del paquete recibido. No obstante, si el conmutador no tiene capacidades de

[4] Ataques en redes SDN: Definido como cualquier intento mal intencionado de comprometer, interrumpir o dañar el funcionamiento normal de una red definida por software (Yoon et al., 2022).
[5] Ataques flooding: Interrupciones generadas por atacantes en los sistemas de comunicaciones cuando la tasa de información a procesar supera la capacidad de procesamiento del elemento de la red comprometido (Kissel, 2011)

almacenamiento en búfer o si el búfer de entrada está lleno, la especificación definida por (ONF, 2013) dicta que el conmutador envíe el paquete completo al controlador, encapsulándolo dentro del mensaje OFPT_PACKET_IN. Si este proceso es realizado, el controlador responde utilizando el mensaje OFPT_PACKET_OUT que también transporta el paquete de datos completo. En este proceso, no se establece ninguna regla de flujo y el conmutador simplemente realiza el proceso de conmutación del paquete al puerto especificado, este escenario es ilustrado mediante un flujograma en la Figura 8.

Figura 8

Mensajes de sobrealmacenamiento de paquetes OFPT en el buffer



Nota. Adaptado de (Kandoi & Antikainen, 2015).

Cuando un conmutador experimenta una alta tasa de llegada de nuevos flujos de paquetes en un corto período, su búfer empieza a saturarse, lo que resulta en la retransmisión de paquetes completos al controlador (Zheng et al., 2020). Esta situación implica un significativo consumo del ancho de banda del plano de control y como consecuencia aumenta la latencia en la instalación de nuevas entradas en la tabla de flujo y en situaciones extremas, el conmutador podría no ser capaz de reenviar el tráfico de estos nuevos flujos (Klöti et al., 2013). La pérdida de paquetes puede manifestarse de varias maneras:

- El tamaño de la cola de salida del conmutador es limitado: si el enlace entre el conmutador y el controlador está congestionado, la cola de salida del conmutador

se llena y el mensaje OFPT_PACKET_IN en sí no se puede enviar, lo que provoca la pérdida de paquetes.

- Alta latencia: la congestión puede introducir latencia en el canal de control. Si el conmutador no recibe una respuesta OFPT_FLOW_MOD dentro de un cierto período de tiempo, descarta el paquete almacenado en el buffer correspondiente.

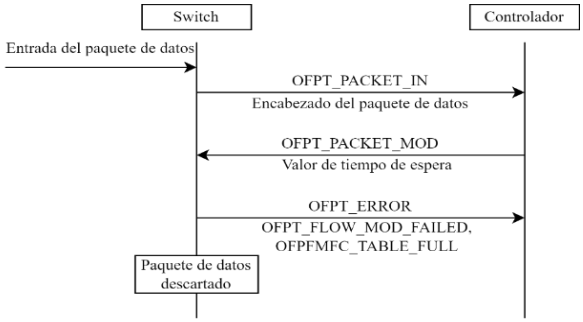
Las repercusiones mencionadas implican un ataque al conmutador objetivo, sin embargo, si el ataque logra abrumar los recursos informáticos del controlador y deniega su servicio, todos los conmutadores conectados al controlador afectado pueden sentir el impacto del ataque (Yue et al., 2020).

b) Ataques a las tablas de flujo del conmutador

Los conmutadores tienen memoria limitada para almacenar reglas de flujo y el tamaño limitado de las tablas es el principal objetivo de los ataques DoS para interrumpir las operaciones de la infraestructura SDN. Una vez que las tablas de flujo han ocupado toda su capacidad el conmutador SDN detecta que no puede instalar una regla adicional, por lo que envía un mensaje de ERROR_OFPT al controlador con el código OFPFMFC_TABLE_FULL, luego descarta el paquete debido a que el switch no puede realizar el proceso de conmutación del paquete (Kandoi & Antikainen, 2015), esto se evidencia en la Figura 9.

Figura 9

Mensajes de tablas de flujo llenas en el conmutador



Nota. Adaptado de (Kandoi & Antikainen, 2015).

El conmutador no puede reenviar paquetes almacenados en el buffer hasta que haya espacio en la tabla de flujo para instalar nuevas reglas de flujo, por consecuente los paquetes que no están almacenados en buffer no se ven afectados ya que no es necesario instalar reglas de flujo para ellos sobrecargando las capacidades del conmutador interrumpiendo sus funciones (Ramachandran & Shanmugam, 2017), de modo que el impacto de este ataque es local para el conmutador.

2.4 Sistemas de detección de Intrusiones

Un sistema de detección de intrusiones o más conocidos como IDS basa su funcionamiento en la detección de acciones mal intencionadas que comprometan el funcionamiento de la red o de un dispositivo en específico, llevando de esta forma un monitoreo del tráfico que circula por la red con el objetivo de identificar este tipo de actividades anómalas (INCIBE, 2020). Esto implica que el sistema realizará una cotización del tráfico entrante, con el propósito de catalogarlo como sospechoso o fuera de lo común si se detectan patrones inusuales, permitiendo generar un mensaje de alerta que permita tomar medidas de respuesta ante los registros identificados por el mecanismo de seguridad.

Un IDS puede formar parte de un sistema de seguridad robusto y complejo que admita el uso de diferentes tecnologías con la finalidad de aportar medidas de ciberseguridad que ayuden a evitar incidentes relacionados a la ciberdelincuencia. Su funcionamiento se basa principalmente en la supervisión en tiempo real mediante puertos que pueden ser de tipo SPAN[6] o TAP[7] para monitorear el tráfico (Sharma, 2023), con este tipo de monitoreo el sistema verifica que las actividades anómalas sean bloqueadas para

[6] Switched Port Analyzer (SPAN): Utilizados en switches para copiar el tráfico de uno o varios puertos y enviarlo a otro puerto designado, permitiendo a los administradores de red examinar el tráfico de manera no intrusiva para realizar análisis de seguridad sin afectar el flujo normal de datos (INCIBE, 2020).
[7] Test Access Point (TAP): Dispositivos físicos que se intercalan en una línea de red para duplicar el tráfico y enviarlo a dispositivos de monitorización o análisis (Fares et al., 2019).

evitar ataques. Es así, que se considera que un IPS actuaría de forma transparente para la red de forma inmediata ante la presencia de posibles amenazas (Nasereddin & Abdelkarim, 2011).

2.4.1 Tipos de IDS

Los sistemas de detección de intrusos son en su mayoría de tipo pasivos, por lo que su clasificación se basa en el modo de aplicación de estos, tres de los tipos de IDS son los siguientes:

a) Sistema de Detección de intrusiones en la red (NIDS)

Es un sistema de detección de intrusiones que se coloca en sectores específicos de la red para monitorear de forma completa toda la red, cubriendo especialmente áreas vulnerables del perímetro la misma, son generalmente aplicados a subredes enteras para supervisar de forma pasiva el tráfico que fluye por la red considerando las formas de detección que pueden ser empleadas (Prokopets, 2023).

b) Sistema de Detección de intrusiones en host (HIDS)

Este sistema funciona en dispositivos de red individuales, lo cual indica que se realizará un monitoreo detallado de cada host o dispositivo, teniendo como punto de partida la comparación de archivos de sistema que se toman progresivamente a modo de "instantáneas" para realizar una comparación entre las versiones y notificar en caso de encontrar diferencias que salgan de los patrones de comportamiento propios del host (Prokopets, 2023).

c) Detección de anomalías en el comportamiento de la red (NBAD)

Este tipo de detección se basa en la visualización del tráfico en varios segmentos de la red para determinar la existencia de comportamientos fuera de lo normal; para este propósito, se considera el uso de sensores que permitan la creación de registros para

realizar comparativas del tráfico que se catalogue como normal para la red y el tráfico que se encuentra en circulación por la red (Santos Kumar et al., 2013).

2.5 Sistemas de Prevención de Intrusiones

Un IPS o Sistema de Prevención de Intrusiones, es aquel que se encuentra realizando un monitoreo constante a una red o host determinado para así verificar que las actividades anómalas sean bloqueadas para evitar ataques. Es así, que se considera que un IPS actuaría de forma transparente para la red actuando de forma inmediata ante la presencia de posibles amenazas (Nasereddin & Abdelkarim, 2011).

Es común que se asocie a un IPS con un firewall tradicional, sin embargo, la diferencia más destacable entre ambos radica en que, al hablar de seguridad, un firewall hace uso de políticas de bloqueo que restringen todo el tráfico salvo el que el corresponda a excepciones establecidas, contrario a un IPS que permite el paso de todo el tráfico exceptuando aquel que tenga una razón para ser bloqueado (Fuchsberger, 2005). Además, debido al tipo de detección que maneje el IPS que bien puede ser, basada en firmas o en anomalías realiza un análisis y monitoreo más profundo de los paquetes que se encuentren circulando en la red con el objetivo prevenir posibles ataques.

El funcionamiento de un IPS, hace uso de bases de datos que permitan detectar comportamientos que se puedan asimilar con intrusiones, de esta forma se pueden definir tecnologías de detección diferentes; las tecnologías basadas en firmas se caracterizan por manejar comparaciones de firmas de amenazas conocidas cuya identificación única se halla definida en el código de un exploit que puede ser descubierto y añadido a la base de datos con el fin de reducir los falsos positivos producto de la detección de exploits individuales que se pueden llegar a etiquetar erróneamente como una amenaza (Huawei, 2023). Por otro lado, la detección basada en anomalías permite tomar muestras del tráfico de la red y genera comparaciones con estándares de comportamiento básicos y genera

alertas y acciones de bloque al hallar muestras que salen del lineamiento de comportamientos normales (Huawei, 2023).

2.5.1 Tipos de IPS

Es posible definir a los IPS como sistemas de protección activa que pueden fácilmente reaccionar y bloquear paquetes que representen una amenaza para la red en general, más al igual que los IDS pueden tener diferentes enfoques por lo que se los cataloga en diferentes tipos.

a) Sistemas de prevención de intrusiones basados en red (NIPS)

Son sistemas que se encargan de monitorear la red completa para detectar actividades sospechosas, detectan anomalías en cada de red y capa transporte de múltiples segmentos de la red en base a la ubicación que este tenga dentro de la red, su desventaja radica en la robustez que debe tener el IPS para poder mantener bajo control la red completa de forma exitosa ya que debe analizar una amplia gana de protocolos (Gillis, 2023).

b) Sistemas de prevención basados en hosts (HIPS)

Este tipo de sistemas se limitan a una cobertura más pequeña al residir en un host de la red y analizar y bloquear únicamente el tráfico propio de este host (Gillis, 2023), los HIPS analizan la actividad del sistema operativo, y actividades de capa aplicación de los hosts además de los paquetes de cada de red de forma directa.

c) Sistemas para análisis de comportamiento de la red (NBA)

Para analizar el comportamiento de la red se basa en monitorear múltiples segmentos y grupos de hosts con especial atención en la capa de red que puede provocar flujos anormales en la red, se considera que tiene una mayor eficacia que otros IPS ya que

puede ser efectivo incluso contra ataques de DDoS en diferentes puntos de la red (Gillis, 2023).

2.6 *IDS/IPS en redes SDN*

Al igual que en una red convencional, una red definida por software es vulnerable a diferentes tipos de ataques, por lo que es importante considerar la implementación de medidas de seguridad que permitan garantizar que la red se mantenga íntegra, confidencial y disponible para cumplir sus funciones regulares. Si bien estas redes traen muchos beneficios, traen consigo también un incremento en el riesgo de sufrir ataques de denegación de servicio (DoS), que coloque en peligro al controlador central afectando a la red por completo; debido a que el protocolo OpenFlow no se enfoca directamente en el monitoreo de la seguridad de la red, la implementación de medidas de seguridad como los IPS e IDS permiten gestionar de mejor forma estas potenciales amenazas disminuyendo así el riesgo existente (Fares et al., 2019).

Al manejar una SDN se cuenta con la ventaja de que es posible integrar soluciones y mejoras de forma directa al plano de datos y que las mismas se apliquen a cada equipo de red que se encuentre enlazado con el protocolo OpenFlow (Fares et al., 2019), por lo que la implementación de sistemas de detección y prevención de intrusos pueden ayudar a que, se realicen procesos para aislar de forma directa todo el tráfico que se catalogue como sospechoso y se provea información al administrador de la red que le permita determinar el objetivo de ataque y de donde proviene el mismo, garantizando de esta forma que la red no se vea afectada y pueda seguir operando normalmente.

CAPÍTULO III: Despliegue y Funcionamiento

Esta sección del proyecto está enfocada en el diseño, implementación y mitigación de ataques DoS de tipo inundación dirigidos a una red híbrida SDN/MPLS, de modo que el objetivo principal de este apartado es evaluar y contrarrestar las consecuencias que generan este tipo de ataques a la disponibilidad e integridad de la red y sus servicios.

Para cumplir con este apartado, es necesario el despliegue de dos entornos de prueba que conforman un testbed, donde el primer escenario presenta un entorno de ataques DoS de tipo inundación a una red SDN/MPLS completamente operativa. Los resultados de este escenario permiten establecer criterios de evaluación de repercusiones generadas a la red y sus dispositivos que permiten el desarrollo del segundo escenario de pruebas correspondiente en implementar un sistema de seguridad IDS/IPS, que se basará en un conjunto de reglas para garantizar la continuidad de la red durante ataques de denegación de servicio.

3.1 Evaluación de herramientas de simulación

En el ámbito de investigación del testbed, la selección apropiada de un software de código abierto que permita la simulación de los escenarios es fundamental para garantizar la efectividad de los estudios a realizar. Por lo tanto, en esta sección se llevará a cabo una evaluación de las herramientas GNS3, OMNet++ y Mininet con el objetivo de identificar la plataforma que presente las mejores prestaciones para la implementación de los escenarios. Esta evaluación se basa en el modelo ISO/IEC 25010, el cual se enfoca en determinar la calidad de un producto de software a partir de nueve características que son identificadas en la Figura 10 y seleccionadas de acuerdo con las métricas de requerimientos especificadas en el Anexo 1.

Figura 10
Modelo ISO/IEC 25010 para calidad del producto de software

Nota. Tomado de (ISO 25000, 2022)

3.1.1 Graphical Network Simulator 3

El software de simulación de redes GNS3, reconocido por sus siglas, destaca como una herramienta fundamental en el ámbito de estudio e investigación de escenarios de redes de datos. Esta plataforma se distingue por su capacidad para configurar, evaluar y resolver problemas tanto en entornos virtuales como en entornos reales de redes. Su versatilidad permite la emulación precisa de dispositivos de red reales, facilitando así la interconexión y el funcionamiento de topologías complejas que requieren de una variedad de complementos de software para su validación. Además, GNS3 ofrece flexibilidad en el diseño y la evaluación de escenarios administrados, proporcionando un entorno controlado para todos los componentes y procesos implementados en la red simulada (GNS3, 2024).

3.1.2 Mininet

Es una plataforma de emulación de redes de código abierto que permite crear redes virtuales a gran escala para propósitos de desarrollo e investigación, de modo que esta herramienta posibilita la emulación de redes conformadas por nodos, enrutadores, enlaces y switches dentro de un entorno controlado y virtualizado. Adicionalmente, el software

es utilizado para realizar simulaciones que involucran SDN que basan su funcionamiento en el protocolo OpenFlow. Adicionalmente, entre una de sus características implica que su ejecución se realiza mediante una interfaz de línea de comandos (CLI) donde se puede interactuar y verificar el funcionamiento de la red debido a que solo es compatible con entornos Linux.

3.1.3 OMNet++

Se presenta como un simulador de eventos discretos con un enfoque modular y orientado a objetos, lo que conlleva a que un modelo en OMNeT++ se estructura mediante módulos jerárquicos que se comunican a través de mensajes. Además, su aplicación es variada y abarca desde la modelación de tráfico en redes de telecomunicaciones, protocolos y sistemas multiprocesadores, hasta la evaluación de aspectos de rendimiento en sistemas de software complejos (Modenesi et al., 2022).

El software en el ámbito de las SDN proporciona una simulación detallada de las interacciones y comportamientos de red, lo que le confiere la capacidad de modelar protocolos y algoritmos utilizados en el funcionamiento de las SDN. Además, permite a los usuarios diseñar y evaluar el rendimiento de estrategias de enrutamiento, políticas de gestión de tráfico y otros aspectos relacionados con la SDN (OMNeT, 2021). Es importante destacar que, aunque ofrece una amplia flexibilidad, todos los escenarios y elementos de red deben ser programados según las necesidades específicas del usuario.

3.1.4 Elección de la plataforma para la implementación del testbed

En la evaluación de las herramientas de simulación correspondientes a OMNeT++, Mininet y GNS3 se destacan como opciones prominentes, para ser consideradas como plataformas que alojarán los escenarios de prueba. El Anexo 1 presenta el proceso de establecimiento del conjunto de métricas de requerimientos que permiten la evaluación de la calidad del software de cada una de estas herramientas de

diseño de redes. Esto facilita una comparación exhaustiva y una selección informada, basadas en las características de eficiencia de desempeño, compatibilidad y capacidad de interacción establecidas por la norma ISO/IEC 25010.

La Tabla 3 exhibe las métricas de requerimientos con un nivel de prioridad alto, seleccionadas para establecer el sistema de evaluación de las plataformas. En su estructura, cada métrica está identificada por su nomenclatura, la característica correspondiente de la norma ISO a la que pertenece, y su respectiva descripción.

Tabla 3
Requerimientos de alta prioridad para seleccionar el software de simulación

Abreviatura	Característica	Descripción
MtRED1	Eficiencia de desempeño	La plataforma de simulación debe utilizar la CPU de manera eficiente, asegurando que no exceda el límite especificado de uso de recursos de la CPU bajo condiciones normales de funcionamiento.
MtRED2		El software de simulación debe gestionar eficientemente la memoria RAM, asegurando que no utilice más memoria de la especificada.
MtRED5		Función de procesar una cantidad específica de datos o tareas, asegurándose de presentar la información del uso de los recursos del sistema al usuario.
MtRC1	Compatibilidad	La plataforma de simulación posee funciones que permiten implementar máquinas virtuales mediante softwares de virtualización y contendores Docker.
MtRC2		La herramienta habilita la integración de los sistemas operativos de dispositivos de red tradicionales y redes definidas por software.
MtRC4		La herramienta de simulación permite la convergencia entre los softwares de virtualización con sus funciones propias para simular la comunicación mediante diferentes tecnologías de red.
MtRCI1	Capacidad de interacción	El software posee una interfaz gráfica que facilita la implementación y conexión de los elementos de red.

Al considerar la ponderación de cada métrica en relación con las características seleccionadas y tener a consideración que la plataforma seleccionada será la encargada

de simular la detección y mitigación de una red SDN bajo ataques DoS. Esto implica que la selección será en base a la plataforma que optimicen el uso de los recursos del sistema, facilite la interoperabilidad con softwares externos de virtualización y, al mismo tiempo, sean intuitivas para el usuario.

Con base en este criterio de selección, la Tabla 4 muestra la ponderación obtenida, donde el valor de 1 indica que el software cumple con el requerimiento, mientras que el 0 indica lo contrario. Esto permite obtener una tabla de evaluación de las plataformas, en la cual el software de simulación que obtenga la puntuación más alta será seleccionado.

Tabla 4
Evaluación de las plataformas de simulación de redes

Software	MtRED 1	MtRED 2	MtRED 5	MtRC 1	MtRC 2	MtRC 4	MtRCI 1	Total
GNS3	1	1	1	1	1	1	1	7
Mininet	1	1	0	1	0	1	1	5
Omnet	1	1	0	0	0	0	0	2

De acuerdo con los resultados obtenidos en la Tabla 4, el software de GNS3 se destaca como la plataforma de diseño de redes de código abierto ideal para alojar los escenarios del testbed. Esto se debe a que GNS3 muestra un rendimiento óptimo en términos de eficiencia de desempeño, garantizando un uso eficiente de la CPU, una gestión adecuada de la memoria RAM y la presentación detallada de información sobre el uso de los recursos del sistema. Además, demuestra una excelente capacidad de coexistencia al permitir la implementación de máquinas virtuales, integrar sistemas operativos de dispositivos de red tradicionales y facilitar la convergencia entre diferentes tecnologías de red de datos.

Los puntajes obtenidos por GNS3, conforme a los requerimientos de alta prioridad establecidos en la Tabla *3*, consolidan su posición como la solución más completa e idónea para cumplir con los objetivos del proyecto. Tomando la premisa que el testbed

debe alojarse en un escenario de acceso para los estudiantes de la Carrera de Telecomunicaciones de la Universidad Técnica del Norte, el

Anexo 2 proporciona la información de acceso y especificaciones del sistema del host Debian que alojará los escenarios, mientras que el Anexo 3 detalla los requisitos y el procedimiento de instalación de GNS3 en sistemas Linux basados en Debian.

3.2 Red híbrida SDN/MPLS

La implementación de la red híbrida SDN/MPLS marca uno de los procesos iniciales en el desarrollo del presente trabajo de investigación. Esta etapa es esencial, ya que establece la infraestructura de red general, que posteriormente alojará los dos escenarios del testbed: el entorno de ataques DoS y la implementación del mecanismo de seguridad contra este tipo de amenazas. Esta fase abarca aspectos clave como el diseño, configuración y puesta en marcha de la red infraestructura híbrida SDN. En esta sección, se profundiza en la justificación del diseño general de la topología, los requisitos de los sistemas que componen la infraestructura de red híbrida, incluyendo hosts y dispositivos de red. Además, se aborda el direccionamiento IPv4 de la topología lógica, la selección e implementación del controlador SDN, y el despliegue efectivo de la red híbrida SDN/MPLS. Este enfoque detallado proporciona la información necesaria para comprender el funcionamiento de este tipo de redes y sienta las bases para el despliegue de los escenarios del testbed.

3.2.1 Diseño de la topología de red

El diseño de red propuesto del escenario general del testbed presentado en la Figura 11 se fundamenta en una infraestructura de red híbrida SDN/MPLS. Esta estructura se apoya en la interconexión entre dos componentes principales: la infraestructura SDN, compuesta por conmutadores que son administrados por un controlador SDN, y la infraestructura de red tradicional, que cuenta con dispositivos de

red basados en hardware. Estos elementos se integran mediante el uso del plano de datos de la arquitectura SDN, lo que facilita la interoperabilidad entre ambas tecnologías de red.

El enfoque del diseño de la topología se centra en establecer un entorno de red que aproveche las capacidades de conmutación de la infraestructura SDN, integrándolas con tecnologías que mejoran la conmutación de paquetes como lo es MPLS. Esto permite obtener una red híbrida en la que la infraestructura SDN asume las funciones de conmutación de etiquetas, mientras que los routers tradicionales se encargan del encaminamiento de los paquetes mediante la selección de un protocolo de enrutamiento especificado en el

Anexo ***4***. Esta interoperabilidad de las redes definidas por software ofrece una solución integral que combina su flexibilidad y programabilidad con la eficiencia y escalabilidad de MPLS, asegurando un rendimiento óptimo y una gestión eficaz del tráfico de la red.

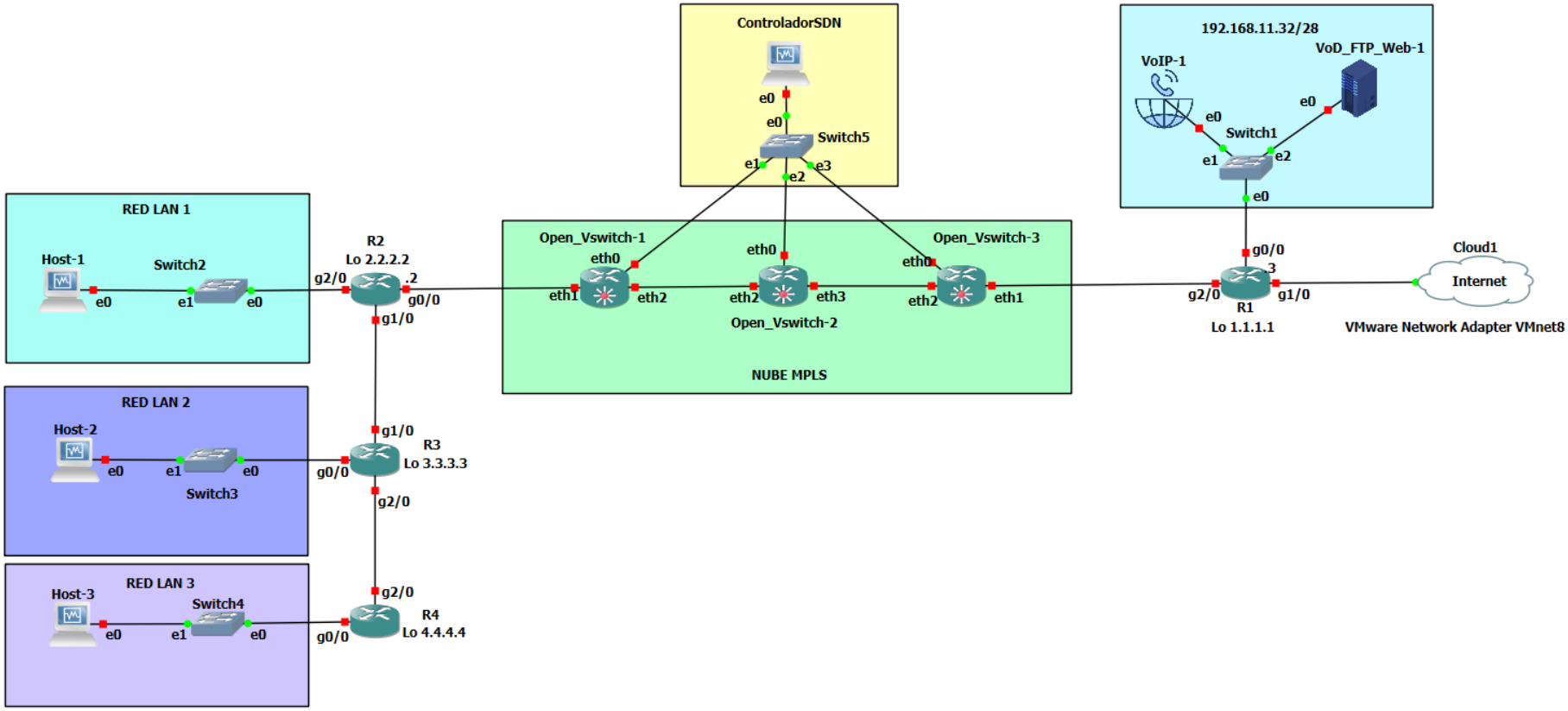
En lo que respecta a la infraestructura de la nube SDN/MPLS se basa en los principios de una red de backbone distribuida, donde el tráfico de datos se transporta a través de múltiples nodos o centros de conexión en lugar de depender de un único punto centralizado. La representación esquemática en la Figura 11 ilustra esta interconexión mediante 3 conmutadores (OpenVswitches) que conforman una topología de árbol, en la cual cada conmutador SDN establece un enlace con el controlador para que puedan ser administrados por este elemento. Esta configuración le otorga a la topología características como tolerancia ante fallos y a la vez escalabilidad, debido a que la red puede adaptarse acorde a las necesidades de adición o eliminación de equipos o enlaces en la red.

El diseño establece también la ubicación del controlador SDN de acuerdo con la distribución de los conmutadores en la topología, de modo que la fundamentación de una conexión directa del controlador SDN a la nube MPLS en la topología lógica que es

representada en la Figura 11 se convierte en una estrategia fundamental para garantizar una mayor seguridad en la infraestructura. Esta configuración permite aislar la red de control SDN de la infraestructura general, reduciendo así los puntos de acceso potenciales ante elementos que quieran comprometer el controlador SDN. Además, al vincular el controlador directamente a la nube MPLS, se minimiza el riesgo de exposición a través de dispositivos intermedios como los routers tradicionales, manteniendo un nivel adecuado de seguridad.

Además, en la topología se considera la implementación de una zona de seguridad perimetral (DMZ) con el propósito de reforzar la seguridad, especialmente para los servicios de VoIP, WEB, FTP y VoD accesibles desde Internet. Asimismo, se destaca la relevancia del router de frontera (R1), encargado de separar la red interna de la red pública, ya que funciona como el último punto de control antes de acceder a Internet, siendo la primera y última barrera de defensa en el sistema.

Figura 11
Topología de red SDN/MPLS



3.2.2 Dispositivos que conforman la topología

Una vez definido el escenario general del testbed y ubicados los dispositivos conforme al diseño de la topología representada en el apartado anterior, esta sección se encarga en detallar la información sobre cada uno de los hosts, dispositivos de red y servidores que conforman la topología, centrándose en las especificaciones de uso de recursos del sistema dentro del entorno de simulación. La Tabla *5* presenta información detallada sobre cada dispositivo en términos uso de memoria RAM y CPU, tomando en cuenta las capacidades del host anfitrión presentadas en la **Tabla *60***, así como los sistemas operativos utilizados por cada dispositivo y las funciones que cumplen dentro de la red híbrida SDN.

Tabla 5
Dispositivos que conforman las topologías

Dispositivo	RAM	CPU	Sistema Operativo	Rol en la red
Controlador	1 GB	2	Ubuntu 20.04	Controlador SDN
Switches	256 MB	1	Open vSwitch 2.17.8	Switches SDN
Routers c7200	512 MB	1	Cisco IOS 15.2(4)S6	Enrutadores
VoIP	512 MB	1	Issabel-CentOS 7	Servidor de VoIP
FTP_Web_Hosting	1 GB	2	Ubuntu 20.04	Servicios
Hosts 1-3	2 GB	1	Debian 12	Hosts LANs

3.2.3 Direccionamiento IPv4 de la topología

Establecida la distribución lógica y de dispositivos de cada uno de los elementos de red que conforman el escenario general del testbed, la siguiente fase que tiene gran relevancia corresponde a la asignación de direccionamiento IP correspondiente para cada enlace, mismo que se encuentra configurado dentro de los enrutadores, conmutadores SDN, servidores y host correspondientes a cada LAN. La Tabla 6 se describe a detalle el dispositivo identificado en la topología, su interfaz, dirección IP, máscara de red, gateway

y la dirección de la subred. Para aclarar, se ha tomado como referencia la IP 192.168.11.0 /24, la cual por medio de VLSM se ha determinado las siguientes máscaras: /28 para las redes LAN y DMZ, /27 correspondiente a los enlaces de la nube MPLS, y finalmente un /30 para los enlaces WAN que conforman la topología.

Tabla 6
Direccionamiento IPv4 de la topología del escenario del testbed

Disp.	Interfaz	Dirección IP	Gateway	Máscara de subred
R1	f0/0	192.168.11.33	N/D	255.255.255.240
	f1/0	10.10.10.2	10.10.10.1	255.255.255.252
	f2/0	192.168.11.1	N/D	255.255.255.240
R2	f0/0	192.168.11.2	N/D	255.255.255.224
	f1/0	192.168.11.97	N/D	255.255.255.252
	f2/0	192.168.11.49	N/D	255.255.255.240
R3	f0/0	192.168.11.65	N/D	255.255.255.240
	f1/0	192.168.11.98	N/D	255.255.255.252
	f2/0	192.168.11.101	N/D	255.255.255.252
R4	f0/0	192.168.11.81	N/D	255.255.255.240
	f2/0	192.168.11.102	N/D	255.255.255.252
OpenVsw1	eth0	192.168.11.3	192.168.11.1	255.255.255.224
OpenVsw2	eth0	192.168.11.4	192.168.11.1	255.255.255.224
OpenVsw3	eth0	192.168.11.5	192.168.11.1	255.255.255.224
VoIP	eth0	192.168.11.34	192.168.11.33	255.255.255.240
VoD_FTP_Web	eth0	192.168.11.35	192.168.11.33	255.255.255.240

3.2.4 Selección del controlador SDN

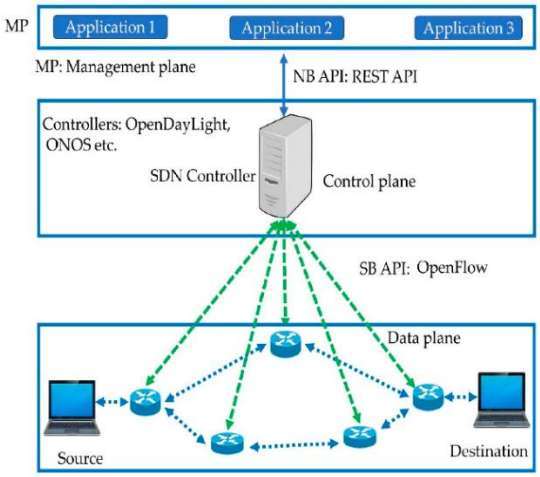
La elección del controlador es uno de los componentes fundamentales para la implementación de la SDN. Esto se debe a que es necesario obtener el escenario de red que cumpla con los componentes de hardware y software que conforman las 3 capas de la arquitectura SDN, siendo los siguientes apartados:

a) Plano de datos

El plano de datos tiene la responsabilidad de asegurar la interconexión y facilitar la comunicación entre los usuarios finales y los servidores dentro de la red SDN/MPLS.

Para que esta operación sea ejecutada, es esencial contar con conmutadores SDN que sean compatibles con el protocolo OpenFlow, debido a que el protocolo permite definir en estos dispositivos las reglas de flujo y las rutas que serán utilizadas para el reenvío de paquetes o a su vez su modificación o descarte según el funcionamiento programado de la SDN. Figura 12 representa el modelo centralizado utilizado en los conmutadores OpenvSwitch que conforman la topología de red, en donde cada conmutador perteneciente al plano de datos es administrado por el controlador SDN.

Figura 12
Infraestructura del plano de datos centralizado de la SDN



Nota. Obtenido de (Ali et al., 2020).

b) Plano de control

El plano del control posee las funciones centralizadas correspondientes a la gestión, supervisión y administración de los conmutadores de la red SDN en conjunto con las reglas y políticas de enrutamiento que permiten su conectividad a las redes tradicionales que forman la red de los escenarios, de modo que esta capa de la arquitectura SDN aloja todas estas funciones en un elemento principal denominado controlador SDN. Este elemento es el responsable de enrutar los paquetes de datos en la red para determinar la mejor ruta para cada paquete y posteriormente enviar instrucciones a los conmutadores

para que los paquetes sean enrutados de acuerdo con las reglas predefinidas por el controlador.

Existen diferentes distribuciones y versiones de controladores de código abierto que pueden ser utilizados para la implementación de la SDN en la topología propuesta, no obstante, la elección de este elemento deberá ser en base al soporte del protocolo OpenFlow en conjunto con las vulnerabilidades de este protocolo y los conmutadores OpenVswitch ante ataques DoS. La ONF en el año 2016 en su publicación titulada "*Requerimientos de seguridad para controladores SDN*" describe los requisitos de seguridad de los activos críticos de los controladores SDN correspondientes a ataques DoS, datos de configuración, gestión de las interfaces de comunicación y autenticación de usuarios.

Considerando la publicación de la (ONF, 2016) respecto la información obtenida de varios controladores SDN de código abierto en lo que respecta a sus requisitos de seguridad, han permitido establecer la Tabla *7* con la lista de características de seguridad y soporte requeridas para la administración de la red SDN. Esta tabla utiliza la nomenclatura Crts_# para referirse a la característica de seguridad y soporte identificado para la selección de un controlador en base a sus aspectos de seguridad.

Tabla 7
Características de seguridad para controladores SDN

Característica	Descripción
Crts_1	Soporte del protocolo OpenFlow en versiones 1.0 en adelante.
Crts_2	Protección de la integridad de los datos en tránsito.
Crts_3	Funciones de registro de actividad en el controlador.
Crts_4	Protección de la confidencialidad de los datos en tránsito.
Crts_5	Anti-DoS por agotamiento de la capacidad de cómputo.
Crts_6	Anti-DoS por consumo excesivo de recursos.
Crts_7	Anti-DoS desde interfaces Northbound/Southbound.

*No*ta. Adaptado de (ONF, 2016).

Las características de seguridad consideradas para la seleccionar el controlador se evalúan mediante un sistema de puntuación establecido en la en la Tabla *8*. Esta tabla califica a los controladores según la presencia de las cualidades de seguridad identificadas por la publicación de la (ONF, 2016), las cuales se encuentran recopiladas en la Tabla *7*. En este sistema, si el controlador evaluado cuenta con la característica identificada, se le asigna un puntaje de 1; de lo contrario, obtiene un valor de 0 por no disponer de dicho aspecto.

Tabla 8
Evaluación de controladores SDN en parámetros de seguridad y convergencia

Características/ Controladores	**ONOS**	**OpenDaylight**	**Rosemary**	**Ryu**
Crts_1	1	1	1	1
Crts_2	1	1	1	1
Crts_3	1	1	1	1
Crts_4	1	1	1	1
Crts_5	0	0	0	0
Crts_6	0	0	0	0
Crts_7	0	0	0	0
Total	4	4	4	4

Con una puntuación total de cuatro puntos igualada entre los controladores SDN opensource evaluados en la Tabla *8* se establece que cada uno de ellos presenta las funcionalidades fundamentales relacionadas al soporte del protocolo OpenFlow en conjunto con las funciones de acceso a los recursos de administración del plano de control del controlador. Además, los criterios del 5 al 7 recopilados Tabla 8 establecen que en términos de vulnerabilidades de seguridad los controladores no poseen mecanismos de mitigación ante ataques de denegación de servicio, de modo que es necesario tomar un criterio de selección más detallado que permita evaluar otros aspectos críticos para la implementación exitosa de la red SDN.

La selección específica del controlador SDN a ser implementado en el testbed se basa en el establecimiento de criterios utilizando la ISO/IEC 25010 que evalúa la calidad del producto de software respecto a parámetros de eficiencia de desempeño y compatibilidad que fueron explicados en el Anexo 1, de modo que tomando estos conceptos de selección en un inicio se requiere fijar distintas abreviaturas, debido a que cada una de ellas definirá su asociación a cada métrica establecida según las características seleccionadas de la ISO/IEC 25010. Este proceso es realizado con el objetivo de que se permita una mejor comprensión de la información presentada en siguientes secciones de este apartado, de modo que la Tabla 9 presenta la nomenclatura utilizada por las métricas en base a las características de eficiencia de desempeño, compatibilidad y capacidad de interacción.

Tabla 9
Abreviaturas de las métricas utilizadas para selección del controlado

Descripción	Abreviatura
Métrica de requerimientos de eficiencia de desempeño	MtCRED
Métrica de requerimientos de compatibilidad	MtCRC

- **Métricas de requerimientos de eficiencia de desempeño**

Las métricas de requerimientos de eficiencia de desempeño se centran en la evaluación del rendimiento de los controladores SDN. Esta evaluación implica identificar cómo los recursos informáticos del hardware que alojan al controlador ejecutan sus funciones en condiciones normales, en relación con los requisitos mínimos de uso de CPU, RAM y almacenamiento. Todos estos aspectos están detallados en las métricas de requerimientos recopiladas en la Tabla 10.

Tabla 10
Requerimientos de eficiencia de desempeño para seleccionar el controlador

Abreviatura	Descripción del requerimiento	Prioridad		
		Alta	Media	Baja
Métricas de utilización de recursos				
MtCRED1	El controlador SDN debe utilizar la CPU de manera eficiente, asegurando que no exceda el límite especificado de uso de recursos bajo condiciones normales de funcionamiento.	X		
MtCRED2	El controlador SDN debe gestionar eficientemente la memoria RAM, asegurando que no exista una sobreutilización de recursos.	X		
MtCRED3	El controlador debe utilizar eficientemente el almacenamiento disponible.		X	
Métricas de capacidad				
MtCRED4	El controlador debe ser capaz de programar el funcionamiento de los conmutadores SDN.	X		
MtCRED5	El controlador SDN puede manejar la estructura de las reglas de flujo para realizar procesos de conmutación de paquetes.	X		
MtCRED6	El controlador posee capacidad de programabilidad en sus funciones para soluciones de seguridad.	X		

- **Métricas de requerimientos de compatibilidad**

Las métricas de compatibilidad se centran en garantizar que el controlador pueda interactuar de manera efectiva con otras herramientas, como software de análisis de paquetes y herramientas de análisis de tráfico. Por otro lado, las métricas de interoperabilidad se enfocan en asegurar que las funciones del controlador puedan integrarse sin problemas utilizando la programabilidad que poseen, permitiendo compartir recursos del sistema y funcionar en conjunto para garantizar la operabilidad de la SDN. Estas métricas se presentan en la Tabla 11 y son esenciales para garantizar la flexibilidad, compatibilidad y funcionalidad de la red SDN en base al controlador seleccionado en el contexto del proyecto.

Tabla 11
Requerimientos de compatibilidad para seleccionar el controlador

Abreviatura	Descripción del requerimiento	Prioridad		
		Alta	Media	Baja
Métricas de coexistencia				
MtCRC1	El controlador permite integrar software complementario que complemente con sus funciones de operación y administración de la SDN.	X		
MtCRC2	El controlador habilita la integración de bibliotecas de software externo en la programación de sus funciones para gestionar la SDN.	X		
Métricas de interoperabilidad				
MtCRC3	El controlador SDN recopila la información analizada por un software externo y posteriormente analizarla.	X		
MtCRC4	El controlador SDN posibilita que la operabilidad de softwares externos a él no interrumpa con su funcionamiento.	X		

Una vez que se han identificado los criterios de selección del controlador, la Tabla 12 exhibe los criterios con un nivel de prioridad alto que son considerados los más adecuados para establecer el sistema de evaluación entre los controladores. En su estructura, cada métrica está identificada por su nomenclatura, la característica correspondiente de la norma ISO a la que pertenece, y su respectiva descripción.

Tabla 12
Requerimientos de alta prioridad para seleccionar el controlador

Abreviatura	Característica	Descripción
MtCRED1	Eficiencia de desempeño	El controlador SDN debe utilizar la CPU de manera eficiente, asegurando que no exceda el límite especificado de uso de recursos bajo condiciones normales de funcionamiento.
MtCRED2		El controlador SDN debe gestionar eficientemente la memoria RAM, asegurando que no exista una sobreutilización de recursos.
MtCRED4		El controlador debe ser capaz de programar el funcionamiento de los conmutadores SDN.

MtCRED5		El controlador SDN puede manejar la estructura de las reglas de flujo para realizar procesos de conmutación de paquetes.
MtCRED6		El controlador posee capacidad de programabilidad en sus funciones para soluciones de seguridad.
MtCRC1	Compatibilidad	El controlador permite integrar software complementario que complemente con sus funciones de operación y administración de la SDN.
MtCRC2		El controlador habilita la integración de bibliotecas de software externo en la programación de sus funciones para gestionar la SDN.
MtCRC3		El controlador SDN recopila la información analizada por un software externo para posteriormente analizarla.
MtCRC4		El controlador SDN posibilita que la operabilidad de softwares externos a él no interrumpa con su funcionamiento .

Al considerar la ponderación de cada métrica en relación con las características seleccionadas y tener en cuenta que el controlador seleccionado será el encargado de gestionar la red SDN del testbed y a la vez posea capacidades que permitan integrar soluciones de seguridad. Esto implicaría que la selección será en base al controlador SDN que posea las funciones de programabilidad en la gestión de la SDN y a la vez permita la integración de software para complementar su funcionamiento.

Con base en este criterio de selección, la Tabla 13 muestra la ponderación obtenida, donde el valor de 1 indica que el software cumple con el requerimiento, mientras que el 0 indica lo contrario. Esto permite obtener una tabla de evaluación de los controladores, en la cual el controlador que obtenga la puntuación más alta será seleccionado en el contexto de desarrollo de este proyecto.

Tabla 13
Evaluación de controladores SDN

Características/ Controladores	**ONOS**	**OpenDaylight**	**Rosemary**	**Ryu**
Crts_1-7	4	4	4	4
MtCRED1	1	1	1	1

MtCRED2	1	1	1	1
MtCRED4	1	1	1	1
MtCRED5	1	1	1	1
MtCRED6	1	0	0	1
MtCRC1	1	1	0	1
MtCRC2	0	0	0	1
MtCRC3	0	0	0	1
MtCRC4	1	1	0	1
Total	11	10	8	13

De acuerdo con los resultados obtenidos en la Tabla 13, el controlador Ryu se destaca como el controlador SDN más adecuado para el testbed, cumpliendo con los requerimientos clave de eficiencia de desempeño y compatibilidad. Esto se debe a que las funciones del controlador aseguran una utilización eficiente de la CPU y RAM (MtCRED1 y MtCRED2), a la vez que puede establecer las funciones de los conmutadores SDN mediante programación (MtCRED4). Además, permite la integración de software complementario a sus funciones (MtCRC1), bibliotecas externas dentro del código de su funcionamiento (MtCRC2), y garantizar que la operabilidad del software externo no interrumpa su funcionamiento (MtCRC4). Estas capacidades, junto con su habilidad para manejar y analizar información de software externo (MtCRC3), hacen de Ryu la opción más competente, con un puntaje total de 13, superando a los controladores evaluados.

c) Plano de aplicación

En la capa de aplicación, el controlador Ryu requiere instalar herramientas correspondientes al módulo de gestión web flowmanager master y el API REST, los cuales posibilitan la administración de acuerdo con las necesidades de los usuarios en la SDN. Estas necesidades pueden variar desde el descubrimiento de nuevas redes hasta la implementación de servidores o la gestión de interfaces bloqueadas, entre otras funciones que puedan surgir.

3.2.5 Implementación del controlador SDN

La implementación de la red SDN comienza principalmente en el plano de control, ya que este elemento de red de la arquitectura aloja todas las funciones de control y gestión de la red SDN. La selección de Ryu como controlador se justifica en la sección anterior del presente capítulo, utilizándolo como la opción más adecuada para gestionar las funciones de la red SDN y a la vez identificar las funciones respecto a la seguridad que posee ante ataques de denegación de servicio.

El controlador corresponde a una distribución de software de código abierto que basa sus funciones en un framework[8] para gestión y administración de redes SDN. Este controlador se distribuye bajo la licencia Apache 2.0 y todas las aplicaciones destinadas a redes SDN se basan en archivos de programación escritos en Python (Ryu, 2014).

Además, el controlador Ryu es compatible con el protocolo OpenFlow desde la versión 1.0 hasta la 1.4 para administración de conmutadores SDN, lo que permite la implementación de diversas aplicaciones para gestionar eventos de red y responder a los cambios que pueden suscitarse mediante la instalación de nuevas reglas de flujos en los conmutadores. Además, (RYU project team, 2020) presenta otras características que posee el controlador siendo estos los siguientes:

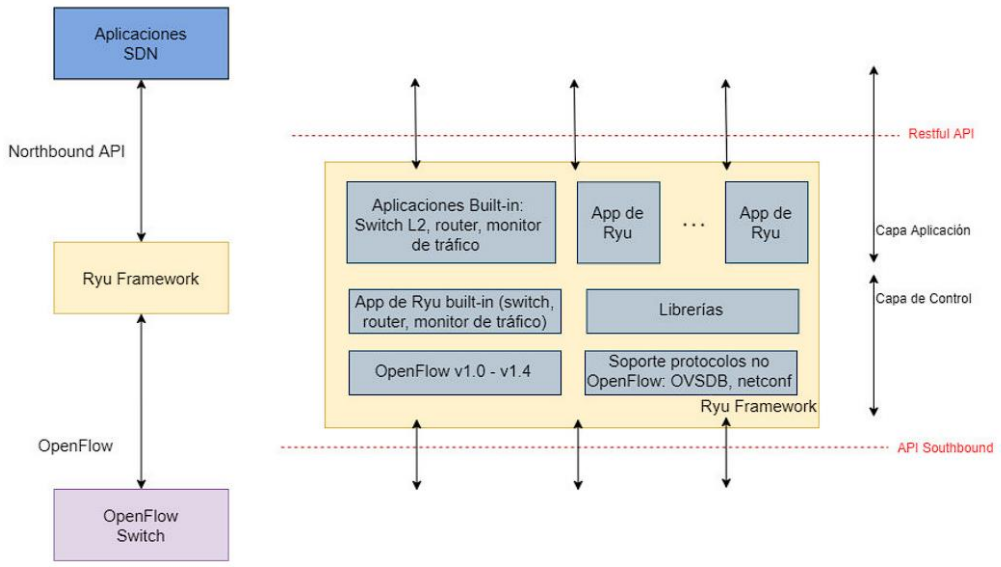
- Capacidad para analizar paquetes entrantes y dirigirlos a la red.
- Facilita la generación y transmisión de paquetes OpenFlow.
- Incluye diversas bibliotecas para respaldar las distintas operaciones de procesamiento de paquetes.
- Viene equipado con aplicaciones como monitor de tráfico, enrutador, cortafuegos e IDS.

[8] Framework: Estructura conceptual y técnica que proporciona soporte y funcionalidades comunes para facilitar el desarrollo de software.

Tomando la premisa de que Ryu proporciona un entorno de desarrollo a través de un framework, la Figura 13 presenta la arquitectura del controlador, destacando su compatibilidad con interfaces Northbound RESTful para las operaciones OpenFlow (Ryu, 2014). No obstante, las aplicaciones internas de Ryu pueden funcionar sin necesidad de utilizar esta interfaz, gracias a la presencia del gestor de aplicaciones de Ryu (ryu-manager). Por lo tanto, esta arquitectura indica que la implementación de las entradas de flujo deseadas por el usuario se realizará a través de las interfaces Southbound hacia los conmutadores SDN.

Figura 13

Arquitectura del controlador Ryu



Nota, Adaptado de (Ryu, 2014)

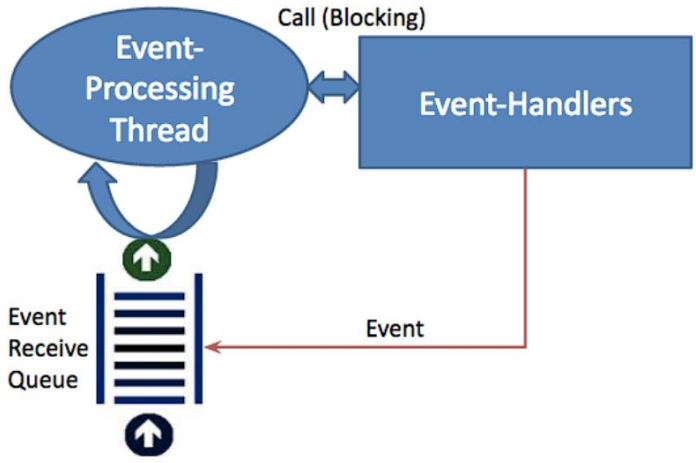
La arquitectura del controlador establece que la forma en que se desarrollan y ejecutan las aplicaciones se basa en eventos que suceden en la red. Esto significa que las acciones que realizan las aplicaciones están vinculadas a sucesos específicos. Por ejemplo, cuando ocurre un evento de red, como la recepción de un paquete, las aplicaciones pueden responder realizando acciones como la conmutación, el enrutamiento

o la seguridad. Cada una de estas aplicaciones puede considerarse como una entidad independiente dentro del sistema de Ryu.

Cada aplicación de Ryu cuenta con una cola FIFO para recibir eventos y mantener su orden. La Figura 14 presenta la arquitectura de una aplicación ejecutada en el controlador, la cual representa que para cada aplicación hay un hilo dedicado que se encarga de procesar los eventos en la cola. Dentro de este hilo, hay un bucle principal que extrae los eventos de la cola y llama a un controlador de eventos correspondiente para su procesamiento (Rao S, 2024).

Figura 14

Arquitectura funcional de la aplicación Ryu

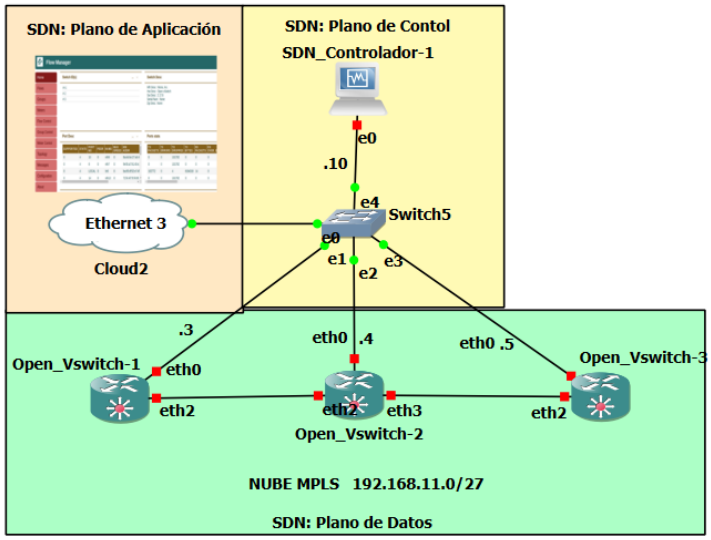


Finalmente, las aplicaciones desarrolladas para la gestión de redes SDN mediante el controlador SDN se realiza a partir de una sintaxis de ejecución correspondiente a `ryu-manager app_lists`. En esta sintaxis establece que, `ryu-manager` se refiere a la herramienta principal de gestión de aplicaciones de Ryu, que facilita la carga y ejecución de las aplicaciones creadas por los usuarios, mientras que `app_lists` se utiliza para listar las aplicaciones disponibles y gestionadas por el controlador. De esta manera, los usuarios pueden identificar las aplicaciones instaladas y verificar su estado de ejecución en el entorno de Ryu.

3.2.6 Despliegue de la red híbrida SDN

El proceso de despliegue de la red híbrida empieza con la asociación de los conmutadores SDN comienza al determinar la conexión de cada uno de los puertos de los Open Vswitch dentro de la plataforma de GNS3. Este procedimiento tiene como propósito establecer los enlaces necesarios para interconectar los switches con el controlador, lo que permite alcanzar la gestión centralizada de estos componentes que conforman la red. La Figura 15 representa los enlaces utilizados para obtener la topología de red definida por software propuesta.

Figura 15
Establecimiento de enlaces de la red SDN



Establecida la topología de la red SDN, es necesario reconocer los puertos que poseen conexión directa hacia el controlador, debido a que estos requerirán la asignación de una dirección IPv4 de acuerdo con la Tabla 6 y especificaciones de levantamiento de interfaces especificadas en el Anexo 6. Esto es realizado con el objetivo de que cada uno de los puertos de los conmutadores SDN sean enlazados, establezcan la comunicación con el controlador mediante el protocolo OpenFlow mediante las sentencias especificadas en la **Tabla *14***.

Tabla 14
Comandos que permiten la asociación de los switches SDN al controlador

Sentencia	Descripción
`ovs-vsctl del-port br0 eth0`	Elimina un puerto específico del puente del switch con el objetivo de separar el tráfico Openflow del puerto y permitir la conexión al controlador.
`ovs-vsctl set-controller br0 tcp:ip_contolador:6633`	Establece la comunicación hacia un controlador SDN que utiliza el protocolo OpenFlow mediante conexiones TCP hacia su dirección IP mediante el puerto 6633.
`ovs-vsctl set bridge br0 other-config:datapath-id=000000000000000#`	Establece un identificador único al switch dentro de la red SDN, de modo que el # corresponde al número que puede ser asignado a cada switch.
`ovs-vsctl set bridge br0 protocols=OpenFlow13`	Establece que el switch utilizará la comunicación con el controlador mediante el protocolo OpenFlow 1.3
`ovs-vsctl set-manager ptcp:6632`	Configura al OpenVswitch para que acepte conexiones de control remoto a través del puerto TCP 6632.

Nota. Adaptado de (Open vSwitch, 2024)

Identificadas las configuraciones que permiten la asociación de los conmutadores SDN hacia el controlador, la Figura 16 registra los comandos establecidos en el Open Vswitch-1 para lograr este objetivo. Estas configuraciones serán replicadas en los demás switches, variando únicamente el apartado de hardware según el número del switch en el que se realicen las configuraciones.

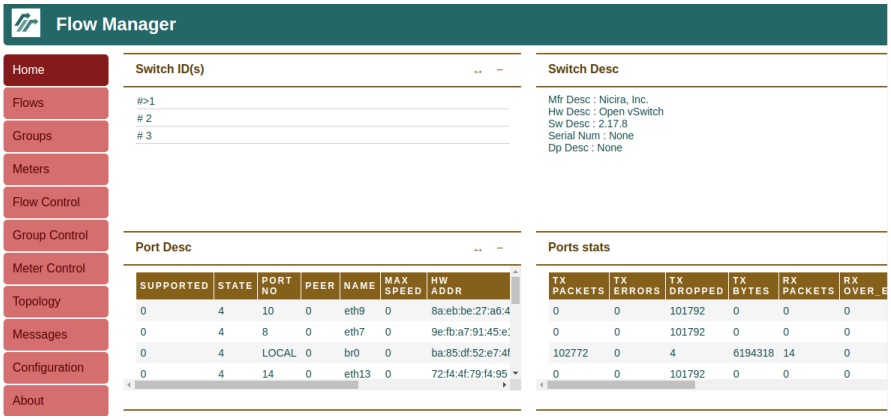
Figura 16
Configuraciones de enlace del Open Vswitch-1 al controlador

```
/ # ovs-vsctl del-port br0 eth0
/ # ovs-vsctl set-controller br0 tcp:192.168.11.10:6633
/ # ovs-vsctl set bridge br0 other-config:hwaddr=00:00:00:00:00:01
```

Una vez que se han aplicado todas las configuraciones en cada conmutador de la red SDN, el controlador inicia con el registro de cada switch para poder ser visualizados en la interfaz gráfica de administración web como se evidencia en la Figura 17.

Figura 17

Registro de los conmutadores SDN en el controlador

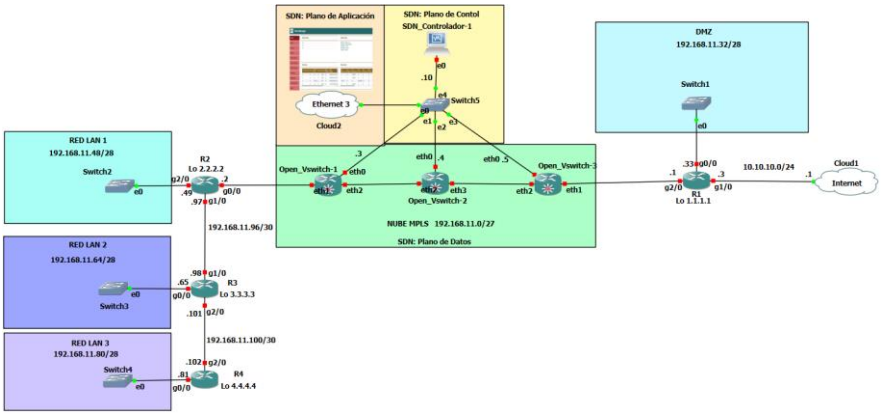


El siguiente proceso de despliegue de la red híbrida corresponde en establecer un entorno de red que combina la funcionalidad de los elementos de la red definida por software, enfocada en la centralización y control de acciones en la red, junto con la estabilidad e interoperabilidad que constituyen la infraestructura de redes tradicionales basadas en hardware.

La Figura 18 proporciona una visualización de los enlaces que permiten las conexiones entre la red SDN y una infraestructura de red tradicional. Esta infraestructura está compuesta por 4 enrutadores Cisco c7200, en donde la información de cada uno de los enlaces de interconexión representados en la topología es de acuerdo con la asignación de direccionamiento IPv4 detallada en la Tabla 6.

Figura 18

Red híbrida SDN de los escenarios del testbed



Una vez establecidas las interfaces y el direccionamiento de cada enrutador, el siguiente paso para lograr la convergencia total de la red, implica asegurar que cada enrutador tenga asignada la dirección IPv4 correcta en sus interfaces, según lo especificado en la Tabla 6. Además, deben ser configurados de acuerdo con las especificaciones presentadas en la Tabla 63, que detalla la implementación del protocolo de enrutamiento OSPF en los routers. La Figura 19 muestra el proceso de asignación de direcciones IPv4 y configuración del enrutamiento OSPF en el router R1. Estas configuraciones se replicarán en los tres routers restantes utilizando la información proporcionada en las Tabla 6 y Tabla 63, previamente mencionadas.

Figura 19

Configuración de interfaces y enrutamiento del enrutador R1

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int lo0
R1(config-if)#ip add 1.1.1.1 255.255.255.255
R1(config-if)#no sh
R1(config-if)#ip ospf 1 area 0
R1(config-if)#exit
R1(config)#int g0/0
R1(config-if)#ip add 192.168.11.33 255.255.255.240
R1(config-if)#no sh
R1(config-if)#ip ospf 1 area 0
R1(config-if)#int g2/0
R1(config-if)#ip add 192.168.11.1 255.255.255.224
R1(config-if)#no sh
R1(config-if)#ip ospf 1 area 0
R1(config-if)#exit
R1(config)#int g1/0
R1(config-if)#ip add dhcp
R1(config-if)#no sh
R1(config-if)#ip ospf 1 area 0
R1(config-if)#exit
R1(config)#end
```

3.2.7 *Calidad de servicio en la red híbrida SDN*

La calidad de servicio (QoS) en una red híbrida SDN es fundamental para garantizar un rendimiento óptimo y una gestión eficiente del tráfico de red. En esta sección se presenta todo el proceso realizado para implementar calidad de servicio dentro de los enrutadores tradicionales que forman parte de la red híbrida SDN y a la vez la aplicación de QoS en el entorno SDN.

a) Calidad de servicio en las redes tradicionales del testbed

Una vez establecido el escenario de la red híbrida SDN, uno de los siguientes pasos en el desarrollo de la topología es configurar la calidad de servicio (QoS) utilizando el modelo de Diferenciación de Servicios (DiffServ) para asignar niveles de prioridad a cada uno de los servidores que prestan servicio a los hosts dentro de la red.

La premisa de QoS, que se basa en la aplicabilidad de la frontera de confianza, indica que esta esté lo más cerca posible de la fuente del tráfico. De esta manera, al aplicarse la calidad de servicio los demás dispositivos de la red pueden respetar y confiar en el marcado de los paquetes, de acuerdo con los Diffserv Code Point (DSCP), los cuales son responsables de la clasificación, gestión y aplicación de QoS dentro de las redes IP.

En base a los conceptos de QoS establecidos anteriormente, uno de los primeros procesos por realizar corresponde en establecer la tabla de priorización del tráfico que pasa por la red, de acuerdo con los 4 servicios correspondientes a VoIP, VoD, FTP y Hosting, en base a la información de etiquetado DSCP otorgado por la tabla de clasificación del tráfico de Cisco para QoS (Cisco, 2024) .

La Tabla 15 proporciona información detallada sobre las prioridades y las aplicaciones correspondientes. Por ejemplo, se establece que el tráfico de voz, que utiliza el protocolo RTP en el rango de puertos UDP de 10000 a 20000, se marca con la etiqueta DSCP EF (Expedited Forwarding), lo que indica una prioridad crítica. Del mismo modo, la señalización SIP se marca con la etiqueta DSCP AF31 (Assured Forwarding 31), mientras que el tráfico VoD es marcado con la etiqueta DSCP AF43 (Assured Forwarding 43) y el tráfico FTP es marcado respectivamente con la etiqueta DSCP AF33 (Assured Forwarding 33), de modo que estos dos marcajes indican prioridades altas en el manejo de este tipo de datos en la red. Las aplicaciones web, que utilizan el protocolo HTTP en

el puerto TCP 80, se marcan con la etiqueta DSCP AF23 (Assured Forwarding 23) para prioridades medias.

Tabla 15
Tabla de priorización del tráfico para aplicación de QoS

Prioridad	Aplicación	Protocolo	TCP/UDP	Puertos	DSCP
Critico	VoIP	RTP	UDP	10000-20000	EF
	Señalización	SIP	TCP	5060	AF31
	VoD	TCP	TCP	8096	AF43
Alta	Transferencia de archivos	FTP	TCP	20,21	AF33
Media	Web	HTTP	TCP	80	AF23

Establecida la tabla de priorización del tráfico, su implementación es realizada mediante la MQC[9] que poseen los enrutadores de borde de la frontera de confianza, como lo son los dispositivos Cisco en la topología de red. Esta interfaz establece una serie de sintaxis de comandos que permiten la aplicación de la calidad de servicio dentro de los dispositivos de red. El MQC establece de tres fases que serán aplicadas en el enrutador R1 que forma parte de la frontera de confianza de la topología y cumple con los procesos correspondientes a:

- **Definición de clases de servicio**

Fase que se basa en la creación de listas de control de acceso (ACLs) con el propósito de clasificar el tráfico de red que pasa por el dispositivo y a la vez llevar el número de paquetes marcados de acuerdo con el DSCP asignado, para la calidad de servicio. La Figura 24 muestra la configuración implementada en el router R1 de la

[9] MQC: Interfaz de línea de comandos de calidad de servicio modular.

topología presentada en la Figura 18, de acuerdo con la información recopilada en la Tabla 15 que define los protocolos y puertos utilizados por cada servicio.

Figura 20

Establecimiento de ACLs para clasificación del tráfico en el router R1

```
R1(config)##Marcar VoIP con EF
R1(config)#access-list 101 permit ip any any dscp ef
R1(config)#
R1(config)##Marcar SIP con AF31
R1(config)#access-list 102 permit ip any any dscp af31
R1(config)#
R1(config)##Clasificar y marcar VoD con AF43
R1(config)#access-list 103 permit ip any any dscp af43
R1(config)#access-list 103 permit tcp any eq 8096 any
R1(config)#
R1(config)##Marcar FTP con AF33
R1(config)#access-list 104 permit ip any any dscp af33
R1(config)#
R1(config)##Marcar HTTP con AF23
R1(config)#access-list 105 permit ip any any dscp af23
R1(config)#
R1(config)##Clasificar puertos RTP
R1(config)#access-list 106 permit udp any range 10000 20000 any
```

El proceso siguiente consiste en crear clases utilizando el comando class-map y luego asociar las ACL correspondientes mediante el comando match access-group. Esto da lugar a la creación de dos clases para cada servicio: las clases con la designación "in" clasificarán el tráfico, mientras que las clases con la designación "out" serán responsables de establecer el marcado de cada paquete IP de acuerdo con su DSCP establecido. Estas configuraciones se ilustran en la Figura 21 y su implementación es crucial para avanzar con la configuración de las políticas de calidad de servicio (QoS).

Figura 21

Creación de clases y asociación de ACLs

```
R1(config)##////Clases//////
R1(config)#
R1(config)##VoIP
R1(config)#class-map match-any voz
R1(config-cmap)#match protocol rtp audio
R1(config-cmap)#match access-group 106
R1(config-cmap)#exit
R1(config)#
R1(config)#class-map match-all voz-out
R1(config-cmap)#match access-group 101
R1(config-cmap)#exit
R1(config)#
R1(config)##sip
R1(config)#class-map match-all sip
R1(config-cmap)#match protocol sip
R1(config-cmap)#exit
R1(config)#
R1(config)#class-map match-all sip-out
R1(config-cmap)#match access-group 102
R1(config-cmap)#exit
R1(config)#
R1(config)#class-map match-all vod-out
R1(config-cmap)#match access-group 103
R1(config-cmap)#exit
R1(config)#
R1(config)##ftp
R1(config)#class-map match-any ftp
R1(config-cmap)#match protocol ftp
R1(config-cmap)#exit
R1(config)#
R1(config)#class-map match-all ftp-out
R1(config-cmap)#match access-group 104
R1(config-cmap)#exit
R1(config)#
R1(config)##web
R1(config)#class-map match-any web
R1(config-cmap)#match protocol http
R1(config-cmap)#exit
R1(config)#
R1(config)#class-map match-all web-out
R1(config-cmap)#match access-group 105
R1(config-cmap)#exit
```

- **Establecimiento de políticas para las clases**

El segundo paso consiste en establecer las políticas de tráfico y asociarlas con las clases previamente definidas. En este proceso, se establece una única política, denominada 'testbed_policy', que abarca un conjunto específico de clases y define las acciones correspondientes.

La Figura 22 muestra el establecimiento de una política de marcaje denominada testbed_mark, la cual posee la función de definir clases de tráfico, como VoIP, SIP, VoD, FTP y web, asignándoles etiquetas DSCP según la prioridad de cada aplicación, como EF, CS3, AF43, AF33 y AF23 respectivamente, basándose en la información de la Tabla 17. Estas etiquetas DSCP se aplican a través del comando `set ip dscp` dentro de una política de marcaje `policy-map`, garantizando así la priorización adecuada del tráfico y la gestión eficiente de la calidad de servicio en la red híbrida SDN.

Figura 22
Establecimiento de políticas de marcaje

```
R1(config)#policy-map testbed_mark
R1(config-pmap)#
R1(config-pmap)#class voz
R1(config-pmap-c)#set ip dscp ef
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class sip
R1(config-pmap-c)#set ip dscp af31
R1(config-pmap-c)#exit
R1(config-pmap)#class vod
R1(config-pmap-c)#set ip dscp af43
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class ftp
R1(config-pmap-c)#set ip dscp af33
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class web
R1(config-pmap-c)#set ip dscp af23
R1(config-pmap-c)#exit
```

La siguiente política corresponde en establecer el tamaño del ancho de banda del encolamiento manejado por el dispositivo para cada servicio, que en este caso será en base a CBWFQ[10], la cual define las clases de tráfico según criterios de coincidencia de protocolos, ACLs e interfaces de entrada. Todo el apartado del proceso de cálculo del ancho de banda para establecer las políticas se muestra en el Anexo 5, de modo que la

[10] CBWFQ: Colas justas ponderadas basadas en la clase

Figura 28 establece las acciones de reserva de ancho de banda de acuerdo con los servicios que posee la topología.

Figura 23
Establecimiento de ancho de banda de cada servicio

```
R1(config)#policy-map testbed_policy
R1(config-pmap)#
R1(config-pmap)#class voz-out
R1(config-pmap-c)#priority percent 38
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class sip-out
R1(config-pmap-c)#bandwidth percent 7
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class vod-out
R1(config-pmap-c)#priority percent 7
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class ftp-out
R1(config-pmap-c)#priority percent 3
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class web-out
R1(config-pmap-c)#priority percent 19
R1(config-pmap-c)#exit
R1(config-pmap)#
R1(config-pmap)#class class-default
R1(config-pmap-c)#fair-queue
R1(config-pmap-c)#exit
R1(config-pmap)#end
```

- **Aplicación de políticas de QoS**

Por último, la Figura 24 representa la implementación de la política de tráfico, abordando tanto el marcaje de paquetes como la teoría de colas, considerando la dirección del tráfico en la red. En este escenario, el objetivo principal es asegurar la entrega de calidad de servicio (QoS) y distinguir los servicios de los paquetes que se encaminan hacia la zona desmilitarizada (DMZ) cuando son solicitados por los usuarios.

La interfaz G2/0 del router R1 se asigna como el enlace de salida hacia la DMZ, ya que se busca gestionar los paquetes provenientes de las redes LAN según su prioridad y ancho de banda asignados. Esto se ejecuta con el propósito de evitar la congestión de la red y garantizar la administración de servicios diferenciados mediante políticas de QoS. En contraste, la interfaz G1/0 se configura como la entrada, dado que por esta interfaz circulan los paquetes generados en la red híbrida SDN.

Figura 24
Aplicación de políticas de marcaje y asignación de ancho de banda

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int g0/0
R1(config-if)#service-policy input testbed_mark
R1(config-if)#exit
R1(config)#
R1(config)#int g2/0
R1(config-if)#service-policy output testbed_policy
R1(config-if)#exit
R1(config)#end
```

b) Calidad de servicio en la red SDN del testbed

La red SDN que conforman la parte central del testbed permite la gestión de la calidad de servicio mediante el controlador, el cual es el encargado de distribuir las configuraciones de QoS a todos los dispositivos de la red. Esta capacidad de administración centralizada facilita y optimiza los procesos de configuración de la red, especialmente en escenarios que requieren configuraciones por salto, como es el caso de DiffServ que fue aplicado en los enrutadores tradicionales que conforman la red.

El controlador SDN administra la QoS a través de APIs, siendo una de las arquitecturas más comunes la RESTful. Esta arquitectura, basada en la configuración cliente-servidor, la cual ofrece una forma sencilla de gestionar y aplicar políticas de QoS mediante solicitudes para crear, leer, actualizar y eliminar. Estas solicitudes se describen en la Tabla 16, la cual identifica los métodos HTTP utilizados para realizar estas acciones.

Tabla 16
Métodos o verbos HTTP

Método HTTP	Descripción
GET	Permite recuperar un recurso.
POST	Se utiliza para enviar datos al servidor y crear registros en la API.
PUT	Actualiza un recurso completo.
PATCH	Actualiza parcialmente un recurso.
DELETE	Elimina un recurso.

Nota. Adoptado de Oracle (2024).

Las respuestas se determinan mediante códigos de estado, que indican si la solicitud fue exitosa, si hubo algún error o si se necesitan acciones adicionales como se identifican en la Tabla 17.

Tabla 17
Códigos de estado en respuestas HTTP.

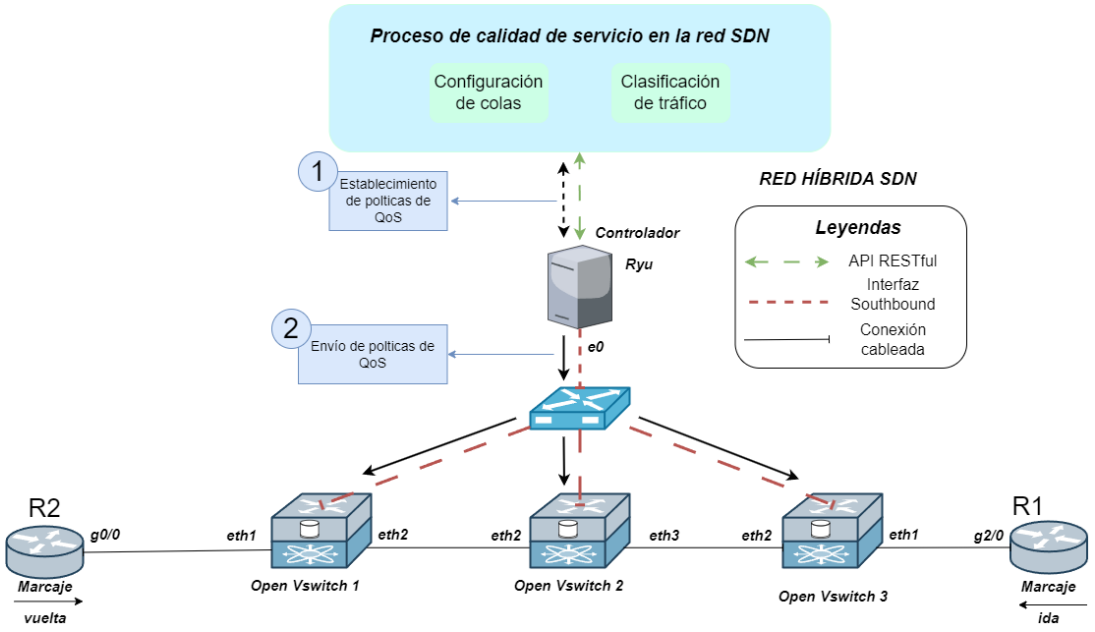
Código de estado	Descripción
100 – 199	Son mensajes informativos.
200 – 299	Son respuestas satisfactorias, indican el éxito a las peticiones realizadas.

300 – 399	Proporciona información de redireccionamiento de recursos.
400 – 499	Indica errores por parte del cliente (solicitud a recursos inexistentes, sintaxis incorrecta, etc).
500 – 599	Indica errores en el servidor.

Nota. Adoptado de Postman (2024).

Una vez que se han identificado los mensajes que intercambian el controlador y la API RSTful para establecer la calidad de servicio dentro de la red SDN, el siguiente paso en la implementación corresponde en analizar como las interfaces de la arquitectura SDN permiten la implementación de QoS en los conmutadores. La Figura 25 presenta los componentes involucrados en el proceso, en donde la API RESTful es la encargada de administrar los procesos relacionados a la configuración de colas y clasificación del tráfico, mientras que la interfaz Southbound de la arquitectura SDN es la encargada de enviar las configuraciones establecidas en el controlador hacia los conmutadores SDN de la red.

Figura 25
Designación de QoS en la infraestructura SDN



El establecimiento de priorización de los servicios para el tráfico que requiere de calidad de servicio en la SDN se mantiene de acuerdo con la fundamentación de niveles

de prioridad establecidos en la Tabla 15. Para la aplicación de QoS dentro de la SDN en un inicio será necesario establecer el encolamiento en las interfaces de los conmutadores, de acuerdo con las configuraciones de establecimiento de porcentajes de ancho de banda otorgados a cada servicio que pasa en las redes tradicionales, de modo que los valores de las colas se mantendrán iguales que en la sección anterior.

Para conseguir establecer las colas en los conmutadores será necesario utilizar las sentencias de los conmutadores OpenVSwitch para definir y configurar las políticas de calidad de servicio. La Tabla 18 incluye las sintaxis de de configuración de QoS específicas relacionadas con el establecimiento de colas con sus respectivas tasas mínimas y máximas de ancho de banda.

Tabla 18
Sentencias de configuración de colas en OpenVswitch

Parámetro	Función
ovs-vsctl set port name qos=@newqos	Establece la QoS en un puerto (name) y asigna la configuración @newqos. El valor name se debe reemplazar con el nombre del puerto que se desea configurar
--id=@newqos create qos	Crea una nueva configuración de QoS con el identificador @newqos.
type=linux-htb	Especifica el tipo de QoS, en este caso linux-htb (Hierarchical Token Bucket[11]).
other-config:max-rate=100000000	Establece la tasa máxima de la QoS a 100000000 bps (100 Mbps).
queues=#=@q0,#=@q#	Define las colas y sus identificadores correspondientes. El valor # es un número asignado que identifica a la cola creada.
--id=@q# create queue	Crea una cola con el identificador @q#.
other-config:max-rate=100000000	Establece la tasa máxima de la cola @q# a 100000000 bps (100 Mbps).
other-config:min-rate= #	Define la cantidad mínima de ancho de banda que se garantiza a una cola en particular. El valor

[11] Hierarchical Token Bucket: Algoritmo de control de tráfico utilizado en redes para gestionar y priorizar el ancho de banda. Funciona mediante la asignación de tasas mínimas y máximas de transferencia de datos a diferentes colas, permitiendo una distribución equitativa del ancho de banda entre distintos flujos de tráfico (Open vSwitch, 2024).

	# representa la tasa mínima en bits por segundo (bps).
ovs-vsctl set port name qos=id_qos	Asigna una configuración de QoS específica al puerto mencionado. El valor d_qos debe ser reemplazado con el identificador de la configuración de QoS que se ha creado previamente.

Nota. Adaptado de (OpenVswitch,2023)

Identificadas las sentencias que permite la configuración de las colas en los conmutadores SDN, el siguiente proceso consta en establecer el encolamiento para el tráfico de cada servicio del testbed en los OpenVswitch. Para eso se establece la Tabla 19 que identifica el número de colas a ser creadas en cada conmutador, la cantidad máxima y mínima de ancho de banda a utilizar, además de su relación con el tipo de tráfico al cual se asociara cada cola.

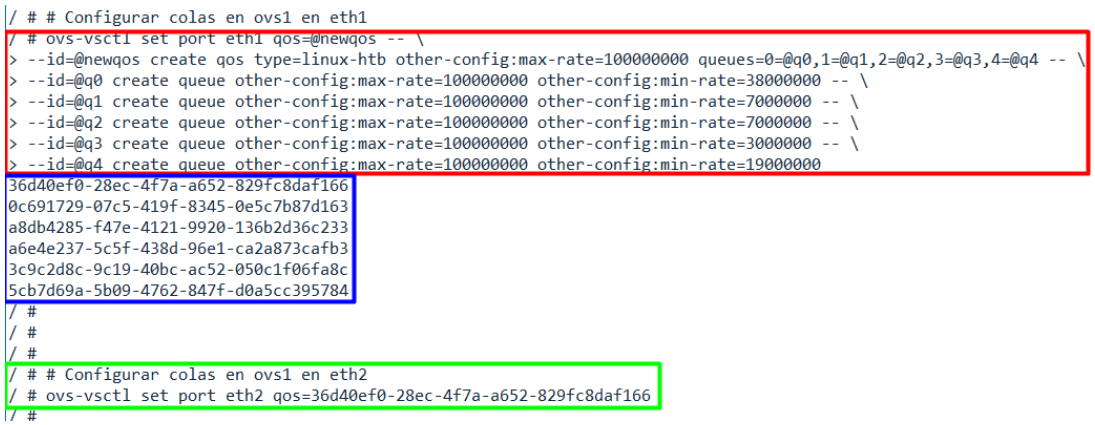
Tabla 19
Establecimiento de colas para cada tipo de tráfico

Cola (#)	Max-rate (bps)	Min-rate (bps)	Protocolo-Servicio	DSCP
0	100000000	38000000	RTP – VoIP	EF
1	100000000	7000000	SIP – VoIP	AF31
2	100000000	7000000	HTTP -VoD	AF43
3	100000000	3000000	FTP -FTP	AF33
4	100000000	1900000	HTTP – Web	AF23

Una vez identificados los parámetros de cada una de las colas y el tipo de tráfico a los cuales serán asignados, el siguiente paso corresponde en configurar las interfaces de los conmutadores SDN, los cuales son identificados en la Figura 25 con el nombre de eth. La Figura 26 muestra en el recuadro rojo la configuración de las colas en el puerto eth1 del OpenVswitch-1, creando una política de QoS con cinco colas, cada una con las tasas mínimas y máximas específicas para gestionar el tráfico de red utilizando el algoritmo de control linux-htb.

En la misma sección de la imagen, la información enmarcada en el recuadro azul contiene a los identificadores únicos (UUID) generados para las colas y la configuración de QoS, necesarios para referenciar estas configuraciones en otros comandos. Finalmente, el cuadro verde muestra el comando para aplicar la configuración de QoS creada (UUID del cuadro azul) al puerto eth2 de1 conmutador, asegurando que este puerto también utilice las mismas reglas de tráfico definidas en la política de QoS.

Figura 26
Configuración de colas en el conmutador OpenVswitch1



Nota. Todas las configuraciones se alojan en el repositorio de Github correspondiente al Anexo 7

Las mismas configuraciones de colas y políticas de calidad de servicio (QoS) establecidas para el puerto eth1 del OpenVswitch1 se aplicarán a las interfaces correspondientes identificadas en la Figura 25 de los conmutadores OpenVswitch 2 y 3. Esto asegura una gestión uniforme del tráfico en múltiples puertos y conmutadores dentro de la red, permitiendo una distribución equitativa del ancho de banda y priorización del tráfico crítico de manera consistente en toda la infraestructura de red.

Establecido el encolamiento en cada uno de los conmutadores SDN, la siguiente fase consta en establecer el marcaje de vuelta en los routers que conforman las redes LAN de la topología del escenario general del testbed. Esto es realizado debido a que al realizar un marcaje de tráfico de ida como de vuelta, se asegura que todos los paquetes

relacionados con una sesión específica sean tratados de manera uniforme a lo largo de toda la red. En este caso, las aplicaciones de VoIP, VoD y FTP que se alojan en el testbed requieren este nivel de tratamiento debido a que requieren una baja latencia y alta prioridad en ambos sentidos.

Tomando la premisa que en la sección 3.2.7 existe un apartado denominado "Calidad de servicio en las redes tradicionales del testbed", donde se presentan todas las configuraciones relacionadas con la implementación de QoS en los routers tradicionales, esta parte se centra en presentar el establecimiento de las ACLs que permiten que el tráfico generado por cada LAN sea marcado de manera uniforme, manteniendo los mismos principios de configuración y asignación de políticas mencionados en la sección anterior. Esto asegura que tanto el tráfico de ida como el de vuelta que pasa por la SDN sea priorizado y gestionado adecuadamente por las colas previamente creadas en cada uno de los puertos de los conmutadores.

La Figura 27 presenta las configuraciones de marcaje de vuelta que garantizan una clasificación y priorización adecuada del tráfico de red para asegurar una calidad de servicio óptima para las aplicaciones siguientes aplicaciones: VoIP se marca con EF para alta prioridad y baja latencia, SIP con CS3 para la señalización, VoD con AF43 para un servicio garantizado, y FTP y HTTP con AF33 y AF23 respectivamente para un servicio aceptable. Además, se clasifica el tráfico de puertos RTP para gestionar aplicaciones en tiempo real, mejorando la administración de la red y la experiencia del usuario final.

Figura 27

Listas de control de acceso para QoS de trafico de ida de cada LAN

```
R2(config)##Marcar VoIP con EF
R2(config)#access-list 101 permit ip any any dscp ef
R2(config)#
R2(config)##Marcar SIP con CS3
R2(config)#access-list 102 permit ip any any dscp af31
R2(config)#access-list 102 permit udp any any eq 5060
R2(config)#
R2(config)##Clasificar y marcar VoD con AF43
R2(config)#access-list 103 permit ip any any dscp af43
R2(config)#access-list 103 permit tcp any any eq 8096
R2(config)#
R2(config)##Marcar FTP con AF33
R2(config)#access-list 104 permit ip any any dscp af33
R2(config)#
R2(config)##Marcar HTTP con AF23
R2(config)#access-list 105 permit ip any any dscp af23
R2(config)#
R2(config)##Clasificar puertos RTP
R2(config)#access-list 106 permit udp any any range 10000 20000
```

Nota. Todas las configuraciones se alojan en el repositorio de Github correspondiente al Anexo 7

En lo que respecta al controlador SDN, este deberá ejecutar tres aplicaciones denominadas: `ryu-manager ryu.app.rest_qos ryu.app.qos_simple_switch_13 ryu.app.rest_conf_switch flowmanager-master/flowmanager.py`. En particular, la aplicación ryu.app.rest_qos proporciona una API REST para gestionar la calidad de servicio (QoS), ryu.app.qos_simple_switch_13 implementa un conmutador simple compatible con QoS utilizando el protocolo OpenFlow 1.3, ryu.app.rest_conf_switch permite la configuración de switches mediante una API REST, y flowmanager-master/flowmanager.py es un script personalizado para gestionar flujos en la red mediante acceso web. Juntas, estas aplicaciones habilitan y configuran la QoS en los switches, así como la administración de flujos de red a través de interfaces RESTful.

Posteriormente para aplicar las configuraciones de QoS de forma automática es necesario la creación de un archivo de comandos de lenguaje de secuencia denominado `config_qos.sh`, el cual será ejecutado una vez que el controlador haya iniciado las funcionalidades de los conmutadores SDN mediante las aplicaciones mencionadas anteriormente.

En las primeras líneas del archivo se alojan las funcionalidades que configuran las direcciones del protocolo OVSDB (Open vSwitch Database) para cada switch en la red SDN utilizando la herramienta curl para enviar solicitudes HTTP. Cada comando curl -X PUT envía una solicitud PUT a la API RESTful del controlador Ryu, especificando la dirección IP y el puerto de OVSDB para cada switch. Estas configuraciones permiten que el controlador Ryu gestione y configure los switches a través de las direcciones OVSDB especificadas como se muestra en el Bloque de comandos 1.

Bloque de comandos 1
Configuración del protocolo OVSDB en los conmutadores SDN

```
curl -X PUT -d "tcp:192.168.11.3:6632"
http://localhost:8080/v1.0/conf/switches/0000000000000001/ovsdb_addr
curl -X PUT -d "tcp:192.168.11.4:6632"
http://localhost:8080/v1.0/conf/switches/0000000000000002/ovsdb_addr
curl -X PUT -d "tcp:192.168.11.5:6632"
http://localhost:8080/v1.0/conf/switches/0000000000000003/ovsdb_addr
```

Adicionalmente, el script define una asociación entre diferentes valores de DSCP (Differentiated Services Code Point) y colas específicas utilizando un arreglo asociativo, donde los valores DSCP se relacionan con los siguientes marcajes: 46 para EF (Expedited Forwarding), 24 para CS3 (Class Selector 3), 38 para AF33 (Assured Forwarding 4, Class 3), 30 para AF33 (Assured Forwarding 3, Class 3), y 22 para AF23 (Assured Forwarding 2, Class 3). Estos valores se asignan a las colas 0, 1, 2, 3 y 4 respectivamente de acuerdo el nivel de prioridad del tráfico a ser tratado.

En este caso el nivel de prioridad otorgado permite que los paquetes que coincide con múltiples reglas alojadas en el conmutador, este tenga la capacidad de seleccionar la regla con el número de prioridad más alta y realizar su respectiva acción asociada, respecto a otras reglas de menor prioridad, asegurando así la calidad de servicio del tráfico en la red SDN. La Tabla 20 presenta la información relacionada al nivel de

prioridad de la regla de flujo que realiza el encolamiento del tráfico de los diferentes marcajes que ingresan a la red.

Tabla 20
Niveles de prioridad de las reglas de flujo para cada tipo de tráfico

Prioridad	Tráfico	DSCP	Número de cola
100	Expedited Forwarding	46	0
90	Assured Forwarding 3, Class 1	26	1
80	Assured Forwarding 4, Class 3	38	2
70	Assured Forwarding 3, Class 3	30	3
60	Assured Forwarding 2, Class 3	22	4

Establecidos los niveles de prioridad para cada tipo de tráfico, el siguiente paso consiste en establecer las reglas de flujo en cada conmutador de la SDN mediante solicitudes de configuración de reglas de QoS mediante el uso de la API que posee el controlador, especificando la prioridad de los paquetes con el valor DSCP correspondiente con la finalidad de que sean enrutados a la cola designada, asegurando así un manejo adecuado del tráfico que posee los distintos marcajes de acuerdo con las políticas de calidad de servicio, como se muestra en el Bloque de comandos 2.

Bloque de comandos 2
Prioridad y encolamiento del tráfico en base al marcaje

```
#-------------------OpenVswitch-1-----------------------
# EF (Expedited Forwarding)
curl -X POST -d '{"priority": "100", "match": {"ip_dscp": "46"},
"actions":{"queue": "0"}}'
http://localhost:8080/qos/rules/0000000000000001

# CS3 (Class Selector 3)
curl -X POST -d '{"priority": "90", "match": {"ip_dscp": "24"},
"actions":{"queue": "1"}}'
http://localhost:8080/qos/rules/0000000000000001

# AF43 (Assured Forwarding 4, Class 3)
curl -X POST -d '{"priority": "80", "match": {"ip_dscp": "38"},
"actions":{"queue": "2"}}'
http://localhost:8080/qos/rules/0000000000000001

# AF33 (Assured Forwarding 3, Class 3)
curl -X POST -d '{"priority": "70", "match": {"ip_dscp": "30"},
"actions":{"queue": "3"}}'
http://localhost:8080/qos/rules/0000000000000001
```

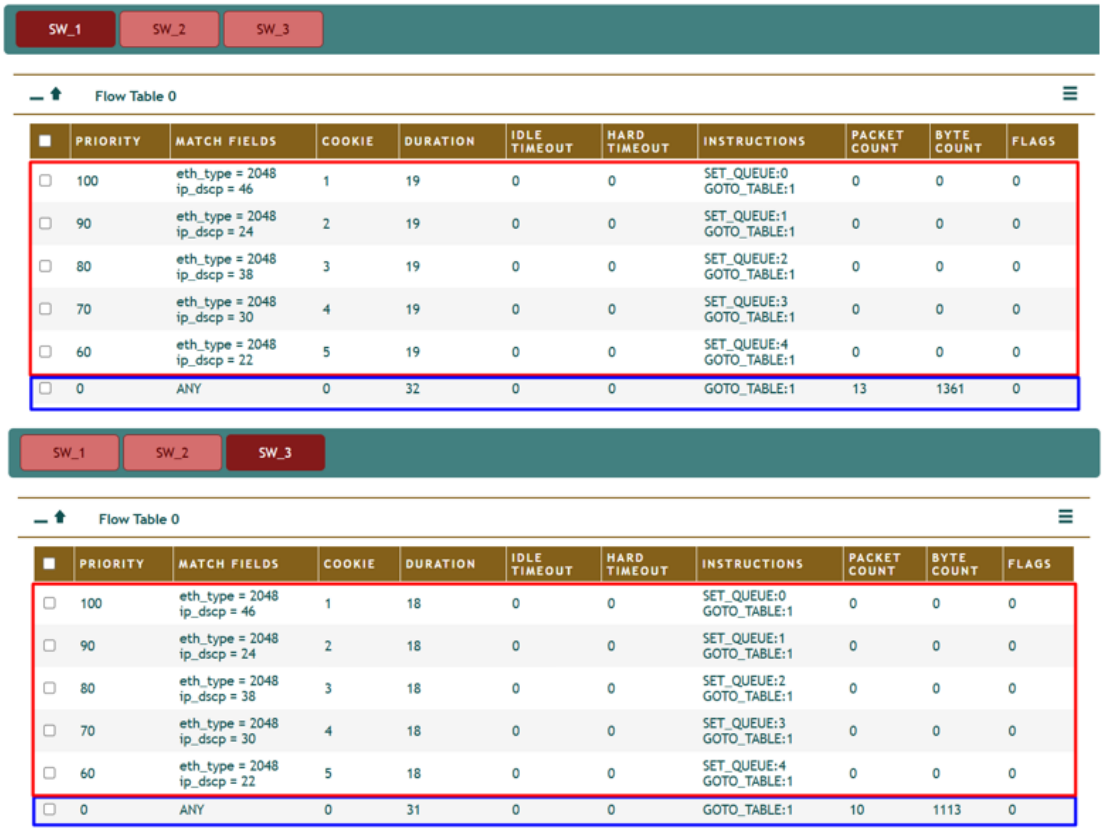
```
# AF23 (Assured Forwarding 2, Class 3)
curl -X POST -d '{"priority": "60", "match": {"ip_dscp": "22"},
"actions":{"queue": "4"}}'
http://localhost:8080/qos/rules/0000000000000001

#------------------OpenVswitch-2----------------------

# EF (Expedited Forwarding)
curl -X POST -d '{"priority": "100", "match": {"ip_dscp": "46"},
"actions":{"queue": "0"}}'
http://localhost:8080/qos/rules/0000000000000002

# CS3 (Class Selector 3)
curl -X POST -d '{"priority": "90", "match": {"ip_dscp": "24"},
"actions":{"queue": "1"}}'
http://localhost:8080/qos/rules/0000000000000002

# AF43 (Assured Forwarding 4, Class 3)
curl -X POST -d '{"priority": "80", "match": {"ip_dscp": "38"},
"actions":{"queue": "2"}}'
http://localhost:8080/qos/rules/0000000000000002

# AF33 (Assured Forwarding 3, Class 3)
curl -X POST -d '{"priority": "70", "match": {"ip_dscp": "30"},
"actions":{"queue": "3"}}'
http://localhost:8080/qos/rules/0000000000000002

# AF23 (Assured Forwarding 2, Class 3)
curl -X POST -d '{"priority": "60", "match": {"ip_dscp": "22"},
"actions":{"queue": "4"}}'
http://localhost:8080/qos/rules/0000000000000002

#------------------OpenVswitch-3----------------------
# EF (Expedited Forwarding)
curl -X POST -d '{"priority": "100", "match": {"ip_dscp": "46"},
"actions":{"queue": "0"}}'
http://localhost:8080/qos/rules/0000000000000003

# CS3 (Class Selector 3)
curl -X POST -d '{"priority": "90", "match": {"ip_dscp": "24"},
"actions":{"queue": "1"}}'
http://localhost:8080/qos/rules/0000000000000003

# AF43 (Assured Forwarding 4, Class 3)
curl -X POST -d '{"priority": "80", "match": {"ip_dscp": "38"},
"actions":{"queue": "2"}}'
http://localhost:8080/qos/rules/0000000000000003

# AF33 (Assured Forwarding 3, Class 3)
curl -X POST -d '{"priority": "70", "match": {"ip_dscp": "30"},
"actions":{"queue": "3"}}'
http://localhost:8080/qos/rules/0000000000000003

# AF23 (Assured Forwarding 2, Class 3)
```

```
curl -X POST -d '{"priority": "60", "match": {"ip_dscp": "22"},
"actions":{"queue": "4"}}'
http://localhost:8080/qos/rules/0000000000000003
```

La verificación de las reglas de flujo que aseguran el encolamiento del tráfico para para proporcionar calidad de servicio en la red SDN puede realizarse a través del plano de aplicación, el cual identifica las reglas de cada conmutador SDN mediante su interfaz web. La Figura 28 presentan las reglas de flujo establecidas en los conmutadores, en donde las reglas remarcadas con el recuadro rojo corresponden a los flujos con una alta prioridad que se procesaran primero, mientras que la regla del recuadro azul actúa como una red de seguridad para cualquier tráfico que no coincida con las reglas de mayor prioridad. Este enfoque asegura que el tráfico crítico (como el tráfico con ip_dscp = 46) reciba un tratamiento preferencial, mientras que el tráfico no coincidente se maneje por la regla predeterminada.

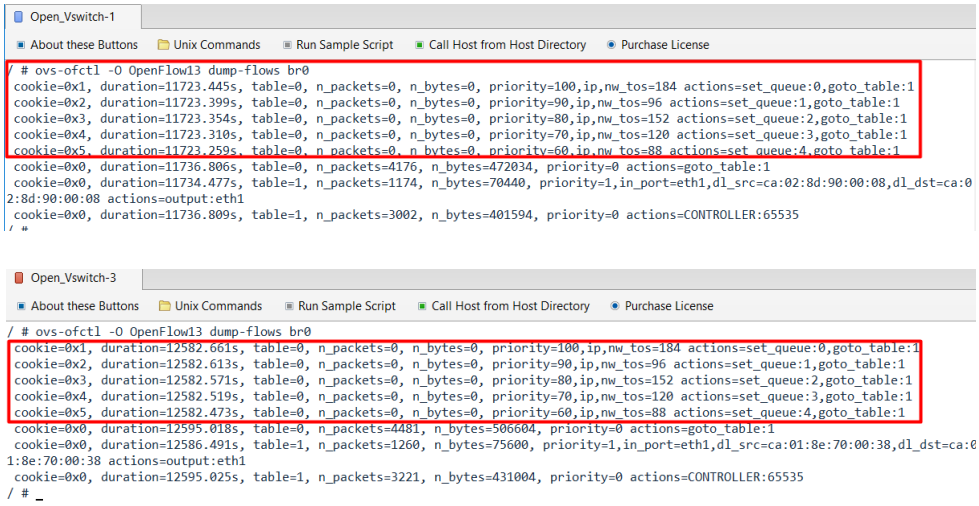
Figura 28

Reglas de flujo de calidad de servicio en los conmutadores SDN



Finalmente, en la consola de comandos de cada conmutador SDN se pueden identificar de igual manera las reglas de flujo registradas que poseen la información relacionada con el nivel de prioridad y el encolamiento que realiza a los diferentes tipos de tráfico que coinciden con los distintos DSCP que aseguran la calidad de serivicio en la red híbrida SDN. La Figura 29 presenta la información desplegada en la tabla de flujos de los conmutadores OpenFlow 1 y 3 mediante el comando `ovs-ofctl -O OpenFlow13 dump-flows br0`. Las reglas alojadas en el recuadro rojo están poseen diferentes niveles de prioridad (100, 90, 80, 70, 60) y coinciden con paquetes IP con valores específicos de nw_tos correspondiente a los valores del DSCP de cada marcaje. Las acciones asociadas a estas reglas incluyen la asignación de los paquetes a diferentes colas y su redirección a la siguiente tabla de flujo (goto_table:1).

Figura 29
Reglas de flujo de calidad de servicio en los conmutadores SDN

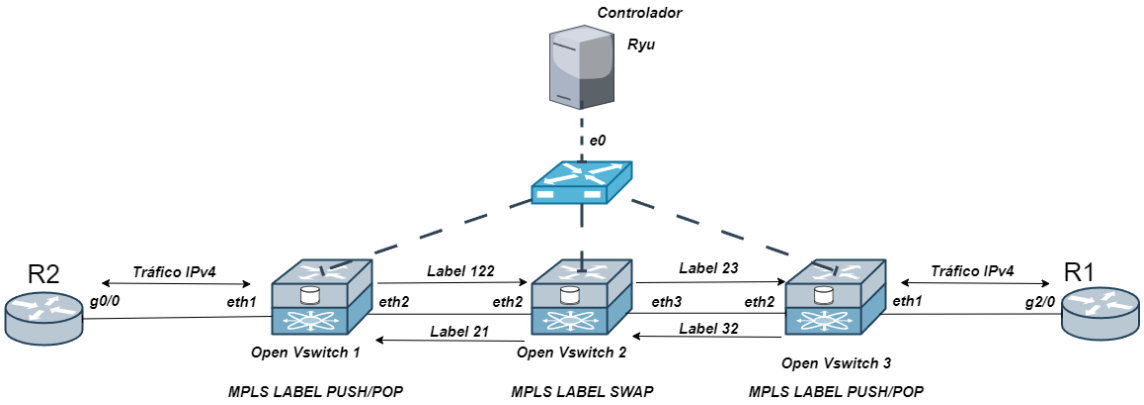


3.2.8 Implementación de SDN/MPLS

La implementación de las funciones de MPLS dentro de la red SDN corresponde en establecer reglas de flujo que permitan la clasificación de los paquetes IPv4 provenientes de la infraestructura de red tradicional, de modo que al ingresar a la red SDN los conmutadores que la conforman realicen las operaciones de label push, label swap y

label pop, permitiendo así establecer la red híbrida SDN/MPLS . La Figura 30 presenta la infraestructura MPLS, en donde se presentan las etiquetas intercambiadas por cada conmutador SDN y las operaciones que realizan cada uno de ellos.

Figura 30. Distribución del core MPLS dentro de la red SDN



Una vez definidas las funciones MPLS que desempeñará cada conmutador y las etiquetas que intercambiarán, el siguiente paso consiste en identificar el conjunto de comandos de configuración de los conmutadores SDN (Open vSwitches), con el fin de que puedan llevar a cabo operaciones MPLS dentro del plano de datos. En la

Tabla **21** se detallan las sentencias requeridas para establecer los flujos dentro de los conmutadores SDN en el marco de la implementación de la tecnología MPLS.

Tabla 21
Sentencias OpenFlow en conmutadores OpenVswtich

Sentencia	Descripción
`ovs-ofctl`	Línea de comandos que permite monitorear y administrar conmutadores OpenFlow
`add-flow`	Agrega un nuevo flujo en una interfaz bridge del conmutador SDN.
`table=number`	Establece que en una tabla se va a guardar el flujo establecido.
`in_port=port`	Define un puerto de entrada de un conmutador OpenFlow.
`dl_type=#`	Los valores pueden alinearse precisamente con el valor específico.En el contexto de la implementación de MPLS, se emplean tres tipos distintos: 0x0800 (para IPv4), 0x8847 (para MPLS) y 0x0806 (para ARP).

actions=push_mpls	Realizan la operación push de MPLS para permitir el envió de las etiquetas al siguiente salto definiendo el valor de ethertype 0x8847.
actions=pop_mpls	Realizan la operación pop de MPLS para permitir el envió de las etiquetas al siguiente salto definiendo el valor de ethertype 0x8847.
output:port	Define un puerto de salida de un conmutador OpenFlow.

Nota. Adaptado de (OpenVswitch,2023)

Definidas las sentencias de entrada para nuevos flujos en los conmutadores, se lleva a cabo la configuración de cada Open vSwitch. A continuación, se presentan la Tabla 22 y Tabla 24 con las reglas de flujo establecida para las funciones POP y PUSH en los conmutadores LER de la topología, mientras que la Tabla 23 presenta las reglas de flujo para establecer la operación SWAP.

Tabla 22

Reglas de flujo para operaciones PUSH Y POP en el OpenVswitch-1

Descripción	Acción	Sentencia
Si el paquete llega por el puerto 2 y es un paquete IPv4:	-Push label 122 -Establecer el campo "mpls_label" en 122 -Enviar el paquete por el puerto 3	`ovs-ofctl -O OpenFlow13 add-flow br0 "table=1,priority=100,in_port=2,eth_type=0x800,ip_proto=17 ,actions=push_mpls:0x8847,set_field:122->mpls_label,output:3"`
Si el paquete llega por el puerto 3 y es un MPLS unicast con etiqueta 21:	-Pop label -Establecer el tipo de Ethernet como IPv4 (0x0800) -Enviar el paquete por el puerto 2	`ovs-ofctl -O OpenFlow13 add-flow br0 "table=1,priority=100,in_port=3,eth_type=0x8847,mpls_bos=1 ,mpls_label=21,actions=pop_mpls:0x0800,output:2"`

Tabla 23

Reglas de flujo para operaciones SWAP en el OpenVswitch-2

Descripción	Acción	Sentencia
Si el paquete llega por el puerto 3 y es un paquete MPLS:	- Pop label 122 - Push label 23 - Establecer el campo "mpls_label" en 23 - Enviar el paquete por el puerto 4	`ovs-ofctl -O OpenFlow13 add-flow br0 "table=1,priority=100,in_port=3,eth_type=0x8847,mpls_label =122,actions=pop_mpls:0x800, push_mpls:0x8847,set_field:23->mpls_label,output:4"`
Si el paquete llega por el puerto 4 y es un paquete MPLS:	- Pop label 32 - Push label 21 - Establecer el campo "mpls_label" en 21 - Enviar el paquete por el puerto 3	`ovs-ofctl -O OpenFlow13 add-flow br0 "table=1,priority=100,in_port=4,eth_type=0x8847,mpls_label =32,actions=pop_mpls:0x800, push_mpls:0x8847,set_field:21->mpls_label,output:3"`

Tabla 24
Reglas de flujo para operaciones PUSH y POP en el OpenVswitch-3

Descripción	Acción	Sentencia
Si el paquete llega por el puerto 1 y es un paquete IPv4:	- Push label 32 - Establecer el campo "mpls_label" en 32 - Enviar el paquete por el puerto 2	`ovs-ofctl -O OpenFlow13 add-flow br0 "table=1,priority=100,in_port=2,eth_type=0x800,ip_proto=17 ,actions=push_mpls:0x8847,set_field:32- >mpls_label,output:3"`
Si el paquete llega por el puerto 2 y es un MPLS unicast con etiqueta 23:	-Pop label -Establecer el tipo de Ethernet como IPv4 (0x0800) -Enviar el paquete por el puerto 1	`ovs-ofctl -O OpenFlow13 add-flow br0 "table=1,priority=100,in_port=3,eth_type=0x8847,mpls_bos=1 ,mpls_label=23,actions=pop_mpls:0x0800,output:2"`

3.3 Primer escenario: Entorno de ataques DoS

El primer escenario del testbed toma la infraestructura híbrida SDN puesta en marcha en la sección anterior, con la premisa de que en ella se alojaran hosts mal intencionados que tienen como finalidad de afectar la disponibilidad de la red aprovechándose de las capacidades del plano de datos de la arquitectura SDN en conjunto con los mensajes del protocolo OpenFlow que intercambian con el controlador, de modo que la operabilidad de este tipo de infraestructuras se vea afectado en su funcionamiento ante este tipo de amenazas.

Esta sección del documento establece el diseño de la topología del escenario del entorno de ataques DoS de tipo inundación ICMP, la planificación del proceso de despliegue y finalmente el análisis de sus repercusiones generadas a la infraestructura de red híbrida SDN mediante la metodología STRIDE.

3.3.1 Diseño de la topología del entorno de ataques

Tomando como base la infraestructura de red híbrida SDN presentada en la sección 3.2.1, el primer escenario del testbed establece que los hosts ilegítimos establecerán una conexión directa hacia los conmutadores SDN que conforman la nube MPLS ilustrada en el esquema de red de la Figura 11. Este planteamiento en el diseño de la topología se debe a que la nube MPLS establece una conexión segura y dedicada entre

las redes corporativas, de modo que, si los atacantes logran acceder a dicha nube, tendrán acceso directo a las redes de las organizaciones, como se muestra en la Figura 31.

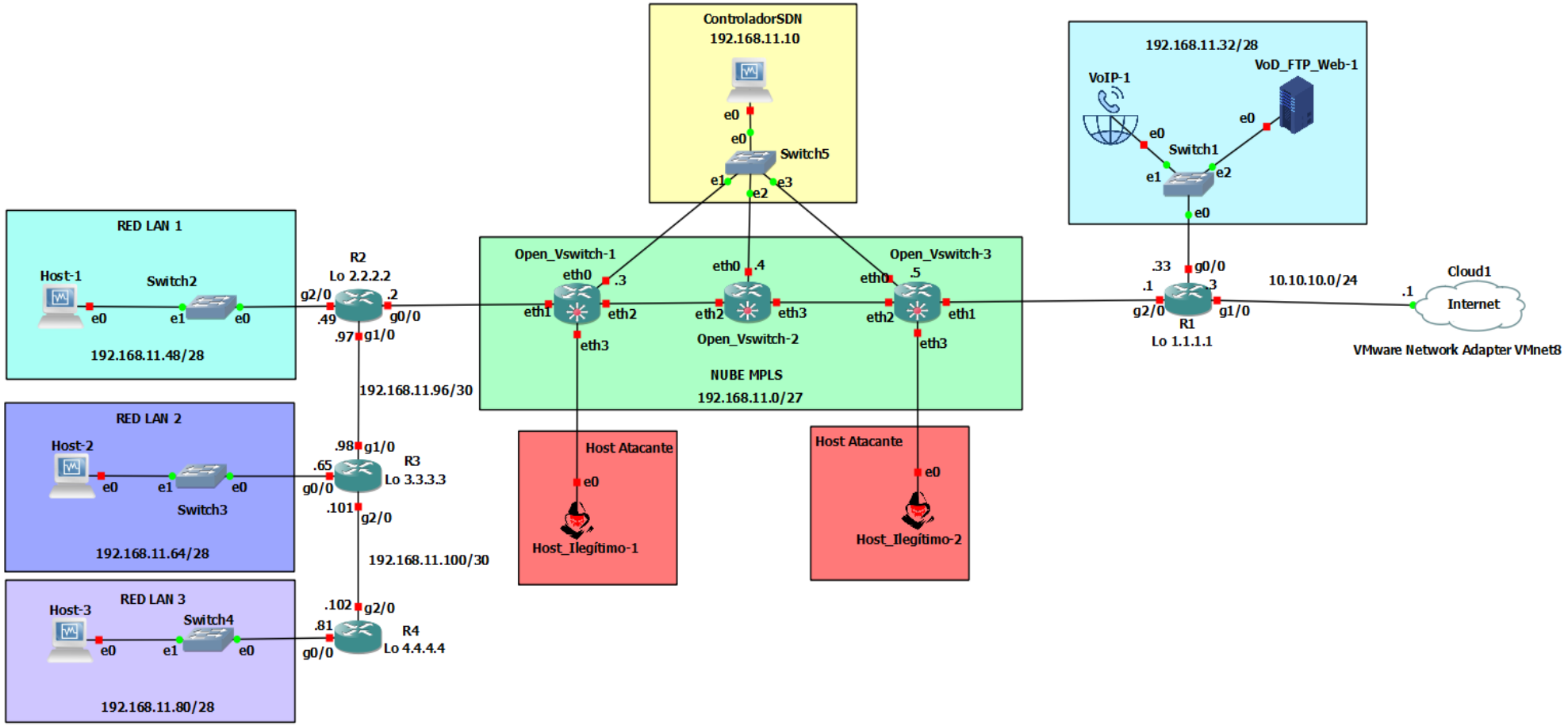
Este tipo de ataques provocan la saturación del búfer en los conmutadores SDN, lo que resulta en la retransmisión completa de paquetes al controlador. Con el tiempo, esto puede sobrecargar la capacidad de procesamiento del controlador, causando que los switches pierdan su capacidad para reenviar tráfico legítimo y, en última instancia, dejando la red inoperable. La nube MPLS al poseer características de alta velocidad y baja latencia representa una ventaja para los atacantes, ya que les permite enviar tráfico a gran velocidad y con poca demora, lo que dificulta su detección cuando se establecen ataques DoS.

La ejecución de este tipo de ataques a la infraestructura híbrida SDN tienen como propósito principal saturar las funciones del controlador, ya que al generar altos volúmenes de tráfico también compromete de forma directa al plano de datos que maneja cada uno de los conmutadores SDN, lo que genera que experimenten una alta tasa de llegada de nuevos flujos de paquetes en las tablas TCAM en un corto período.

Este tipo de ataques provocan la saturación del búfer en los conmutadores SDN, lo que resulta en la retransmisión completa de paquetes al controlador. Con el tiempo, esto puede sobrecargar la capacidad de procesamiento del controlador, causando que los switches pierdan su capacidad para reenviar tráfico legítimo y, en última instancia, dejando la red inoperable.

Figura 31

Topología de red del entorno de ataques del primer escenario



3.3.2 Plan de ataques

El diseño del plan de ataques toma como referencia el OSSTMM3[12], publicado por la ISECOM (2024), el cual describe a la metodología Data Networks Security Testing[13] como la encargada de establecer un proceso secuencial que permite realizar pruebas de penetración en redes de datos. Por lo tanto, este enfoque metodológico se alinea con el propósito de este apartado, que es establecer un plan de acciones para llevar a cabo ataques de denegación de servicio en redes híbridas SDN.

En el contexto del presente proyecto, se toma como base a la sección de Auditoria de Visibilidad de la metodología elegida, de modo que este apartado presenta una serie de criterios necesarios a tomar en cuenta en el plan de ataques. Esto se realiza con el propósito de establecer un modelo, el cual el atacante utiliza para realizar las pruebas de seguridad a la red SDN. De acuerdo con esta sección la (ISECOM, 2024) establece los siguientes criterios a tomar en cuenta.

a) Topografía de red

Visión general de cómo están conectados los dispositivos dentro de la red, mostrando la ubicación física de cada dispositivo y cómo se comunican entre sí para transmitir datos. Los criterios seleccionados para establecer el desarrollo del plan son identificados con la nomenclatura TR, siendo los siguientes:

- **TR-1:** Identificar los segmentos de red y los objetivos desde el cual se probarán los ataques**.**
- **TR-2:** Utilizar el rastreo de redes para identificar protocolos que emanan de respuestas o solicitudes de servicios de red**.**

[12] OSSTM3: Manual de Metodología de Pruebas de Seguridad de Código Abierto.
[13] Data Networks Security Testing (COMSEC): Pruebas de Seguridad en Redes de Datos.

- **TR-3:** Verificar las solicitudes de transmisión y las respuestas de todos los objetivos**.**
- **TR-4:** Verificar las respuestas ICMP para códigos ICMP 0-2 en todos los objetivos.

b) Identificación

Se enfoca en determinar y entender los componentes y características de la red de datos objetivo, previo a realizar pruebas de seguridad. Los criterios seleccionados para establecer el desarrollo del plan son identificados con la nomenclatura ID, siendo los siguientes:

- **ID-1:** El atacante y el objetivo se encuentran separados a una distancia considerada, no obstante, el objetivo del ataque es accesible a través de la red de área global.
- **ID-2:** La distancia entre el atacante y el objetivo se limita a una red local y en principio no poseen interacción directa entre ambos.
- **ID-3:** El individuo puede generar tráfico de forma anónima en la red donde se encuentra el objetivo del ataque.

Una vez que se han identificado los criterios establecidos por la OSSTMM3 para las pruebas de seguridad en redes de datos, la siguiente etapa del desarrollo del plan de ataques implica establecer una nomenclatura que identifique la secuencia de casos de prueba a ser ejecutados para realizar los ataques DoS. Cada caso de prueba se designará con las siglas "COMSEC-CASE-#", donde COMSEC corresponde a las siglas de la metodología utilizada, y "Case-#" indica el número específico del caso que será abordado en el desarrollo.

Finalmente, la Tabla 25 proporciona una lista detallada de casos de prueba específicos diseñados para evaluar la seguridad de una red híbrida SDN. Cada caso de

prueba está etiquetado con un identificador único de acuerdo con la nomenclatura establecida previamente. Además, en cada caso se describen los requisitos asociados, un resumen de la situación esperada y los objetivos que se pretenden alcanzar mediante la realización de la prueba, de modo que la finalidad de esta tabla es guiar el proceso de evaluación de seguridad al proporcionar un conjunto claro de escenarios de prueba que abordan diferentes aspectos de la red.

Tabla 25
Plan de ataques DoS a redes SDN

Caso de prueba	Requisitos	Resumen	Objetivo de la prueba
COMSEC-CASE-001	TR-1 TR-2 ID-2	Es posible identificar los dispositivos que conforman la red SDN, permitiendo establecer una fase inicial de ataques en base a la información obtenida.	Recopilar información de la red, respecto a los hosts y enlaces conectados a los conmutadores SDN.
COMSEC-CASE-002	TR-3 TR-4 ID-1	Identificada la información de la red, se pueden realizar pruebas de conectividad mediante pings a los hosts y enlaces descubiertos.	Identificar la conectividad que puede establecer el atacante con los hosts descubiertos.
COMSEC-CASE-003	ID-2	Con las respuestas de conectividad detectadas, se establecen las herramientas que permiten ataques DoS de inundación ICMP, además es necesario reconocer su sintaxis de ejecución.	Seleccionar las herramientas de ataques DoS que posean funciones de inundación ICMP.
COMSEC-CASE-004	TR-3 TR-4 ID-3	El atacante puede generar ataques DoS a los enlaces que posee la red híbrida SDN, comprometiendo la disponibilidad de la red y los servicios que posee.	Utilizar las herramientas seleccionadas con la premisa de que comprometan la disponibilidad de la red híbrida SDN.
COMSEC-CASE-005	ID-2	Durante el desarrollo de los ataques, es necesario identificar el consumo de recursos por parte del controlador e identificar el intercambio de mensajes OpenFlow entre los conmutadores atacados y el controlador.	Verificar como los ataques DoS comprometen la disponibilidad de la red híbrida SDN en conjunto con las funciones del controlador SDN.

3.3.3 Ejecución de los ataques en el primer escenario

La ejecución de los ataques se llevará a cabo de manera secuencial, siguiendo cada uno de los casos de prueba definidos en el plan descrito en la Tabla 25. Cada caso en esta sección del documento representa una etapa que debe completarse para el despliegue completo de los ataques de denegación de servicio de tipo inundación ICMP hacia una red híbrida SDN. A continuación, se describen dichas etapas:

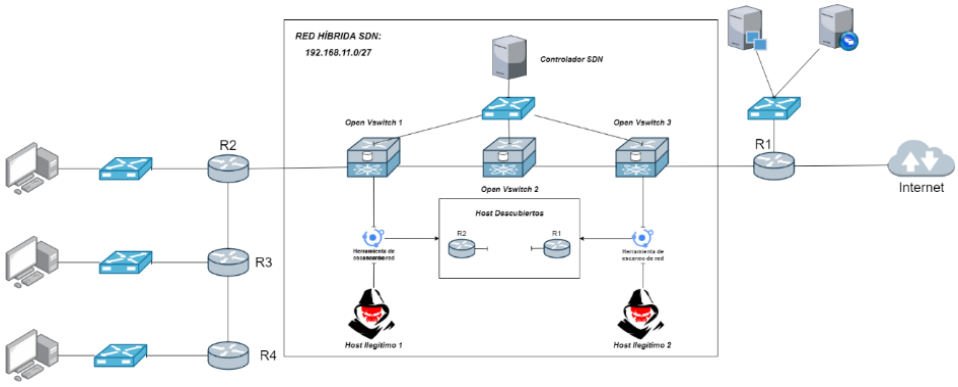
a) Caso de prueba COMSEC-CASE-001

Una de las fases iniciales del plan de ataques implica realizar una inspección de todos los componentes que conforman la red híbrida SDN. Como parte de este proceso, el primer escenario de pruebas del testbed establece, según el diseño presentado en la Figura 31, se evidencia que los atacantes tienen una conexión directa hacia los dos conmutadores (OpenVswitch) de borde con la infraestructura de red tradicional y a la vez poseen una dirección IPv4 dentro de la red 192.168.11.0/27, la cual está asociada a la red SDN.

Identificando que los atacantes cumplen el requerimiento ID-2 del plan de ataques, estos pueden realizar un sondeo de la red con el objetivo de detectar los elementos de red a los cuales pueden acceder desde la SDN. Herramientas como Nmap permiten recopilar esta información al generar una lista de hosts que responden a solicitudes de conexión, como se ilustra en la Figura 32.

Figura 32

Proceso de descubrimiento de hosts disponibles dentro de la red SDN



En la Figura 33 se muestra el uso de la herramienta de escaneo de red conocida como Nmap con la opción "-sn", la cual indica la realización de un escaneo de tipo "ping scan". Por lo tanto, el atacante envía un paquete ICMP Echo Request a cada dirección IP en el rango especificado (192.168.11.0/27) para determinar qué direcciones IP están activas y disponibles en la red, para finalmente enlistar esta información para ser presentada.

Figura 33

Elementos de red detectados en la SDN

```
┌──(root㉿keneth)-[/home/keneth]
└─# nmap -sn 192.168.11.0/27

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-15 00:31 -05
Nmap scan report for 192.168.11.1
Host is up (0.0035s latency).
MAC Address: CA:01:8E:70:00:38 (Unknown)
Nmap scan report for 192.168.11.2
Host is up (0.014s latency).
MAC Address: CA:02:8D:90:00:08 (Unknown)
Nmap scan report for 192.168.11.21
Host is up (0.24s latency).
MAC Address: 08:00:27:53:1F:FC (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.11.20
Host is up.
Nmap done: 32 IP addresses (4 hosts up) scanned in 24.12 seconds
```

Después de completar el escaneo, se identificaron 4 direcciones IPv4 que utilizan el plano de datos de la arquitectura SDN. Entre estas, las direcciones 192.168.11.1 y 192.168.11.2 corresponden a los enlaces establecidos entre la SDN y la infraestructura de red tradicional. Estas direcciones IP se convierten en los objetivos de los atacantes,

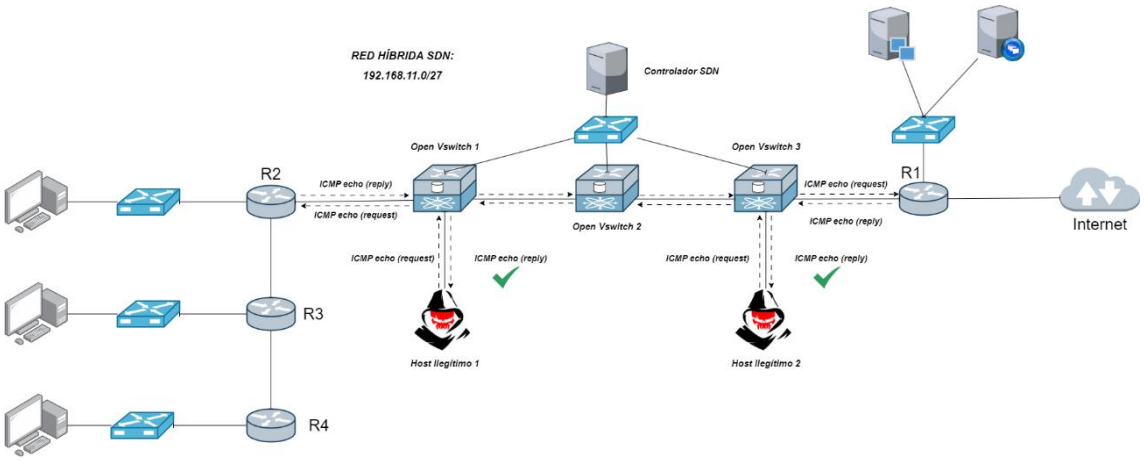
quienes buscan comprometer la disponibilidad de la red híbrida SDN al saturar las capacidades de los conmutadores a los que están conectados.

b) Caso de prueba COMSEC-CASE-002

Tomando la información recopilada del caso de prueba COMSEC-CASE-001 referente a direccionamiento IPv4 disponible dentro del plano de datos de la red SDN, la siguiente fase del plan de ataques corresponde en establecer conectividad mediante solicitudes de ping hacia los hosts descubiertos, que en este caso serán los enlaces de los enrutadores que poseen conectividad y hacen uso de las funciones de conmutación del plano de datos de los switches SDN. En la Figura 34 se representa el proceso de conectividad utilizado por los atacantes para demostrar que tienen comunicación con los hosts que componen la topología SDN.

Figura 34

Comprobación de conectividad de los atacantes a posibles objetivos



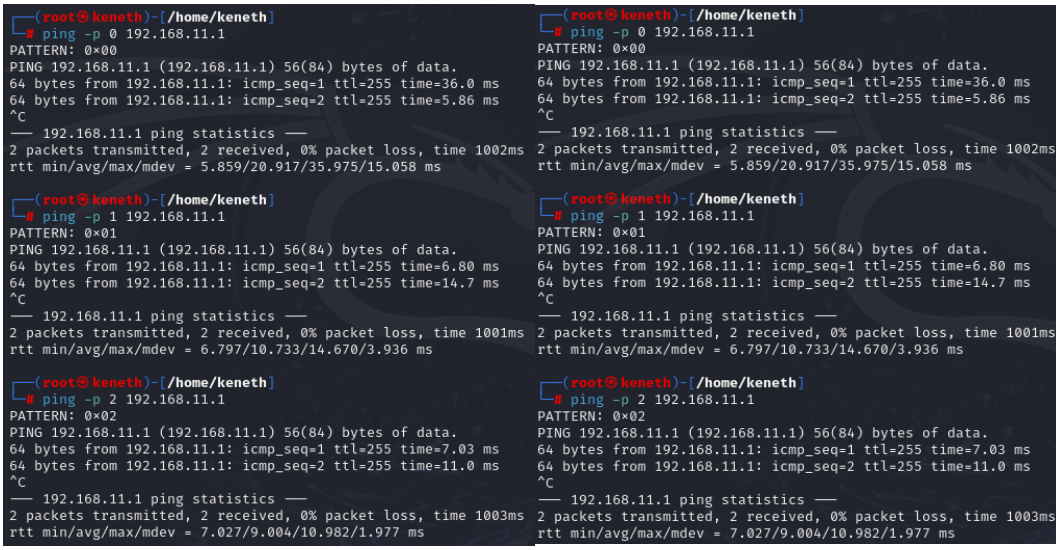
La Figura 35 muestra las pruebas de ping realizadas por los atacantes, las cuales fueron exitosas, cada solicitud realizada por los atacantes a los objetivos utiliza distintos códigos ICMP que van de 0 a 2, siendo estos los siguientes:

- ICMP Echo Request (código 8): tipo de mensaje utilizado en redes para probar la conectividad entre dos dispositivos. Cuando un dispositivo

envía este tipo de solicitud, está pidiendo a otro dispositivo que responda con un *Echo Reply* (código 0).

- ICMP Echo Reply (código 0): Mensaje de respuesta cuando un host recibe un paquete de solicitud de *Echo Request* (código 8), este responde con un paquete de eco de vuelta al remitente y es utilizado comúnmente para probar la conectividad de red entre dos hosts.
- ICMP Destino Inalcanzable (código 1): Este mensaje ICMP se utiliza para indicar que un paquete no pudo ser entregado a su destino final. Este mensaje se envía de vuelta al remitente para informarle sobre la imposibilidad de la entrega del paquete.

Figura 35
Comprobación de conectividad mediante códigos ICMP



c) Caso de prueba COMSEC-CASE-004

La tercera fase del plan de ataques corresponde en seleccionar herramientas que lleve a cabo ataques DoS de tipo inundación ICMP hacia los objetivos. Este proceso es realizado después de verificar en el caso de estudio previo, la existencia de conectividad entre los atacantes y los objetivos. En este caso, los objetivos son los enlaces de los enrutadores que establecen conexión con la infraestructura SDN.

Una herramienta es considerada para llevar a cabo ataques de denegación de servicio (DoS) si exhibe características que indican sus funcionalidades maliciosas, tales como la capacidad de generar grandes volúmenes de tráfico no deseado, como peticiones falsas o paquetes malformados. Estos ataques tienen como objetivo agotar los recursos del sistema, incluyendo el ancho de banda, el uso de la CPU y la memoria. Su propósito es dificultar o bloquear la disponibilidad de los usuarios legítimos a la red y los servicios alojados en ella. Estos ataques suelen aprovechar las vulnerabilidades presentes en los conmutadores SDN, en combinación con los mensajes del protocolo OpenFlow, que constituyen el tema de investigación en cuestión.

La elección de las herramientas como hping3 y t50 se basa en el impacto de saturación de recursos que generan contra el objetivo de destino de forma intencional y con intenciones puramente maliciosas, de modo que estas herramientas permiten ejecutar ataques DoS para interrumpir o degradar la disponibilidad del controlador utilizando las capacidades de conmutación del plano de datos pertenecientes a los switches SDN. La Tabla 26 presenta detalladamente los tipos específicos de ataques de inundación DoS que realizan, además de las funciones y enfoques que posee cada herramienta.

Tabla 26
Principales herramientas para ataques DoS de tipo inundación

Herramienta	Simula	Funcionamiento	Orientación
hping3	Paquetes ICMP/UDP/TCP	Generador de tráfico en red de tipo flooding.	Denegación de servicio y saturación de recursos
t50	Paquetes ICMP/UDP/TCP	Genera tráfico dentro de una red	Denegación de servicio y saturación de recursos

La selección de hping3 como herramienta de implementación para ataques DoS de tipo inundación ICMP se basa en su capacidad de generar grandes volúmenes de tráfico ICMP, UDP y TCP, lo que lo convierte en un generador de tráfico en red versátil y

potente. Además, hping3 está diseñado específicamente para realizar ataques de denegación de servicio (DoS), lo que significa que sus funcionalidades están orientadas hacia la saturación de recursos del sistema objetivo, incluyendo el ancho de banda y capacidades de conmutación de los switches SDN, que son el foco de esta investigación.

El comando `hping3 direccion IP --flood --rand-source --icmp -c #` utilizado para realizar los ataques DoS de tipo inundación ICMP, consta de varios campos que determinan su funcionamiento, siendo estos especificados en la Tabla 18.

Tabla 27
Especificaciones del comando hping para ataques DoS de inundación ICMP

Campo	Descripción
Dirección IP	La dirección IP del objetivo al que se dirigirán los ataques
--flood	Activa el modo de inundación de hping3.
--rand-source	Utiliza direcciones IP de origen aleatorias.
--icmp	Especifica que se utilizarán paquetes ICMP para el ataque.
-c #	Establece el número de paquetes ICMP que se enviarán.

d) Caso de prueba COMSEC-CASE-004

El cuarto caso de prueba implica la ejecución de los ataques DoS de tipo inundación utilizando la herramienta hping3 seleccionada previamente. Los atacantes utilizarán la sintaxis especificada en la Tabla 27 para configurar los parámetros que permiten la inundación ICMP. Los objetivos de estos ataques son las direcciones IPv4 de los enlaces correspondientes a los enrutadores R1 y R2, como se identifican en la Figura 31. Estos routers utilizan las funciones de conmutación del plano de datos de la red SDN para establecer la interconectividad con la infraestructura de red tradicional.

Inicialmente, para comenzar con el proceso de ataques DoS de tipo inundación cada uno de los hosts maliciosos establecerá sus objetivos de ataque. La Tabla 28 detalla

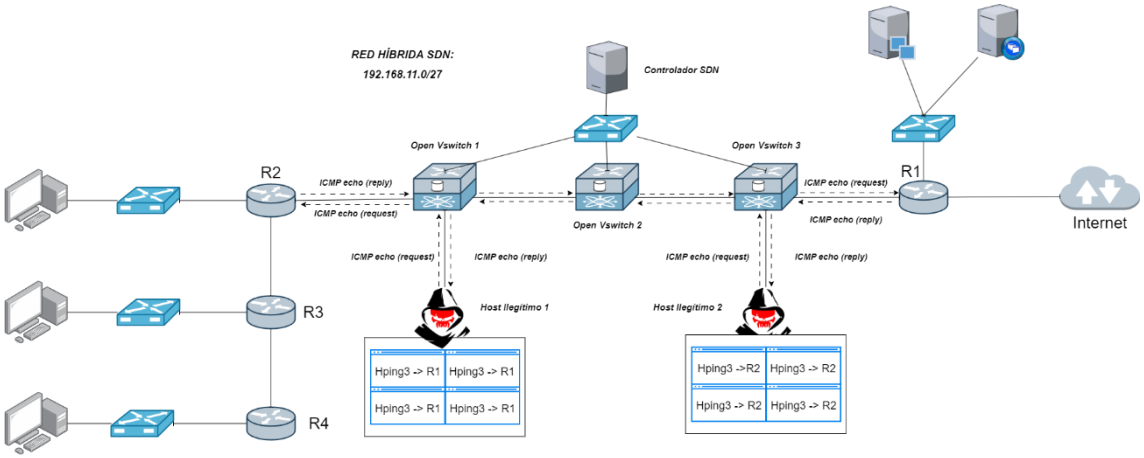
cómo cada atacante dirige un ataque de inundación ICMP hacia las direcciones IPv4 de los objetivos seleccionados.

Tabla 28
Direccionamiento de los ataques hacia los objetivos

Objetivo	Interfaz	Dirección IPv4	Atacante designado
R1	f2/0	192.168.11.1	Atacante-1
R2	f0/0	192.168.11.2	Atacante-2

Considerando la premisa de llevar a cabo ataques simultáneos a los objetivos, la Figura 36 muestra el proceso de ejecución de ataques de Denegación de Servicio (DoS). Cada host malicioso realiza estos ataques mediante cuatro sesiones en las terminales de comandos para iniciar una inundación ICMP. Este método asegura una mayor carga de tráfico y una mayor demanda de capacidad de procesamiento por parte de los conmutadores y el controlador SDN, lo que aumenta la efectividad del ataque y maximiza la posibilidad de saturar las capacidades de funcionamiento de la SDN.

Figura 36
Ataques DoS a la red híbrida SDN



Una vez establecidos los objetivos de cada atacante, el proceso es iniciado utilizando la herramienta hping3 especificando la dirección IPv4 del objetivo designado.

La Figura 37 y Figura 38 muestra el proceso de ejecución de los ataques DoS de inundación ICMP por parte de los hosts maliciosos.

Figura 37
Ataques DoS de inundación ICMP dirigidas al enlace R1

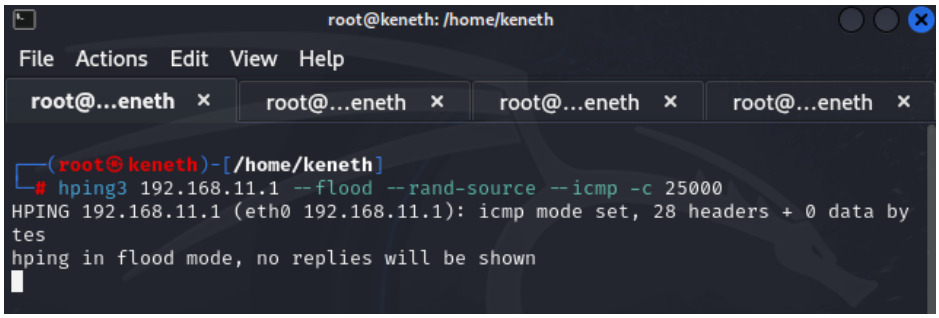
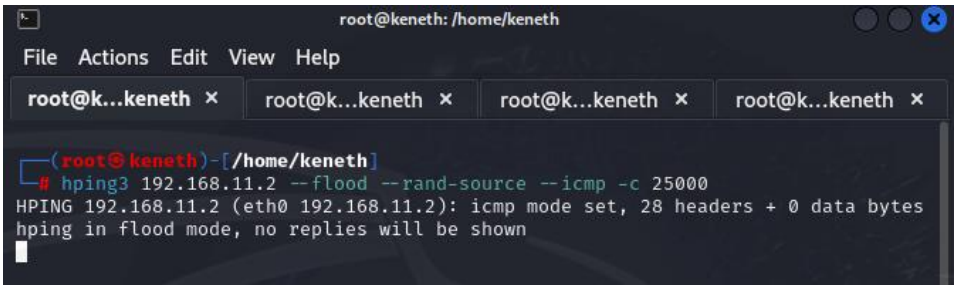


Figura 38
Ataques DoS de inundación ICMP dirigidas al enlace de R2

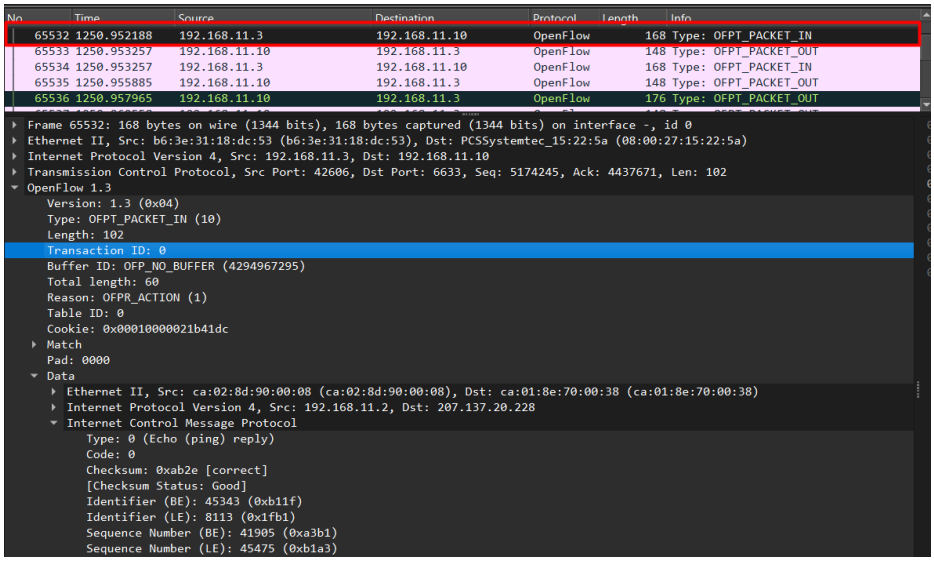


e) Caso de prueba COMSEC-CASE-005

El último caso de prueba recopila las repercusiones generadas por los ataques a la red híbrida SDN. Capturando el tráfico de los enlaces que conectan los conmutadores atacados con el controlador y utilizando herramientas de análisis de paquetes como Wireshark, permiten identificar en primera instancia que los ataques saturan la capacidad de almacenamiento del buffer del conmutador, de modo que el switch SDN envía el paquete completo al controlador, encapsulándolo dentro del mensaje OFPT_PACKET_IN como se muestra en la Figura 39.

Figura 39

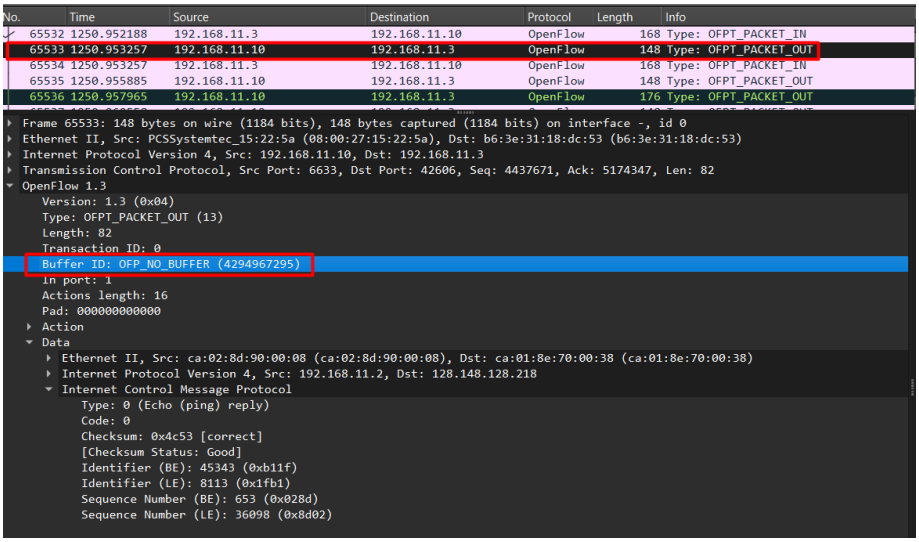
Paquete OFTP_IN con información ICMP encapsulada



Si este proceso es realizado, el controlador responde utilizando el mensaje OFPT_PACKET_OUT que también transporta el paquete de datos completo. En este proceso, no se establece ninguna regla de flujo y el conmutador simplemente realiza el proceso de conmutación del paquete. La Figura 40 muestra el contenido del paquete OFPT_PAKCET_OUT que indica que el almacenamiento en buffer está completamente lleno.

Figura 40

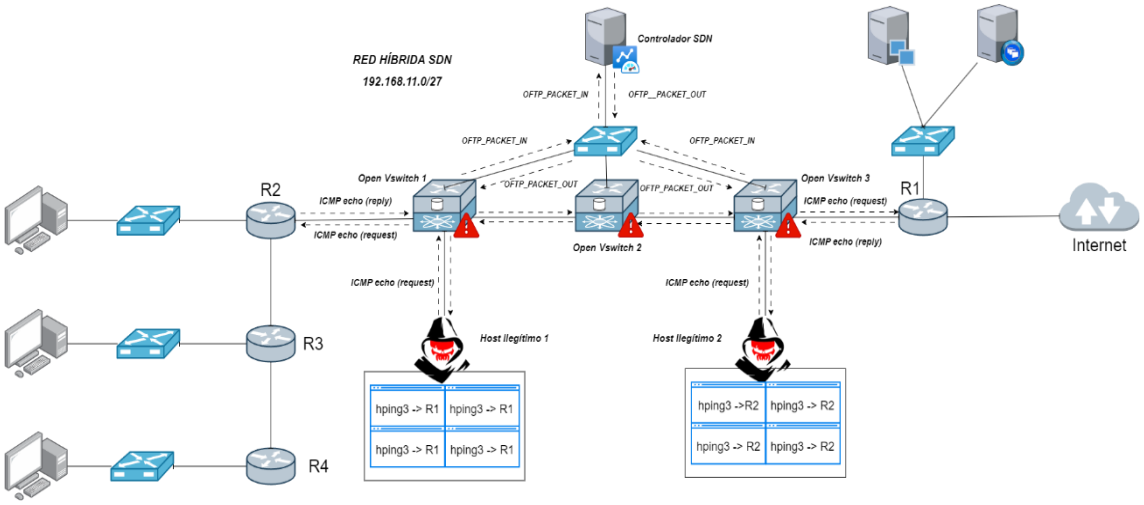
Paquete OFTP_OUT con información ICMP encapsulada



Tomando en cuenta la información recopilada, la Figura 41 ilustra como los conmutadores SDN experimentan altas tasas de llegada de nuevos flujos de paquetes en un corto período, de modo que su búfer puede saturarse, lo que resulta en la retransmisión de paquetes completos al controlador. Esta situación implica un significativo consumo del ancho de banda del plano de control y como consecuencia aumenta la latencia en la instalación de nuevas entradas en la tabla de flujo y en situaciones extremas, el conmutador podría no ser capaz de reenviar el tráfico de estos nuevos flujos.

Figura 41

Comportamiento de los conmutadores SDN ante ataques Dos



La pérdida de paquetes puede manifestarse de las siguientes maneras:

- El tamaño de la cola de salida del conmutador es limitado: si el enlace entre el conmutador y el controlador está congestionado, la cola de salida del conmutador se llena y el mensaje OFPT_PACKET_IN en sí no se puede enviar, lo que provoca la pérdida de paquetes.
- Alta latencia: la congestión puede introducir latencia en el canal de control. Si el conmutador no recibe una respuesta OFPT_FLOW_MOD dentro de un cierto período de tiempo, descarta el paquete almacenado en el buffer correspondiente.

Las repercusiones mencionadas implican un ataque al conmutador objetivo, sin embargo, si el ataque logra abrumar los recursos informáticos del controlador y deniega su servicio, todos los conmutadores conectados al controlador afectado pueden sentir el impacto del ataque.

3.3.4 Análisis de ataques DoS en redes SDN mediante la metodología STRIDE

La selección de la metodología STRIDE se centra en identificar, comprender y mitigar las repercusiones de los ataques de denegación de servicio. Para abordar esta problemática, se emplean tres etapas clave correspondientes a: la identificación de activos y componentes del sistema, la evaluación de amenazas STRIDE y el análisis de riesgos; posterior a la ejecución de los ataques presentados en la sección anterior. Estas fases colaborativas permiten una comprensión integral de la infraestructura, las vulnerabilidades potenciales y las posibles consecuencias de los ataques DoS en el entorno híbrido SDN.

a) Identificación de activos y componentes del sistema

El reconocimiento de los activos y componentes de la red híbrida corresponde a uno de los pilares fundamentales del análisis de seguridad de la SDN, debido a que los activos representan los componentes fundamentales sobre los cuales la infraestructura de red aloja sus funcionalidades. Al realizar la identificación, se pueden determinar los puntos críticos de la red como lo son los switches, los enlaces de red, las aplicaciones y los sistemas de gestión del controlador, de modo que este proceso permite evaluar la interdependencia entre los diferentes componentes y su importancia para el funcionamiento global de la red.

La Tabla 29

Tabla *29* proporciona detalles sobre los activos afectados por los ataques DoS qu se originaron en el primer escenario del testbed. El contenido de la información radica en

ID del Activo	Activo	Tipo Activo	Amenaza	Vulnerabilidad	Descripción
Act-HSDN-001	Controlador SDN	Software y hardware	No disponibilidad en el servicio	Falta de controles en el procesamiento de información de mensajes OpenFlow	No disponibilidad en el ser debido a falta de controles procesamiento de informac mensajes OpenFlow sobre controlador SDN.
Act-HSDN-002	Switches SDN	Red	Degradación de los servicios de red	Falta de gestión de vulnerabilidades de capacidades del buffer de los conmutadores	Degradación de los servicio debido a falta de gestión de vulnerabilidades de capaci buffer de los conmutadores activo switches SDN.
Act-HSDN-003	Enrutadores c7200	Hardware	Problemas de detección de cantidad masiva de tráfico ICMP	Falta de generación y monitoreo de tráfico ICMP concurrente a los enlaces.	Problemas de detección de masiva de tráfico ICMP del falta de generación y monit tráfico ICMP concurrente a enlaces sobre el activo enru c7200.
Act-HSDN-004	Gestión paquetes de red del controlador Ryu	Servicio	Ataque informático DoS ICMP flooding	Configuración débil en el manejo de tráfico ICMP masivo	Ataque informático DoS IC flooding debido a configura en el manejo de tráfico ICM sobre el activo gestión paq red del controlador Ryu.
Act-HSDN-005	Servicios de VoIP, FTP, Web y VoD	Servicio	No disponibilidad de servicios de red	Falta de controles de seguridad en la red híbrida SDN/MPLS	No disponibilidad de servic debido a falta de controles seguridad en la red híbrida SDN/MPLS sobre el activo de VoIP, FTP, Web y VoD

proporcionar un registro detallado de activos, amenazas y vulnerabilidades relacionadas a la infraestructura de la red híbrida SDN que fue comprometida bajo ataques DoS de tipo inundación ICMP, de modo que su finalidad corresponde en identificar y documentar los riesgos y debilidades en la seguridad de la red híbrida SDN bajo este tipo de ataques de denegación de servicio.

Tabla 29
Identificación de activos comprometidos por los ataques DoS

ID del Activo	Activo	Tipo Activo	Amenaza	Vulnerabilidad	Descripción	Componente
Act-HSDN-001	Controlador SDN	Software y hardware	No disponibilidad en el servicio	Falta de controles en el procesamiento de información de mensajes OpenFlow	No disponibilidad en el servicio debido a falta de controles en el procesamiento de información de mensajes OpenFlow sobre el activo controlador SDN.	Máquina virtual que aloja el controlador Ryu.
Act-HSDN-002	Switches SDN	Red	Degradación de los servicios de red	Falta de gestión de vulnerabilidades de capacidades del buffer de los conmutadores	Degradación de los servicios de red debido a falta de gestión de vulnerabilidades de capacidades del buffer de los conmutadores sobre el activo switches SDN.	Open Vswitches
Act-HSDN-003	Enrutadores c7200	Hardware	Problemas de detección de cantidad masiva de tráfico ICMP	Falta de generación y monitoreo de tráfico ICMP concurrente a los enlaces.	Problemas de detección de cantidad masiva de tráfico ICMP debido a falta de generación y monitoreo de tráfico ICMP concurrente a los enlaces sobre el activo enrutadores c7200.	Router: R1 int g2/0 Router: R2 int g0/0
Act-HSDN-004	Gestión paquetes de red del controlador Ryu	Servicio	Ataque informático DoS ICMP flooding	Configuración débil en el manejo de tráfico ICMP masivo	Ataque informático DoS ICMP flooding debido a configuración débil en el manejo de tráfico ICMP masivo sobre el activo gestión paquetes de red del controlador Ryu.	Aplicación del controlador Ryu que gestiona el tráfico de la red SDN.
Act-HSDN-005	Servicios de VoIP, FTP, Web y VoD	Servicio	No disponibilidad de servicios de red	Falta de controles de seguridad en la red híbrida SDN/MPLS	No disponibilidad de servicios de red debido a falta de controles de seguridad en la red híbrida SDN/MPLS sobre el activo servicios de VoIP, FTP, Web y VoD	Servidor de VoIP Servidor de FTP, Web y VoD.

b) Evaluación de amenazas STRIDE

La sección de evaluación se centra en identificar y evaluar las posibles amenazas a la seguridad de una red híbrida SDN. Este enfoque del marco de trabajo STRIDE divide las amenazas en seis categorías principales, las cuales fueron resumidas en la Tabla 30:

Tabla 30
Amenazas registradas en la metodología STRIDE

Amenaza	STRIDE	Descripción
Suplantación de identidad	Spoofing	Falsificación de información para hacerse pasar por otra entidad legítima, engañando a sistemas o usuarios.
Manipulación de datos	Tampering	Alteración no autorizada de datos, ya sea para modificar su contenido o su integridad, comprometiendo su fiabilidad.
Repudio	Repudiation	Negación por parte de una entidad de haber realizado una acción.
Divulgación de información	Information Disclosure	Exposición no autorizada de datos confidenciales o sensibles, poniendo en riesgo la privacidad y seguridad de la información.
Denegación de Servicio	Denial of Service	Intención de bloquear o reducir la disponibilidad de un servicio, impidiendo que los usuarios legítimos accedan a él.
Elevación de Privilegios	Elevation of Privilege	Obtención no autorizada de mayores privilegios o permisos en un sistema, permitiendo realizar acciones no permitidas.

Considerando la información de la Tabla 30 sobre cada uno de los componentes de amenazas STRIDE que identifica la metodología, el siguiente apartado de esta sección corresponde en analizar cada una de las categorías en relación con las amenazas identificadas en los activos.

En primer lugar, es necesario identificar mediante la nomenclatura Am-HSDN-# las amenazas registradas en los activos de la red híbrida SDN/MPLS, de modo que en la nomenclatura el número de identificación del activo registrado en la Tabla 29 se mantiene, debido a que cada número está asociado al activo comprometido. La Tabla 31 enlista las

amenazas de cada activo para posteriormente ser identificados mediante la matriz de amenazas STRIDE.

Tabla 31
Amenazas identificadas en cada activo de la red híbrida SDN

Activo	Amenaza
Am-HSDN-001	No disponibilidad en el servicio
Am-HSDN-002	Degradación de los servicios de red
Am-HSDN-003	Problemas de detección de cantidad masiva de tráfico ICMP
Am-HSDN-004	Ataque informático DoS ICMP flooding
Am-HSDN-005	No disponibilidad de servicios de red

Una vez que se ha identificado cada una de las amenazas que presentan las activos comprometidos de la red SDN, se establece la Tabla 32 la cual presenta una matriz de evaluación de riesgos que analiza las amenazas de los activos con las diferentes amenazas STRIDE, de modo que los puntajes presentados en la tabla analiza la amenaza del activo y la asocia al tipo de amenaza STRIDE a la que pertenece, por ende, el valor de 1 indica que la amenaza de seguridad conforma la categoría seleccionada, mientras que un valor de 0 indica que no lo está.

Tabla 32
Evaluación STRIDE de las amenazas presentes en la red híbrida SDN

Amenaza/Categoría STRIDE	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Am-HSDN-001	0	0	0	0	1	0
Am-HSDN-002	0	0	0	0	1	0
Am-HSDN-003	0	0	0	0	1	0
Am-HSDN-004	0	0	0	0	1	0
Am-HSDN-005	0	0	0	0	1	0
Total	0	0	0	0	5	0

La información recopilada por la Tabla 32 establece que con un puntaje total de 5 puntos todas las amenazas de seguridad identificadas están asociadas a la categoría

ID del Activo	Activo	Tipo Activo	Amenaza	Vulnerabilidad	Descripción
Act-HSDN-001	Controlador SDN	Software y hardware	No disponibilidad en el servicio	Falta de controles en el procesamiento de información de mensajes OpenFlow	No disponibilidad en el ser debido a falta de controles procesamiento de informac mensajes OpenFlow sobre controlador SDN.
Act-HSDN-002	Switches SDN	Red	Degradación de los servicios de red	Falta de gestión de vulnerabilidades de capacidades del buffer de los conmutadores	Degradación de los servicio debido a falta de gestión de vulnerabilidades de capaci buffer de los conmutadores activo switches SDN.
Act-HSDN-003	Enrutadores c7200	Hardware	Problemas de detección de cantidad masiva de tráfico ICMP	Falta de generación y monitoreo de tráfico ICMP concurrente a los enlaces.	Problemas de detección de masiva de tráfico ICMP del falta de generación y monit tráfico ICMP concurrente a enlaces sobre el activo enru c7200.
Act-HSDN-004	Gestión paquetes de red del controlador Ryu	Servicio	Ataque informático DoS ICMP flooding	Configuración débil en el manejo de tráfico ICMP masivo	Ataque informático DoS IC flooding debido a configura en el manejo de tráfico ICM sobre el activo gestión paq red del controlador Ryu.
Act-HSDN-005	Servicios de VoIP, FTP, Web y VoD	Servicio	No disponibilidad de servicios de red	Falta de controles de seguridad en la red híbrida SDN/MPLS	No disponibilidad de servic debido a falta de controles seguridad en la red híbrida SDN/MPLS sobre el activo de VoIP, FTP, Web y VoD

STRIDE de Denegación de Servicio. Esto sugiere que la red híbrida SDN está particularmente vulnerable a este tipo de ataques y que las medidas de mitigación específicas deben dirigirse hacia la prevención y detección de ataques DoS para proteger adecuadamente la infraestructura de la red.

c) Análisis de riesgo

La sección de análisis de riesgos proporciona una evaluación detallada de la probabilidad e impacto de las amenazas identificadas en la Tabla 30, las cuales fueron registradas en el apartado de identificación de activos pertenecientes a la Tabla 29

Tabla *29* y asociadas a la categoría STRIDE a la que pertenecen presentadas en la Tabla 32, de modo que este apartado de la metodología se encarga en presentar los niveles de riesgos que causan cada una de las amenazas a la red. En un inicio se requiere establecer criterios de ponderación para establecer el nivel de riesgo existente, no obstante, la metodología STRIDE no presenta un criterio de ponderación especifico a ser aplicado, de modo que ante tal necesidad la información de evaluación de riegos de la ISO 27001 se adapta a las necesidades de identificar componentes de evaluación para analizar los riesgos de la red SDN. Al tomar esta información como referencia de ponderación permite establecer la Tabla 33, la cual presenta los componentes de ponderación que serán utilizadas para el análisis de riesgos de seguridad de la SDN, siendo estos los siguientes:

Tabla 33
Componentes que permiten el análisis de riesgos

Componente	Definición	Ponderación
Amenaza (ID)	Identifica la amenaza de asociada al activo al que pertenece.	Nomenclatura de identificación de amenazas establecidas en la Tabla 31
Categoría (STRIDE)	Categoría a la que pertenece la amenaza según la metodología STRIDE.	Amenaza asociada a que componente de la categoría STRIDE pertenece y fueron identificadas en la Tabla 32
Valoración del impacto CIA	Valoración del impacto de la amenaza en los aspectos de confidencialidad, integridad y disponibilidad.	1 Impacto Bajo a componentes CIA 2 Impacto Medio a componentes CIA 3 Impacto Alto a componentes CIA
Impacto CIA	Impacto de la amenaza en los aspectos de confidencialidad, integridad y disponibilidad	Cálculo del impacto de la amenaza a componentes CIA mediante la expresión (C+I+A)/3, donde: -C,I,A: Corresponde al valor numérico definido para el impacto de la amenaza
Probabilidad de ocurrencia	Probabilidad de que la amenaza ocurra.	1 Muy probable 2 Medianamente probable

		3	Muy probable
Cálculo de riesgo	Resultado del cálculo que combina la valoración del impacto y la probabilidad de ocurrencia de la amenaza.	Cálculo del riesgo mediante la expresión Impacto_CID*Probabilidad, donde: - Impacto_CID: Corresponde al valor numérico del impacto CID de la amenaza. - Probabilidad: Valor numérico que indica que puede ocurrir la amenaza	
Nivel de riesgo	Nivel de riesgo asociado a la amenaza, basado en el cálculo de riesgo.	1<x<2 3<x<5 6<x<9	Nivel de riesgo Bajo Nivel de riesgo Medio Nivel de riesgo Alto

Obtenida la información de cada componente de evaluación para determinar el nivel de riesgo de las amenazas, el siguiente proceso corresponde al análisis de riesgos de estas amenazas identificadas en la Tabla 31, las cuales serán sujetas a los componentes de evaluación presentados en la Tabla 33. Este proceso permite establecer la Tabla 34, la cual tiene como objetivo organizar de manera clara y concisa la información relacionada con cada amenaza identificada. Además, facilita la evaluación de riesgos al proporcionar una visión general de factores clave como el impacto en la confidencialidad, integridad y disponibilidad de la información (CID) y la probabilidad de ocurrencia de cada amenaza.

La puntuación de probabilidad de concurrencia de las amenazas es determinada por el administrador de la red SDN debido a su acceso completo a la topología de la red, las políticas de seguridad implementadas y las anomalías en los flujos de datos que pueden presentarse dentro de la red definida por software.

Tabla 34

Análisis de riesgos de las amenazas de la red híbrida SDN

Amenaza	**Categoría**	**Valoración del impacto CIA**						**Impacto CIA**	**Probabilidad**		**Calculo nivel de riesgo**	**Nivel de riesgo**
ID	**STRIDE**	**Confidencialidad**		**Integridad**		**Disponibilidad**						
Am-HSDN-001	DoS	1	Bajo	3	Alto	3	Alto	2	3	Muy Probable	6	**Alto**
Am-HSDN-002	DoS	1	Bajo	3	Alto	3	Alto	2	3	Muy Probable	6	**Alto**

Am-HSDN-003	DoS	1	Bajo	2	Medio	2	Medio	1	3	Muy Probable	3	**Medio**
Am-HSDN-004	DoS	1	Bajo	3	Alto	3	Alto	2	3	Muy Probable	6	**Alto**
Am-HSDN-005	DoS	1	Bajo	1	Alto	3	Alto	1	3	Muy Probable	3	**Medio**

Finalmente, los niveles de riesgo obtenidos de la Tabla 34 revelan una situación preocupante en cuanto a la seguridad de la red híbrida SDN. Con la mayoría de las amenazas clasificadas como de nivel de riesgo "Alto", de modo que se destaca la urgencia de implementar medidas de seguridad para mitigar los riesgos identificados. Es evidente que la categoría predominante de amenazas es la de Denegación de Servicio (DoS), lo que subraya la importancia de fortalecer las defensas de la red contra este tipo de ataques.

3.4 Segundo escenario: Mecanismo de seguridad para ataques DoS

Esta sección del documento establece el diseño de la segunda topología del testbed correspondiente al mecanismo de seguridad para mitigación de ataques DoS de tipo inundación ICMP, la planificación del proceso de selección del IDS/IPS como mecanismo de seguridad, el proceso de implementación dentro del escenario y finalmente el análisis de su funcionamiento.

3.4.1 Diseño de la topología del mecanismo de seguridad ante ataques DoS

El diseño del segundo escenario del testbed toma como referencia el análisis de riesgos generados por los ataques DoS de tipo inundación a la red, los cuales fueron descritos en toda la sección 3.3.4, con el objetivo de presentar la información necesaria de las amenazas que presentan los activos de la red híbrida SDN y como estos pueden ser

vulnerados por los atacantes con el propósito de comprometer el funcionamiento de la red.

Considerando la premisa extraída de los apartados expuestos en la sección 3.3.4, que señalan cómo los ataques DoS comprometen la disponibilidad de la red SDN, se produce una consecuencia significativa que radica en que el tráfico legítimo no puede ser enrutado hacia su destino, debido a la incapacidad de los switches SDN para llevar a cabo funciones de conmutación si la red está siendo comprometida. Esta situación plantea una problemática grave que demanda acciones correctivas inmediatas. Por ello, en la metodología STRIDE, la cual es utilizada para el manejo de este tipo de repercusiones, considera en su cuarta fase la necesidad de diseñar contramedidas hacia este tipo de ataques a la red. El propósito de este apartado radica en establecer una estrategia efectiva de mitigación frente a las repercusiones generadas por los ataques DoS en la red híbrida SDN/MPLS, lo que a su vez asegura su operatividad y funcionamiento continuo.

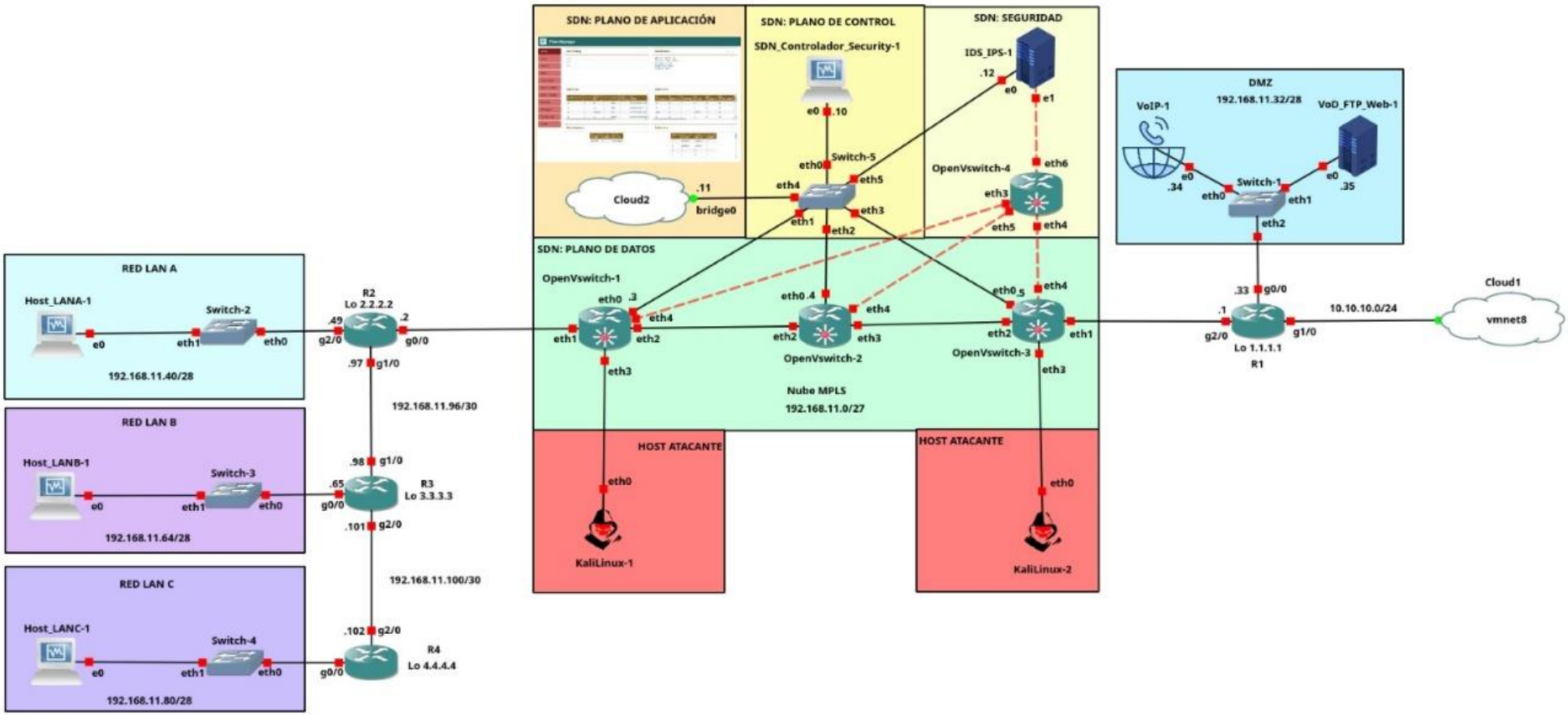
La implementación de un Sistema de Detección y Prevención de Intrusiones (IDS/IPS) en la red SDN/MPLS constituye una herramienta crucial para mitigar los ataques de denegación de servicio. Esto se debe a que en una red híbrida SDN, donde la gestión y control se centralizan en el controlador SDN, un IDS/IPS puede monitorear de manera proactiva el tráfico en busca de patrones anómalos que indiquen un ataque de inundación a la nube MPLS por parte de los nodos atacantes mediante reglas predefinidas establecidas permiten identificar, bloquear y mitigar estos flujos de tráfico malicioso; evitando así la congestión de la red y preservando su funcionalidad.

La Figura 35 presenta el diseño de la topología de red propuesta, la cual consta con un mecanismo de detección que analiza el tráfico sospechoso que el controlador SDN considere riesgoso y afecte el proceso de conmutación de los paquetes de los switches

SDN. Por ende, este concepto brinda a la red SDN la capacidad de adaptarse rápidamente a cambios en el tráfico que afecten el comportamiento de la red, de modo que el IDS/IPS en una arquitectura SDN permite tomar medidas correctivas de manera ágil para contrarrestar estos ataques manteniendo así la integridad y disponibilidad de los servicios que se alojan en la red.

Figura 42

Topología de red SDN/MPLS del escenario del mecanismo de seguridad propuesto



3.4.2 Selección del IDS/IPS

La selección de un sistema de detección y prevención de intrusiones para la mitigación de ataques DoS de tipo inundación ICMP en redes SDN/MPLS requiere una evaluación de parámetros relacionados al soporte del controlador con los softwares que poseen las funciones de IDS/IPS para integrarlas a la programación del controlador, de modo que al aprovechar la arquitectura centralizada de gestión en las redes SDN en conjunto con la implementación de funciones en código. brinda una oportunidad única para que mecanismos utilicen las características de control central para detectar y contrarrestar los ataques DoS. Esta integración aprovecha la visibilidad completa y el control centralizado de las redes SDN sobre los conmutadores para ofrecer una respuesta ágil y precisa ante incidentes de seguridad generados por la inundación ICMP de los atacantes.

En este contexto, la Tabla 35 detalla las características de los principales sistemas IDS/IPS disponibles, incluyendo su método de detección, la compatibilidad con diferentes sistemas operativos, su clasificación como sistema de detección y prevención de intrusiones basadas en red en conjunto con las funciones que permiten su integración en el controlador SDN Ryu.

Tabla 35

Comparativa de funciones de los principales sistemas IDS/IPS

IDS/IPS	Tipo	Funciones Principales	Arquitectura	Método de Detección	Integración de funciones en el código del controlador	Documentación para SDN
Snort	NIDS	Detección basada en reglas, análisis de protocolos y registro de paquetes.	Basada en reglas	Firmas y Anomalías	Si	Buena
Suricata	NIDS/ NIPS	Detección basada en firmas y análisis de protocolos.	Basado en reglas	Firmas y Anomalías	No	Media

La información recopilada en la Tabla 35 permite establecer la elección de Snort como el mecanismo de seguridad óptimo para hacer frente a los ataques de denegación de servicio, debido a sus capacidades de detección basada en reglas predefinidas, además un aspecto crucial para su selección radica en la capacidad de integración de sus funciones al controlador RYU. Esto se debe a que la información de la Tabla 25 indica que Snort es el único sistema entre los analizados que ofrece soporte para integración directa con el controlador, de modo que esta característica es crucial, pues permite una colaboración más eficiente y directa entre el sistema de detección de intrusiones y la infraestructura de red controlada por RYU, facilitando así una respuesta ágil y coordinada ante incidentes de seguridad.

3.4.3 Implementación del IDS en la topología

Esta sección recopila el criterio de implementación del mecanismo de seguridad en la red SDN, la configuración del mecanismo de seguridad que complementa con las funciones del controlador, el establecimiento de los parámetros de las reglas de flujo que permiten redigir el tráfico de hacia el IDS. Finalmente, se establecen las funciones de mitigación del ataque por parte del controlador, las cuales tienen como objetivo evitar que el tráfico DoS de los atacantes sature la disponibilidad de la red SDN, asegurando así su funcionamiento normal ante la presencia de este tipo de irrupciones de seguridad.

a) Criterio de implementación del mecanismo de seguridad

El criterio inicial de implementación del mecanismo se basa en aprovechar las capacidades de programación e implementación de reglas de flujo del controlador SDN hacia los conmutadores. Estas reglas especifican las acciones que los switches deben realizar según el tipo de tráfico que ingrese y se envíe a través de sus puertos. Esto garantiza que todos los paquetes de datos sean conmutados eficientemente a sus destinos dentro de la red. Además, esta estrategia permite duplicar el tráfico y enviarlo a

mecanismos de seguridad, como los IDS, encargados de detectar patrones anómalos en el tráfico analizado.

Este apartado proporciona una visión general del funcionamiento del mecanismo que será implementado, relacionando cada configuración realizada con su respectivo propósito dentro del sistema de seguridad. Sin embargo, la explicación detallada de las funciones se aborda en la siguiente sección.

La Figura 43 ilustra el funcionamiento del mecanismo de seguridad propuesto para la red SDN. En un inicio, el controlador configura reglas de flujo en los conmutadores con el objetivo de que estos realicen las funciones habituales de conmutación del tráfico que pasa por la red híbrida. Sin embargo, si se detecta un alto número de tráfico ingresando a la red en uno de los puertos de los conmutadores, el controlador establece una regla de flujo adicional en el conmutador afectado que redirige el tráfico sospechoso directamente al conmutador OpenVSwitch-4. A este conmutador llega todo el tráfico sospechoso redirigido por los conmutadores principales de la red SDN, de modo que este posee reglas de flujo específicas que filtran únicamente el tráfico proveniente de los switches y lo dirige al puerto conectado al IDS de Snort.

El IDS es una solución eficaz para mitigar el alto volumen de tráfico generado por un ataque DoS, debido a que su funcionamiento corresponde en detectar patrones anómalos en el tráfico de red en tiempo real. Al identificar el flujo de datos inusualmente grandes o repetitivos que son característicos de un ataque DoS, el IDS/IPS está configurado en base a reglas que se activarán si el número de solicitudes supera un umbral predefinido de paquetes por segundo. En caso de que la alarma se active, el IDS se comunica con el controlador y transmite información sobre el atacante, incluyendo su dirección MAC, dirección IPv4 y el tipo de mensaje paquete utilizado en el ataque.

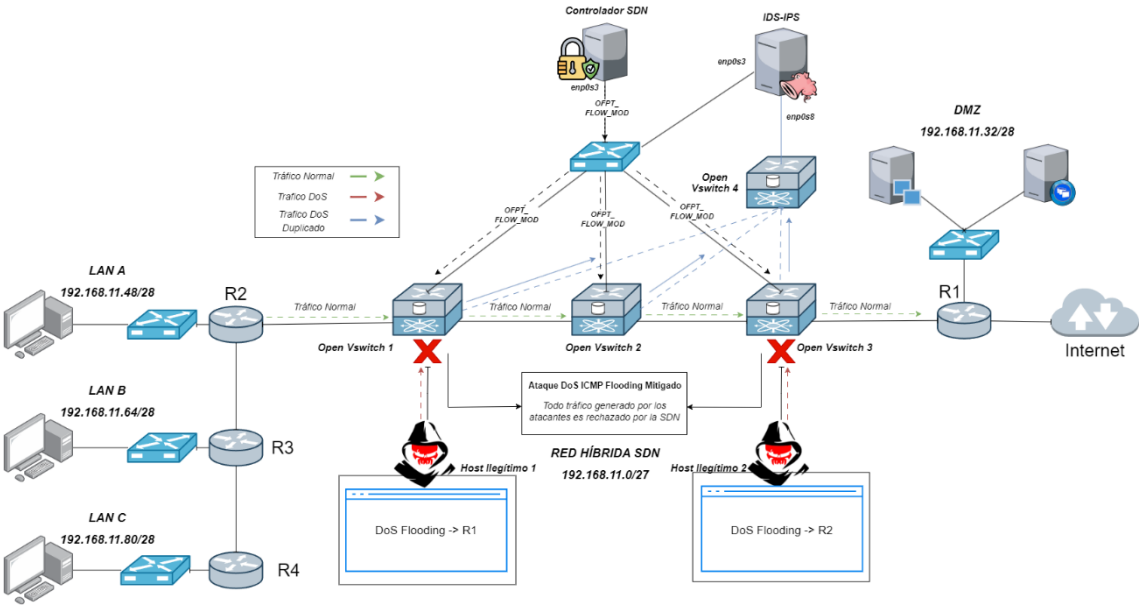
El controlador SDN establece la comunicación con el IDS mediante un socket interno e implementa en su código una función para recibir las alarmas generadas por SNORT. Estas funciones se implementan debido a que la información enviada por el IDS hacia el controlador se realiza en el mismo host, de modo que la interacción entre estos dos sistemas recae en el socket establecido para su comunicación. Al recibir la información sobre un atacante, el controlador SDN extrae de la alarma enviada la dirección IPv4 y la dirección MAC del atacante. Con esta información, el controlador puede establecer una función que niegue al host mal intencionado el uso de la infraestructura SDN para cualquier tipo de comunicación.

La función de bloqueo del tráfico a los atacantes dentro del controlador establece que, al identificar la dirección MAC de origen de los paquetes del ataque DoS que generaron la alarma en el IDS, el controlador SDN debe crear una regla de flujo que indique que todo el tráfico con la dirección MAC del atacante sea descartado al ingresar a la red SDN. Este bloqueo a nivel de dirección MAC se realiza debido a que los atacantes pueden falsificar las direcciones IPv4 de origen en los paquetes. Por consecuente, al realizar el bloqueo a nivel de dirección IP de origen generaría una cantidad masiva de flujos en los switches que descartarían los paquetes de las direcciones IPv4 falsas, no obstante, la red híbrida SDN no lograría mitigar su brecha de seguridad ante este tipo de ataques ya que el host malicioso aún estaría haciendo uso de la infraestructura SDN para ejecutar sus ataques.

Finalmente, el controlador genera una regla de flujo que descarta todo el tráfico del atacante, identificándolo mediante su dirección MAC. Posteriormente, actualiza las tablas de flujo de todos los conmutadores asociados a este mediante mensajes OpenFlow. Con esta regla de flujo, los conmutadores comienzan a mitigar el ataque al descartar todos los paquetes generados por el host malicioso. Como resultado, la inundación masiva de

tráfico hacia los activos de la red híbrida SDN se detiene, ya que los conmutadores rechazan todo tipo de tráfico proveniente del host que había irrumpido en el funcionamiento de la SDN.

Figura 43
Funcionamiento del mecanismo de seguridad



b) Instalación de Snort

La implementación del IDS inicia con la instalación de Snort dentro de la máquina virtual Ubuntu 20.04 que aloja el controlador, de modo que en un inicio es necesario la actualización de dependencias del sistema operativo Linux mediante la sintaxis `apt update.` Posteriormente, es necesario instalar la serie de librerías y dependencias que necesita el sistema para la ejecución de Snort, de modo que la **Tabla *36*** presenta los comandos utilizados para la instalación de dichas dependencias. Estos comandos incluyen la instalación de `build-essential` para asegurar que el sistema tiene las herramientas básicas de compilación, seguido por `libpcap-dev`, `libpcre3-dev`, y `libdumbnet-dev` para el manejo de paquetes y expresiones regulares necesarios en el procesamiento de datos de red.

Tabla 36
Comandos que permiten la instalación de Short

Comando	Finalidad
`sudo apt-get install build-essential`	Instala una serie de paquetes que son esenciales para compilar y construir software en sistemas basados en Debian/Ubuntu. Incluye compiladores como gcc y g++.
`sudo apt-get install libpcap-dev libpcre3-dev libdumbnet-dev`	Instala bibliotecas de desarrollo necesarias para el manejo de paquetes de red (libpcap), expresiones regulares (libpcre3) y funciones de red simplificadas (libdumbnet).
`sudo apt-get install bison flex`	Instala Bison y Flex, que son herramientas necesarias para procesar las gramáticas de entrada y generar los analizadores sintácticos.
`sudo apt-get install snort`	Instala Snort, un IDS/IPS que puede analizar el tráfico de red en busca de actividades maliciosas y registrar paquetes de red.

c) Sintaxis de reglas Snort

El sistema de Snort opera según un conjunto de reglas definidas, las cuales son cruciales para establecer parámetros de seguridad que notifiquen intrusiones de acuerdo con patrones anómalos detectados en el tráfico de red analizado por el IDS. La estructura típica de las reglas de Snort corresponde a la sintaxis: **`[action] [protocol] [sourceIP] [sourceport] -> [destIP] [destport]`** *`([Rule options])`*, la parte en negrita de la sintaxis se denomina encabezado y aparece en todas las reglas de Snort; la explicación detallada de cada uno de estos componentes que integran el encabezado se presenta en la Tabla 37.

Tabla 37
Especificaciones del encabezado de reglas Snort

Sintaxis	Descripción
`[action]`	Define la acción que Snort debe ejecutar si un paquete coincide con los criterios especificados en la regla.
`[protocol]`	Especifica el protocolo que se examina en busca de actividades sospechosas. Snort puede analizar los protocolos como TCP, UDP, ICMP e IP.
`[sourceIP]`	Dirección IP de origen que puede ser especificada directamente o mediante una variable

`[sourceport]`	Puerto de origen que también puede ser fijo o variable.
`->`	Operador de dirección que establece la dirección que debe seguir el tráfico de paquetes para que se active la regla.
`[destIP]`	Dirección IP de destino que puede ser una dirección específica o una variable.
`[destport]`	Puerto de destino que puede ser fijo o variable.

Nota. Adaptado de (SNORT, 2024)

Adicionalmente a los elementos presentes en el encabezado de la regla, las reglas de Snort pueden incorporar opciones adicionales, las cuales se especifican al final de la regla dentro de paréntesis *`([Rule options])`*. Si se incluyen múltiples opciones, estas se delimitan con punto y coma, de modo que la Tabla 38 describen las opciones adicionales que pueden ser añadidas dentro de una regla de Snort.

Tabla 38
Sintaxis adicionales de reglas Snort

Sintaxis	Descripción
`message (msg)`	Una cadena de texto que Snort muestra cuando se activa una regla. Generalmente, este mensaje describe lo que la regla está detectando.
`Flow`	Especifica la dirección que debe seguir el tráfico de red para que se active la regla.
`reference`	Permite incorporar referencias a fuentes de información externas.
`classtype`	Utiliza una palabra clave para describir el impacto potencial del ataque que la regla busca detectar.
`sid/rev`	Identificador único asignado a cada regla y se usa habitualmente junto con el número de revisión (rev).
`detection`	Define criterios más detallados para la detección, como el contenido, el desplazamiento y el tamaño en bytes.
`type both`	Considera el tráfico en ambas direcciones, de modo que esto es útil para detectar patrones de tráfico anómalos bidireccionales.
`track by_dst`	Especifica que el seguimiento de los paquetes para la evaluación del umbral debe realizarse en función de la dirección IP de destino.
`count #`	Define el umbral numérico para la cantidad de eventos que deben ocurrir dentro del intervalo de tiempo especificado.
`seconds #`	Especifica el periodo de tiempo durante el cual se cuentan los eventos.

Nota. Adaptado de (SNORT, 2024)

El establecimiento correcto de la sintaxis de cada una de las reglas Snort se relaciona de forma directa con la eficacia del sistema de detección en identificar las diferentes intrusiones. Para asegurar que el IDS está funcionando correctamente, se configuró una regla de ejemplo diseñada para generar una alarma si detecta un paquete ICMP, de modo que la regla establecida es la siguiente: `alert icmp any any -> any any (msg:"ICMP TEST";sid:1000001;).`

De acuerdo con la sintaxis de las reglas de SNORT identificadas en la Tabla 37, la regla anterior se interpreta de la siguiente manera: el IDS emitirá alertas para los paquetes ICMP dirigidos a una dirección de la red interna, sin importar la dirección de origen ni los puertos empleados, de modo que, si un paquete satisface estas condiciones, se desplegará un mensaje informativo como se especifica en la opción msg, mientras que los demás parámetros ssid son identificadores que ayudan a clasificar las reglas.

d) Definición de reglas Snort para el segundo escenario

La definición de reglas dentro del IDS de Snort para el segundo escenario del testbed consisten en establecer reglas personalizadas con el objetivo de mitigar los riesgos causados por los ataques de inundación ICMP en un inicio a la red SDN, los cuales fueron identificados en la Tabla 34. Además, estas reglas deben ser alineadas a las capacidades que poseen los atacantes para falsificar direcciones IP y el potencial daño que los ataques de inundación ICMP pueden causar a la infraestructura de red SDN, de modo que resulta crucial establecer medidas de seguridad dentro del IDS.

Dentro del establecimiento de reglas en el IDS, se definen cuatro reglas principales que alertan al controlador sobre los ataques DoS de tipo inundación ICMP que pueden afectar la red. La Tabla 39 presenta la información de estas reglas junto con la finalidad que se requiere dentro del IDS. Inicialmente, cada regla utiliza la

nomenclatura "Rule_#", donde "Rule" se refiere a la identificación general de las reglas y "#" corresponde al número de la regla. Posteriormente, estas reglas serán traducidas del lenguaje natural a la sintaxis de Snort.

Tabla 39
Finalidad de las reglas del IDS en lenguaje natural

ID	Finalidad
Rule_1	Detectar y alertar sobre un volumen inusualmente alto de tráfico ICMP, indicativo de un ataque de inundación ICMP.
Rule_2	Identificar y alertar sobre mensajes ICMP de redirección, que podrían ser utilizados para manipular el tráfico de la red SDN.
Rule_3	Detectar ataques de inundación ICMP Echo Request, los cuales intentan abrumar la red con solicitudes de echo ICMP.
Rule_4	Identificar y alertar sobre mensajes ICMP que indican la inaccesibilidad de un destino, posible indicador de intentos de ataque dirigidos a dispositivos específicos en la red SDN.

La implementación de las reglas en el IDS se fundamenta en la información proporcionada en la Tabla 29, debido a que en un inicio las reglas están diseñadas para describir en lenguaje natural su funcionamiento, no obstante, el sistema SNORT funciona en base a reglas que poseen una sintaxis definida, por lo que será necesario establecer las reglas dentro de estas especificaciones.

Para ello, los campos del encabezado de cada regla se definen inicialmente utilizando la información de la Tabla 37 y son adaptadas para identificar el protocolo ICMP empleado por los hosts maliciosos en los ataques de denegación de servicio realizados a la red SDN. Además, se incorporan parámetros adicionales en cada regla, de acuerdo con las sintaxis presentadas en la Tabla 38, con el fin de establecer umbrales para las alertas del IDS. Estos umbrales determinan la cantidad de tráfico que se analiza y se aplican contando los eventos basados en la dirección IP de destino del tráfico de inundación generado hacia los objetivos de los atacantes.

Tabla 40
Reglas de Snort para mitigación de ataques ICMP

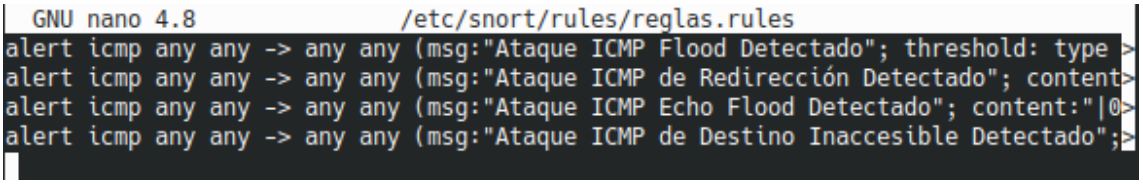
ID	Regla de Snort
Rule_1	`alert icmp any any -> any any (msg:"Ataque ICMP Flood Detectado"; threshold: type both, track by_dst, count 100, seconds 1; sid:1000001; rev:2;)`
Rule_2	`alert icmp any any -> any any (msg:"Ataque ICMP de Redirección Detectado"; content:"|05 00 00 00|"; threshold: type both, track by_dst, count 5, seconds 1; sid:1000002; rev:2;)`
Rule_3	`alert icmp any any -> any any (msg:"Ataque ICMP Echo Flood Detectado"; content:"|08 00 00 00|"; threshold: type both, track by_dst, count 100, seconds 1; sid:1000003; rev:2;)`
Rule_4	`alert icmp any any -> any any (msg:"Ataque ICMP de Destino Inaccesible Detectado"; content:"|03 00 00 00|"; threshold: type both, track by_dst, count 10, seconds 1; sid:1000004; rev:2;)`

e) Implementación de las reglas en el IDS Snort

Una vez que se han definido las reglas que permitan al IDS identificar el tipo de tráfico malicioso generado por los atacantes, el siguiente proceso consiste en la implementación de estas reglas dentro del sistema de Snort. Para lograr que las reglas definidas previamente sean aplicadas y estén operativas en el IDS, es necesario, en primer lugar, definir un archivo denominado `reglas.rules`. Este archivo tiene como función principal servir como repositorio centralizado donde se especifican todas las reglas de detección de intrusiones que el sistema Snort debe aplicar durante el análisis del tráfico de red.

La Figura 44 presenta el contenido de dicho archivo, el cual deberá ubicarse en la ruta `/etc/snort/rules`. Esta ruta es la encargada de almacenar todas las reglas de Snort que el sistema utilizará para la detección de intrusiones. Al colocar el archivo `reglas.rules` en esta ubicación específica, Snort podrá acceder de manera directa a las reglas definidas y aplicarlas durante el monitoreo del tráfico de red.

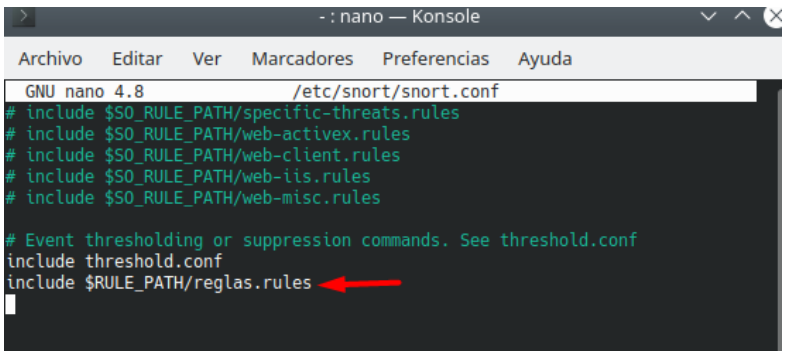
Figura 44
Reglas personalizadas de detección de ataques DoS ICMP



Una vez que las reglas han sido almacenadas en el archivo `.rules`, el siguiente paso en la configuración del IDS corresponde en añadir la ruta donde se almacenan las reglas establecidas al archivo de configuración de Snort denominado `snort.conf`, el cual se aloja dentro de la ruta `/etc/snort/`. La funcionalidad que posee este archivo corresponde a que es el encargado de determinar cómo Snort analiza el tráfico de red, qué reglas de detección debe aplicar, cómo debe registrar los eventos y cómo responder a las alertas al estar en funcionamiento.

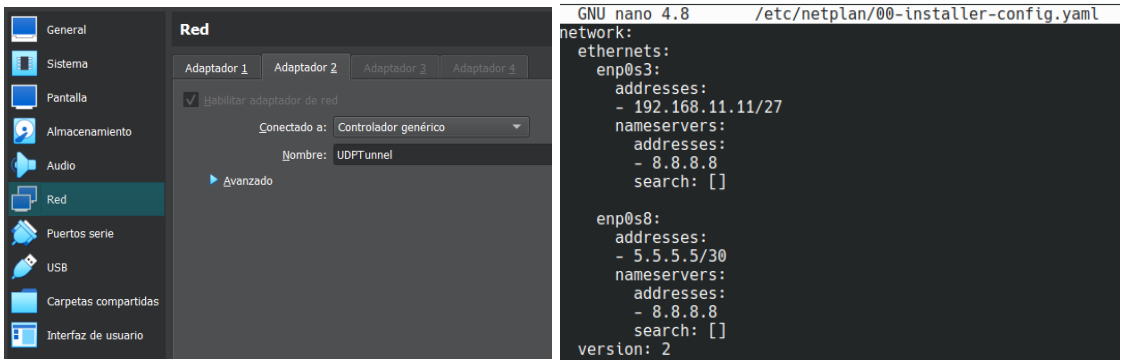
Una vez que se ha identificado la función del archivo `snort.conf` en el funcionamiento del IDS, el siguiente paso es configurarlo para que incluya y aplique las reglas definidas en el documento `reglas.rules`. La Figura 38 presenta el proceso detallado que debe llevarse a cabo dentro del archivo `snort.conf`. Esto implica indicar en la sección final del archivo que el sistema debe incorporar las reglas almacenadas en la ubicación especificada por la variable `$RULE_PATH`. Esta variable apunta a la ruta `/etc/snort/rules`, donde residen las reglas predefinidas del sistema.

Figura 45
Inclusión de las reglas en el archivo de configuración de Snort



Para completar la configuración del IDS en el controlador SDN, se agrega una interfaz de red adicional a la máquina virtual que aloja el IDS, dedicada exclusivamente a Snort para el análisis de tráfico en modo promiscuo. Al operar en este modo, la interfaz captura todos los paquetes de la red sin necesidad de que la dirección IP esté dentro del rango de la red del controlador, ya que no requiere una dirección específica para recibir el tráfico. Por esta razón, se asigna una dirección IPv4 cualquiera, fuera del rango de la red, para evitar que los atacantes puedan identificar esta interfaz como un blanco adicional, en base a este criterio la Figura 46 presenta el proceso de integración de la interfaz en la máquina virtual del IDS y la asignación de la IP correspondiente.

Figura 46
Configuración de la interfaz del IDS Snort

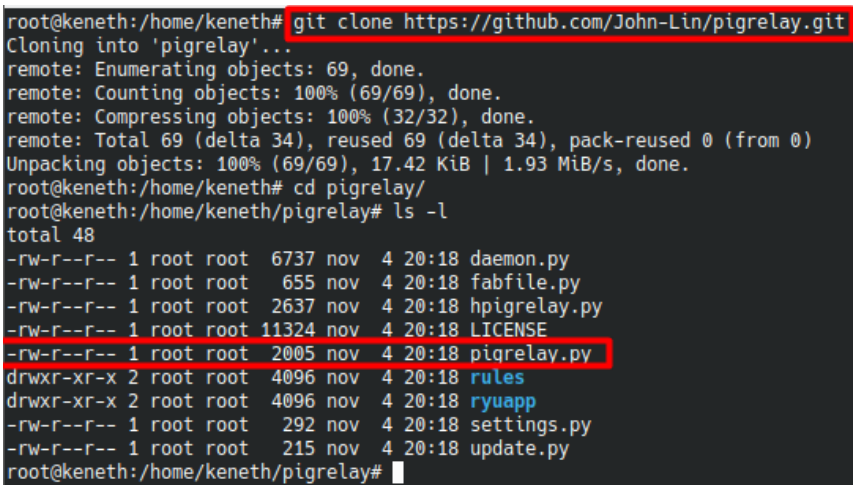


Adicionalmente, el mecanismo de seguridad requiere obtener las funciones de notificación al controlador ante cualquier intrusión de seguridad que pueda suscitarse en la red SDN. Para conseguir esto, es necesario añadir la aplicación denominada *PigRelay* a las funciones del IDS, debido a que este conjunto de scripts escritos en Python permite el envió de las alertas generadas por Snort hacia el controlador para iniciar el proceso de mitigación. La Figura 47 presenta en recuadros rojos las fases iniciales que permiten obtener esta aplicación dentro del mecanismo de seguridad, en el primer recuadro se muestran los comandos utilizados para obtener los archivos de la aplicación directamente

desde su repositorio de GitHub, mientras que el segundo recuadro indica el archivo que será modificado en base a la información del controlador SDN.

Figura 47

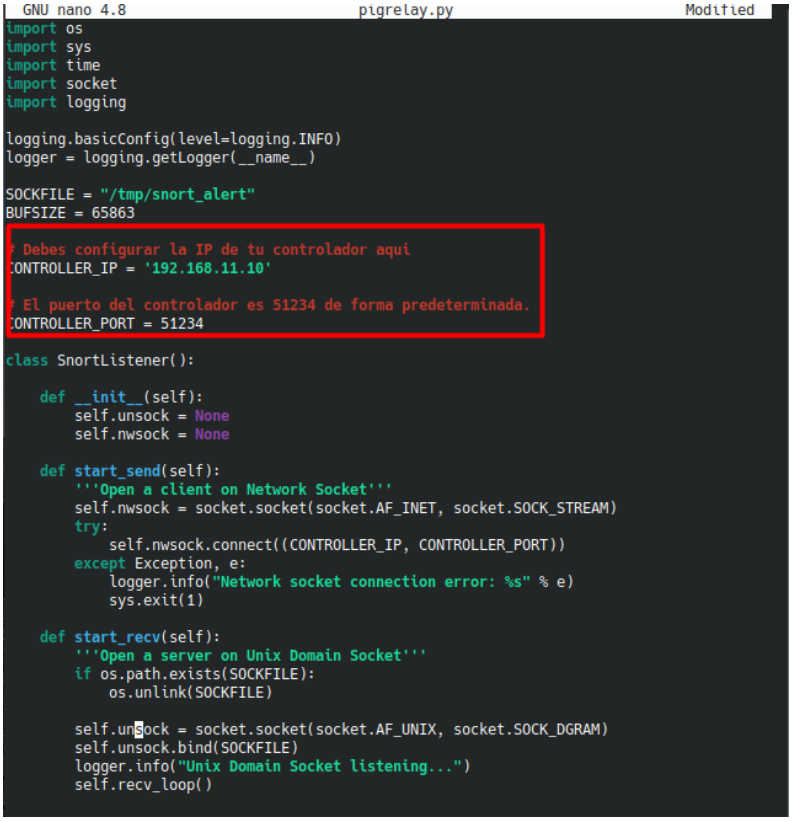
Descarga de la aplicación PigRelay para el IDS



Cabe añadir que la finalidad del archivo pigrelay,py corresponde en implementar un servicio que, a través de un Unix Domain Socket, recibe las alertas generadas de Snort y las reenvía hacia el controlador SDN a través de una conexión TCP establecida. En la Figura # se presenta la estructura del código, que en este caso las secciones remarcadas en rojo corresponden a la dirección IPv4 del controlador SDN en conjunto con el número de socket TCP utilizado para enviar las alertas generadas de Snort.

De este modo, la clase SnortListener establece tanto un socket de red mediante TCP hacia el controlador como un socket interno de dominio Unix para la recepción de alertas locales generados por Snort en su análisis de tráfico. Así, en un bucle continuo, el servicio escucha los datos del IDS y los envía al controlador, permitiéndole actuar en tiempo real en base a las alertas detectadas, facilitando una comunicación constante y automática entre el sistema de detección de intrusos y la red gestionada por el controlador, lo que resulta en una respuesta rápida y coordinada frente a posibles amenazas.

Figura 48
Configuración de la aplicación pigrelay para envio de alertas



```
 GNU nano 4.8                          pigrelay.py                          Modified
import os
import sys
import time
import socket
import logging

logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

SOCKFILE = "/tmp/snort_alert"
BUFSIZE = 65863

# Debes configurar la IP de tu controlador aqui
CONTROLLER_IP = '192.168.11.10'

# El puerto del controlador es 51234 de forma predeterminada.
CONTROLLER_PORT = 51234

class SnortListener():

    def __init__(self):
        self.unsock = None
        self.nwsock = None

    def start_send(self):
        '''Open a client on Network Socket'''
        self.nwsock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        try:
            self.nwsock.connect((CONTROLLER_IP, CONTROLLER_PORT))
        except Exception, e:
            logger.info("Network socket connection error: %s" % e)
            sys.exit(1)

    def start_recv(self):
        '''Open a server on Unix Domain Socket'''
        if os.path.exists(SOCKFILE):
            os.unlink(SOCKFILE)

        self.unsock = socket.socket(socket.AF_UNIX, socket.SOCK_DGRAM)
        self.unsock.bind(SOCKFILE)
        logger.info("Unix Domain Socket listening...")
        self.recv_loop()
```

Nota. El archivo completo se encuentra disponible en el repositorio indicado en el Anexo 7

Después de configurar la aplicación que permite la comunicación del IDS con el controlador, el siguiente paso corresponde en iniciar el funcionamiento del mecanismo de seguridad con el propósito de que este inicie el monitoreo del tráfico de red, el cual el controlador ha redirigido como potencial amenaza. Esto se logra ejecutando el comando `snort -A console -i enp0s8 -A unsock -l /tmp -c /etc/snort/snort.conf`, donde la función de cada argumento de la sintaxis presentada corresponde con los parámetros presentados en la Tabla 41.

Tabla 41
Comandos de funcionamiento de Snort

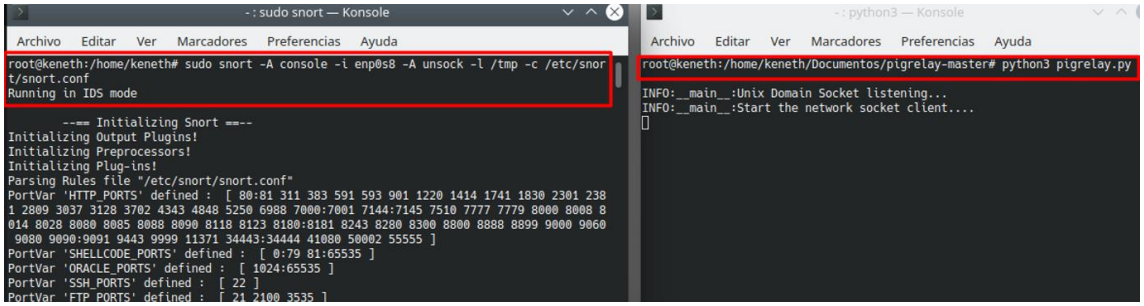
Sentencia	Funcionalidad
snort	Comando para ejecutar el software de Snort.
-A console	Especifica que las alertas generadas deben ser enviadas y a la vez mostradas en consola.
-i enp0s8	Indica la interfaz en la que debe escuchar el IDS.

-A unsock	Envía las alertas utilizando sockets de Unix.
-l /tmp	Indica el directorio donde se quieren guardar los logs.
-c /etc/snort/snort.conf	Señala el directorio donde se encuentra el fichero de configuración de Snort.

Finalmente, para conseguir el funcionamiento en conjunto del IDS y la aplicación que envía las alertas al controlador, es necesario ejecutar ambos programas en paralelo en la terminal de comandos, como se muestra en la Figura 49. De este modo, Snort inicia el monitoreo del tráfico en modo promiscuo, detectando posibles ataques, mientras que el IDS establece una conexión TCP con el controlador SDN que, en caso de detectar tráfico sospechoso, este envíe alertas al controlador posibilitando una respuesta a las amenazas detectadas en la red.

Figura 49

Ejecución de Snort y aplicación de envió de alertas en el IDS



f) Establecimiento de reglas de flujo en el controlador

El establecimiento de reglas de flujo corresponde un componente principal en la implementación del mecanismo de seguridad, debido a que el controlador será el encargado de realizar acciones correctivas al identificar las alarmas generadas por el IDS respecto a los ataques de denegación de servicio, por ende, este elemento de la SDN debe poseer las capacidades de conmutación del tráfico normal que se genera sobre la red híbrida y en paralelo mitigar las intenciones maliciosas de los hosts que buscan comprometer la disponibilidad de la red.

A continuación, se presentan las funciones implementadas en el controlador para establecer la solución de seguridad, no obstante, el código completo denominado testbed_escenario02.py de la solución propuesta en el presente trabajo se encuentra disponible para su acceso y visualización en el Anexo 7 que presenta el enlace del repositorio de Github destinado para este proyecto.

En un inicio las primeras funciones que realizará el controlador corresponden en preparar la aplicación para su interacción con las alertas generadas por el IDS del sistema Snort y a la vez gestionar la conmutación de paquetes en la red. La explicación de las funciones de mitigación del controlador SDN está estructurada en secciones para facilitar su comprensión, de modo que cada sección cumple una funcionalidad especifica dentro de la estructura general del código, comenzando con la detección de la intrusión de seguridad y posteriormente implementar medidas correctivas de mitigación para asegurar el funcionamiento de la red SDN.

El primer apartado en ser analizado en el código del controlador corresponde a la Sección 1, donde se presenta la clase SimpleSwitch13. Esta clase es una aplicación del controlador Ryu para redes SDN que integra funcionalidades de Snort para la detección de intrusiones a la red. La clase utiliza contextos para las bibliotecas snortlib y dpset, indicadas en el diccionario _CONTEXTS, lo que permite acceder a sus funcionalidades de manera estructurada en las secciones posteriores del código que requieren estas funciones. En el método init, se inicializan estas bibliotecas a través de los argumentos kwargs, configurando así la librería snortlib para permitir la interacción conjunta de las funciones de Snort y dpset en la gestión de los switches administrados por el controlador.

Se define un puerto específico, el número 5, para Snort, que será el puerto de los switches con conexión a la interfaz del IDS, lo que permite que este puerto se encargue

de enviar el tráfico de red sospechoso desde cualquier switch hacia el IDS. Además, en el código se establece un diccionario que, en secciones posteriores, registrará las direcciones MAC de los puertos de cada conmutador asociado con el controlador.

El siguiente apartado en esta sección corresponde a un mecanismo para monitorear el tráfico TCP, UDP e ICMP en cada switch, con el objetivo de detectar posibles excesos de paquetes por segundo, lo cual es indicativo de una actividad sospechosa o un potencial ataque. Las variables tcp_packet_count, udp_packet_count e icmp_packet_count usan diccionarios con valores predeterminados en cero para llevar el conteo de este tipo de paquetes en cada switch, mientras que la variable denominada packet_threshold define un umbral de 100 paquetes por segundo, correspondiente al número de paquetes considerados como sospechosos dentro de la red. La variable time_window establece una ventana de tiempo de 1 segundo para que el conteo sea evaluado periódicamente, asimismo last_checked guarda el último tiempo en que se evaluó el tráfico en cada switch, de modo que el controlador pueda resetear los contadores y verificar de nuevo al iniciar cada nueva ventana de tiempo.

La última estructura de código corresponde en establecer un diccionario denominado socket_config con una configuración para el uso de sockets, de modo que al definir 'unixsock': False, se indica que no se utilizarán sockets Unix sino sockets TCP/IP, que en este caso es el socket TCP con número de puerto 51234 que permiten el envió de las alertas generadas por parte del IDS hacia el controlador a través de la red 192.168.0/27. Luego, esta configuración se aplica a la instancia de Snort mediante self.snort.set_config(socket_config), especificando que se usará un socket de red, para finalmente iniciar el servidor de sockets, permitiendo que Snort escuche conexiones remotas, lo cual facilita la recepción de las alertas provenientes del IDS.

Sección 1
Configuración del controlador con integración de Snort

```
### FUNCIONES BASICAS ###
### Inicializacion de la aplicacion del controlador###

class SimpleSwitchSnort(app_manager.RyuApp):
    OFP_VERSIONS = [ofproto_v1_3.OFP_VERSION]
    _CONTEXTS = {'snortlib': snortlib.SnortLib,'dpset': dpset.DPSet}

    # Definir variables
    def __init__(self, *args, **kwargs):
        super(SimpleSwitchSnort, self).__init__(*args, **kwargs)
        self.snort = kwargs['snortlib']
        self.snort_port = 5
        self.mac_to_port = {}
        self.dpset = kwargs['dpset']

        self.tcp_packet_count = defaultdict(int)
        self.udp_packet_count = defaultdict(int)
        self.icmp_packet_count = defaultdict(int)
        self.packet_threshold = 100  # Umbral de paquetes por segundo
        self.time_window = 1  # Ventana de tiempo en segundos
        self.last_checked = defaultdict(lambda: time.time())

        socket_config = {'unixsock': False}
        self.snort.set_config(socket_config)
        self.snort.start_socket_server()
```

Establecida la comunicación del IDS y el controlador, la siguiente función por analizar en la estructura del código del controlador corresponde a la Sección 2. Su función principal es iniciar el funcionamiento de los switches SDN mediante el uso del framework de Ryu, de modo que cuando un nuevo switch se conecta al controlador, se dispara el evento EventOFPSwitchFeatures, en el cual la función switch_features_handler maneja este evento configurando el switch para que envíe al controlador los paquetes que no coinciden con ninguna entrada en la tabla de flujo (table-miss).

Esto se logra instalando una regla de flujo (flow entry) con una prioridad de 0, que aplica una acción para enviar estos paquetes al controlador. De esta manera, el controlador puede decidir cómo manejar estos paquetes, permitiendo una gestión dinámica y centralizada del tráfico en la red.

Sección 2
Configuración Inicial de Switches en SDN

```
    ### Inicializacion de los switches ###
    # Evento configuracion nuevo switch
    @set_ev_cls(ofp_event.EventOFPSwitchFeatures, CONFIG_DISPATCHER)
    def switch_features_handler(self, ev):
        datapath = ev.msg.datapath
        ofproto = datapath.ofproto
        parser = datapath.ofproto_parser

        # Instalar entrada table-miss en el switch
        match = parser.OFPMatch()
        actions = [parser.OFPActionOutput(ofproto.OFPP_CONTROLLER,
                                          ofproto.OFPCML_NO_BUFFER)]
        self.add_flow(datapath, 0, match, actions)
```

Definidas las funciones de asociación de los switches al controlador, el siguiente apartado por analizar en la estructura corresponde a la Sección 3, el cual tiene como premisa gestionar los paquetes entrantes en los switches de la SDN controlados mediante el protocolo OpenFlow 1.3. La función _packet_in_handler es llamado cuando un evento EventOFPPacketIn es recibido, indicando la llegada de un paquete al switch. El código extrae información relevante del paquete correspondientes a las direcciones MAC de origen y destino, y el puerto de entrada del switch. Luego, registra la dirección MAC de origen y determina la acción apropiada basada en la dirección MAC de destino: reenviar el paquete al puerto correspondiente o inundar (flood) si el destino es desconocido.

Al momento de que el tráfico ingresa a la red, el paquete alojado en la variable (msg.data) es transformado en un objeto de paquete que permite la manipulación a nivel de protocolo mediante la sintaxis pkt.get_protocol, de modo que, dependiendo del tipo de tráfico que ingresa a la red, se obtienen las instancias de los protocolos Ethernet, IPv4, ICMP, TCP y UDP presentes en los paquetes. Este proceso es crucial para analizar el tráfico de red que pasa por la SDN y aplicar diferentes políticas de manejo según el tipo de protocolo detectado, lo que permite las acciones de redireccionamiento y bloqueo de los paquetes sospechosos en la red.

Adicionalmente, el controlador establece el registro de la hora actual cuando un paquete ingresa a la red y verifica si ha pasado la duración de la ventana de tiempo (time_window). Si es así, reinicia los contadores de paquetes TCP, UDP e ICMP asociados con el identificador del switch de acuerdo con su dpid para comenzar un nuevo conteo y actualizar la marca de tiempo. Posteriormente, evalúa cada paquete que contiene protocolos IPv4, TCP, UDP o ICMP, incrementando el contador correspondiente y comparando con un umbral (packet_threshold). Si el contador de algún tipo de paquete supera este umbral, el paquete se redirige a un puerto específico (snort_port) para su análisis en un sistema de detección de intrusiones de Snort. Finalmente, configura las acciones para el puerto de salida y asegura que el paquete se envíe al switch a través de un mensaje PacketOut.

Sección 3

Gestión y enrutamiento de paquetes entrantes identificando el tipo de tráfico

```
### Gestion de paquetes entrantes ###
    @set_ev_cls(ofp_event.EventOFPPacketIn, MAIN_DISPATCHER)
    def _packet_in_handler(self, ev):
        msg = ev.msg
        datapath = msg.datapath
        ofproto = datapath.ofproto
        parser = datapath.ofproto_parser
        in_port = msg.match['in_port']

        pkt = packet.Packet(msg.data)
        eth = pkt.get_protocols(ethernet.ethernet)[0]
        ipv4_pkt = pkt.get_protocol(ipv4.ipv4)
        icmp_pkt = pkt.get_protocol(icmp.icmp)
        tcp_pkt = pkt.get_protocol(tcp.tcp)
        udp_pkt = pkt.get_protocol(udp.udp)

        if eth.ethertype == ether_types.ETH_TYPE_LLDP:
            return
        dst = eth.dst
        src = eth.src

        dpid = format(datapath.id, "d").zfill(16)
        self.mac_to_port.setdefault(dpid, {})

        self.logger.info("Paquete en Switch:%s MACsrc:%s MACdst:%s
Port:%s", dpid, src, dst, in_port)

        self.mac_to_port[dpid][src] = in_port
```

```
        if dst in self.mac_to_port[dpid]:
            out_port = self.mac_to_port[dpid][dst]
        else:
            out_port = ofproto.OFPP_FLOOD

        actions = []
        current_time = time.time()

        if current_time - self.last_checked[dpid] > self.time_window:
            self.tcp_packet_count[dpid] = 0
            self.udp_packet_count[dpid] = 0
            self.icmp_packet_count[dpid] = 0
            self.last_checked[dpid] = current_time

        if ipv4_pkt and tcp_pkt:
            self.tcp_packet_count[dpid] += 1
            if self.tcp_packet_count[dpid] > self.packet_threshold:
               actions.append(parser.OFPActionOutput(self.snort_port))

        elif ipv4_pkt and udp_pkt:
            self.udp_packet_count[dpid] += 1
            if self.udp_packet_count[dpid] > self.packet_threshold:
               actions.append(parser.OFPActionOutput(self.snort_port))

        if ipv4_pkt and icmp_pkt:
            self.icmp_packet_count[dpid] += 1
            if self.icmp_packet_count[dpid] > self.packet_threshold:
          actions.append(parser.OFPActionOutput(self.snort_port))
        else:
            actions.append(parser.OFPActionOutput(out_port))

        if out_port != ofproto.OFPP_FLOOD:
            match = parser.OFPMatch(in_port=in_port, eth_dst=dst,
eth_src=src)
            if msg.buffer_id != ofproto.OFP_NO_BUFFER:
                self.add_flow(datapath, 1, match, actions,
msg.buffer_id)
                return
            else:
                self.add_flow(datapath, 1, match, actions)
        data = None
        if msg.buffer_id == ofproto.OFP_NO_BUFFER:
            data = msg.data
        out = parser.OFPPacketOut(datapath=datapath,
buffer_id=msg.buffer_id,in_port=in_port, actions=actions, data=data)
        datapath.send_msg(out)
```

Establecida la gestión de los paquetes por parte del controlador, la siguiente función del código se centra en cómo reconoce las alertas y realiza acciones de mitigación ante un ataque. La Sección 4 presenta la estructura definida para el manejo de las alertas

generadas por Snort. En este caso, se utiliza un decorador[14] @set_ev_cls para definir un controlador de eventos denominado _dump_alert. El código especifica que cuando se reciba un evento de alerta de Snort (snortlib.EventAlert), se ejecutará la función _dump_alert. Dentro de esta función, se extrae el mensaje del evento (ev.msg) y se convierte en un objeto de paquete (packet.Packet).

Posteriormente, el controlador obtiene el protocolo IPv4 del paquete, de modo que, si este es encontrado, el paquete es mostrado en consola mediante la función self.packet_print(msg.pkt) con la finalidad de presentar la información al administrador de la red. Finalmente, se llama a la función self.fix_alert(ev) para establecer la mitigación del ataque tras la recepción y procesamiento de la alerta por parte del controlador.

Sección 4
Recepción y procesamiento de alertas de Snort

```
### RECEPCION DE ALARMAS ###
    # Evento alerta de Snort
    @set_ev_cls(snortlib.EventAlert, MAIN_DISPATCHER)
    def _dump_alert(self, ev):
        msg = ev.msg

        pkt = packet.Packet(array.array('B', msg.pkt))
        _ipv4 = pkt.get_protocol(ipv4.ipv4)

        if _ipv4:
            self.packet_print(msg.pkt)
        self.fix_alert(ev)
```

El último apartado por ser analizado dentro del código corresponde a la función de mitigación de los ataques, en este caso la Sección 5 presenta un método llamado fix_alert, el cual se activa al recibir el evento de la alerta generada por el IDS en conjunto con la función de recepción que posee el controlador SDN. Este método extrae del evento generado el paquete de datos obteniendo la dirección IP y la dirección MAC de origen del atacante a partir de esa información, lo que conlleva a que se muestre estas direcciones

[14] Decorador: Patrón de diseño en Python que permite al usuario añadir nuevas funciones a un objeto existente sin modificar su estructura .

y se presenten en consola al administrador de la SDN, pasa posteriormente a iniciar el proceso de mitigación.

La fase inicial de la mitigación de ataques consiste en obtener la lista de todos los datapaths (o rutas de datos) de los switches asociados al controlador mediante el método self.dpset.get_all(). Este paso permite identificar el número total de switches en la red, de manera que el controlador pueda recorrer cada uno de ellos en un ciclo y aplicar la regla de mitigación correspondiente a todos los conmutadores SDN. En cada iteración del ciclo, el índice i aumenta para llevar un registro del número de switches en los que se ha implementado la nueva regla de flujo, lo que conlleva a que es necesario configurar un parser[15] de protocolos en cada switch, que se encargarán de construir la regla de flujo de coincidencia (match).

La regla de flujo añadida en cada conmutador está diseñada para bloquear todo el tráfico proveniente de la dirección MAC del atacante (eth_src=srcMAC). Para lograr esto, la lista de acciones (actions) se establece como vacía (actions = []), lo que implica que el tráfico coincidente no se redireccionará, sino que será bloqueado. Una vez definida la regla, la función add_flow la inserta en la primera tabla de flujos (table_id=0) de cada switch con una prioridad de 150, de modo que esta regla tenga precedencia sobre reglas menos prioritarias y sea evaluada de inmediato cuando el tráfico ingrese al switch. Al completar esta configuración, el sistema imprime un mensaje de confirmación que indica que la fase de mitigación ha finalizado.

Sección 5
Función de mitigación de los ataques por parte del controlador

```
### PROGRAMACION DE RESPUESTAS A LOS ATAQUES ###
    # Establecer la respuesta frente a alerta del Snort
    def fix_alert(self, ev):
        msg = ev.msg
```

[15] Parser: Herramienta que se utiliza en el procesamiento de lenguaje natural para analizar y comprender la estructura sintáctica de una frase o un texto.

```
pkt = msg.pkt
pkt = packet.Packet(array.array('B', pkt))
_ipv4 = str(pkt.get_protocol(ipv4.ipv4))
_mac = str(pkt.get_protocol(ethernet.ethernet))

srcIP = _ipv4.split("'")[3]
srcMAC = _mac.split("'")[3]

def print_box2(text):
    border = '+' + '-' * (len(text) + 2) + '+'
    print(border)
    print('| ' + text + ' |')
    print(border)

print_box2('IP Origen del atacante= %s' %srcIP)
print_box2('MAC Origen del atacante= %s'%srcMAC)

print_box2('Mitigacion Iniciada....')
print_box2('Bloqueando la MAC de origen: %s...' %srcMAC)

dp_set = self.dpset.get_all()

i = 0
for dp in dp_set:
    i += 1
    datapath = dp[1]

    parser = datapath.ofproto_parser
    match = parser.OFPMatch(eth_src=srcMAC)
    actions = []
    self.add_flow(datapath, 150, match, actions, table_id=0)

    print_box2('Regla de flujo anadida al switch: %d' %i)

print_box2('Mitigacion Terminada....')
```

3.4.4 Funcionamiento del mecanismo de seguridad

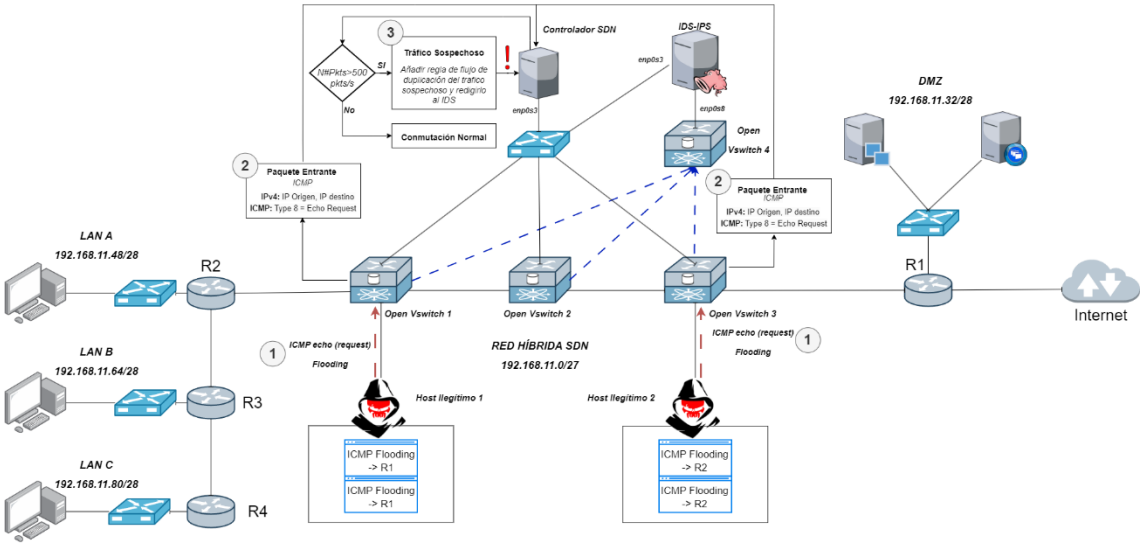
Establecido el funcionamiento del mecanismo de seguridad en el código del controlador, esta sección se encarga de presentar el funcionamiento del segundo escenario del testbed, de modo que para conseguir esta finalidad se utilizarán ilustraciones que representan a cada una de las fases que la solución de seguridad de la SDN utiliza para establecer la mitigación.

La Figura 50 muestra la situación inicial que activa la respuesta al ataque. Los hosts maliciosos dirigen los ataques a los enlaces de los enrutadores que se interconectan

con los switches SDN, inundándolos con un masivo número de tráfico ICMP, UDP y TCP dirigido a sus direcciones IP. Una vez que este tráfico ingresa a los conmutadores SDN, el controlador establece e inicia el umbral de número de paquetes por segundo y, si este es superado el tráfico es denominado como sospechoso. Caso contrario, si el umbral no es superado el controlador realiza la función de conmutación de paquetes de forma normal.

Figura 50

Fase de identificación de los ataques DoS



En el caso que el umbral ha sido superado, el controlador SDN inicia una respuesta coordinada para gestionar el flujo anómalo en la red, como se muestra en la Figura 51. El proceso comienza con la generación de una regla de flujo que instruye al conmutador afectado a duplicar y redirigir dicho tráfico al puerto que conecta con el conmutador Open vSwitch-4, el cual, a su vez, está enlazado directamente al sistema IDS. Esta configuración permite que el IDS analice el tráfico conforme a sus reglas predefinidas para la detección de ataques ICMP, TCP y UDP.

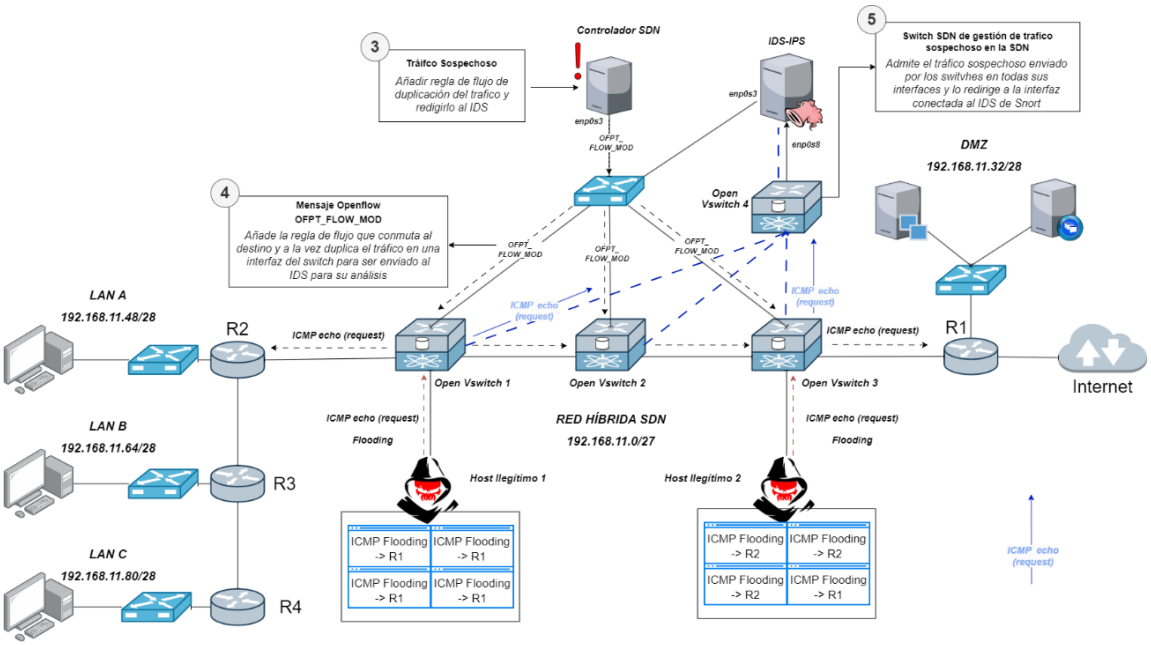
Para implementar esta funcionalidad, el controlador envía una instrucción mediante un mensaje OpenFlow OFPT_FLOW_MOD al conmutador que está recibiendo el tráfico sospechoso. Al recibir el mensaje, el conmutador actualiza su tabla de flujos

con la nueva acción, configurándose para duplicar el tráfico y redirigirlo al puerto específico conectado al IDS. No obstante, debido a la arquitectura de la red, el tráfico redirigido pasa primero por un conmutador intermedio, el Open vSwitch-4, el cual actúa como un enlace entre los conmutadores principales y el sistema IDS.

Este conmutador posee la responsabilidad de garantizar que el tráfico duplicado ingrese correctamente a la interfaz del IDS mediante sus propias reglas de flujo, que aseguran el encaminamiento adecuado y el cumplimiento de los requisitos de inspección de los ataques DoS de tipo inundación ICMP, TCP y UDP.

Figura 51

Fase de duplicación y envío del tráfico sospechoso al IDS

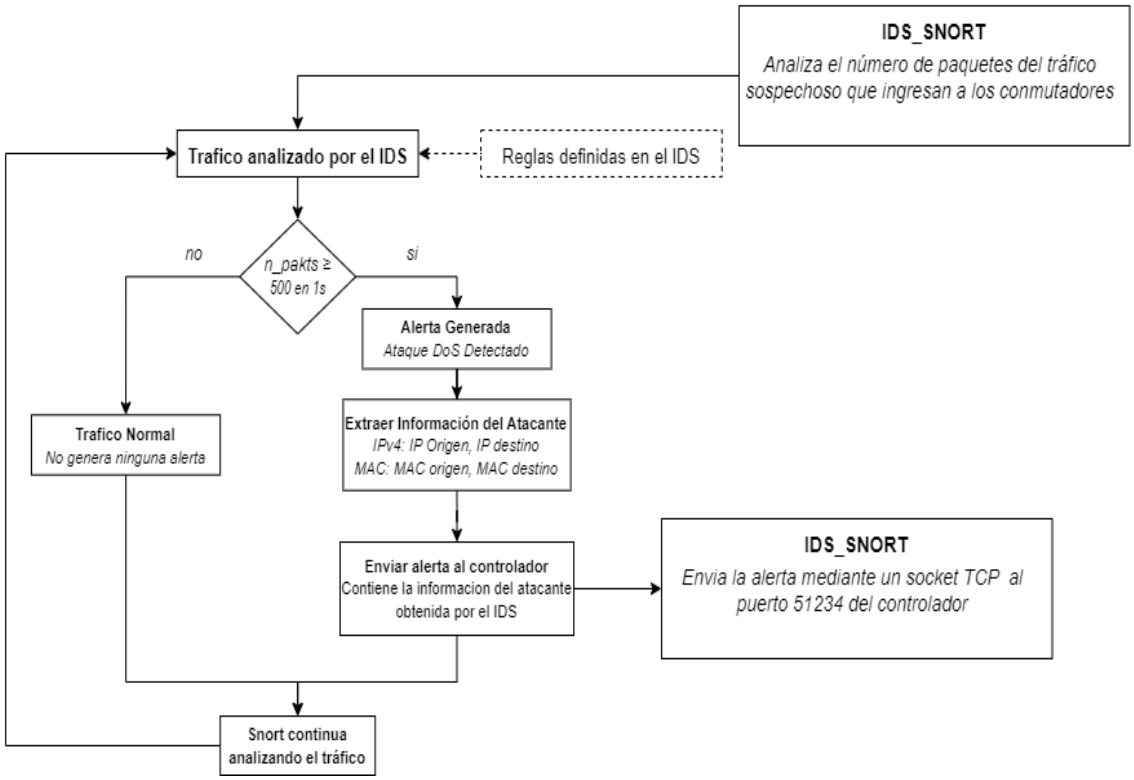


Una vez que los paquetes duplicados llegan al IDS, el software Snort analiza el tráfico para identificar patrones de ataque. La Figura 52 muestra el algoritmo que sigue el IDS para esta tarea, la cual indica que, si el número de paquetes coincidentes con una de sus reglas supera un umbral de 500 paquetes por segundo, el software de Snort clasifica la actividad como un ataque de denegación de servicio de tipo inundación en la red híbrida SDN. En respuesta, el IDS emite una alerta indicando la detección del ataque y extrae de

inmediato información detallada del atacante, incluyendo direcciones MAC e IP de origen y destino, junto con el tipo de tráfico generado.

La alerta es entonces enviada al controlador a través de un socket TCP en el puerto 51234 debido a que el controlador y el mecanismo de seguridad se encuentran en la misma red, desencadenando así el proceso de mitigación.

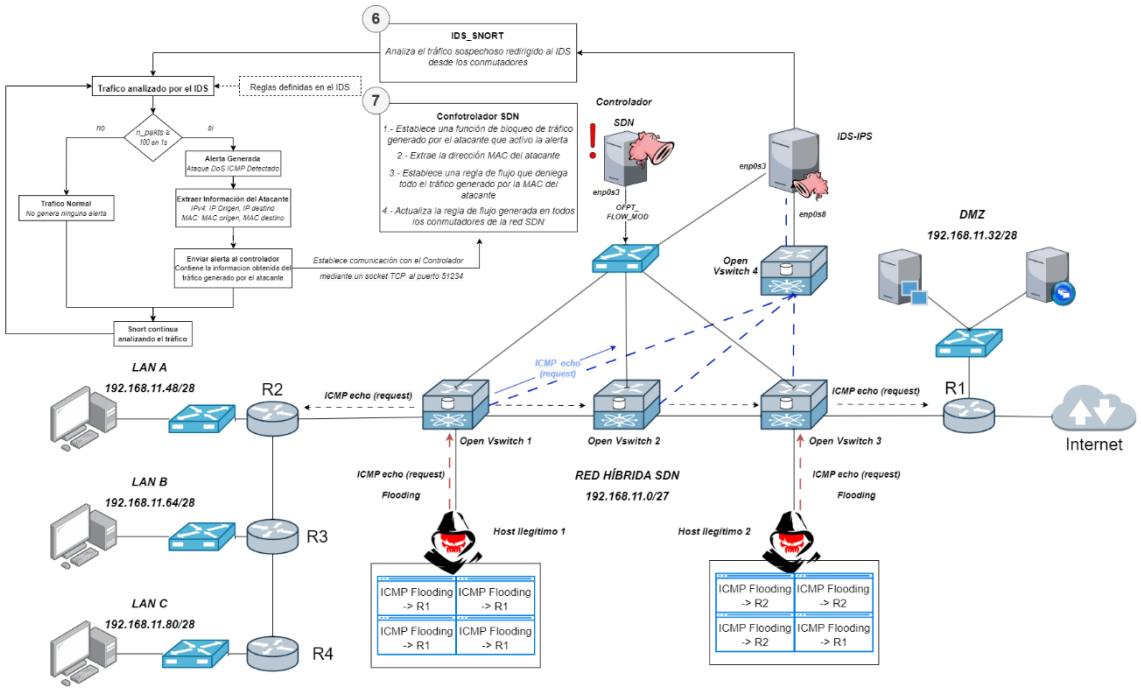
Figura *52*
Algoritmo de generación de alertas del IDS



La **Figura 53** presenta el proceso posterior a la alerta generada por el IDS, la cual consiste en que la información del atacante es envía al controlador SDN mediante un socket TCP de red al controlador SDN alojado en la misma red. Este al recibir la información de la alerta generada por el IDS, establece una función para bloquear el tráfico del atacante, la cual consiste en configurar reglas de flujo en los switches que deniega todo el tráfico generado por parte del atacante mediante su dirección MAC.

Figura 53

Fase de alerta y comunicación del IDS con el controlador

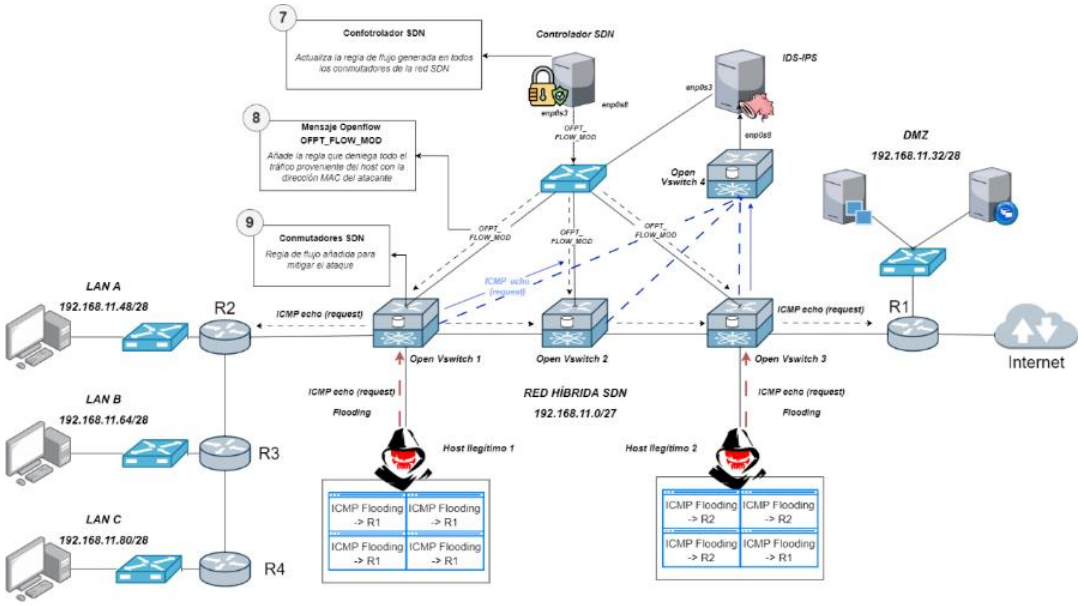


Establecida y enviada la alerta al controlador por parte del IDS, el siguiente proceso que realiza este elemento central de la SDN se representa en la Figura 54. El controlador SDN, a través del mensaje OpenFlow OFPT_FLOW_MOD, distribuye la nueva regla de flujo que deniega todo el tráfico proveniente de la dirección MAC del atacante por parte de los conmutadores de la red.

Al recibir el mensaje OpenFlow con las acciones de mitigación, los switches actualizan su lista de reglas de flujo, añadiendo la regla de mitigación con un nivel de prioridad establecido en 150, debido a que el nuevo flujo tiene como prioridad establecer parámetros de seguridad para resguardar el funcionamiento de la SDN. Es importante destacar que el controlador realiza esta actualización en todos los conmutadores de la red, asegurando que todo el tráfico generado por el atacante sea rechazado, sin importar desde qué punto de conexión, por ende, esta acción garantiza una protección integral y consistente en toda la red.

Figura 54

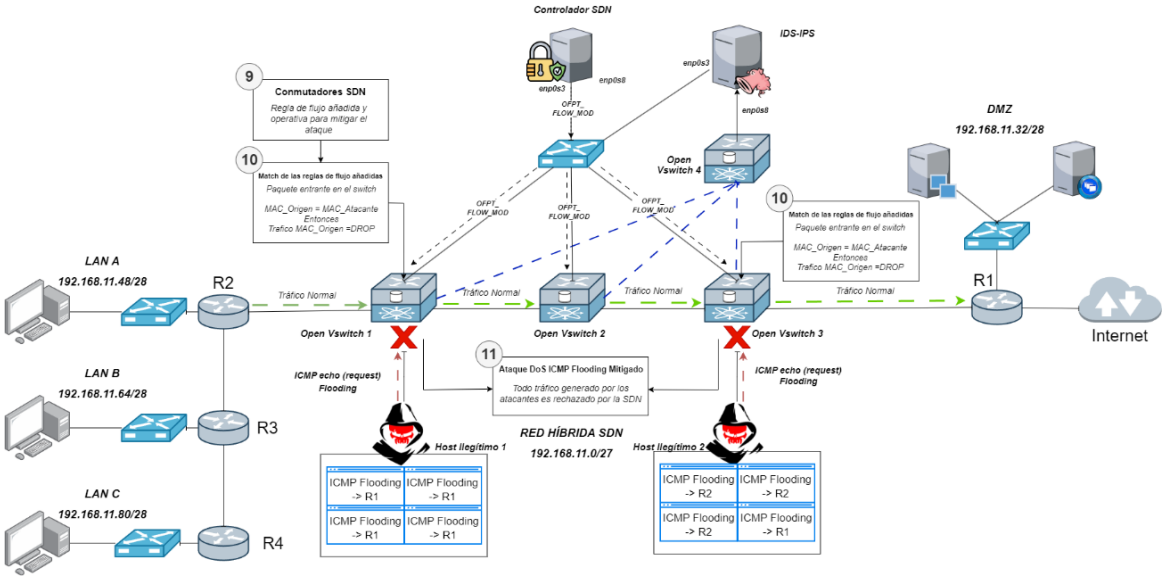
Fase de actualización de reglas de flujo para la mitigación



Finalmente, el tráfico malicioso generado por los atacantes es denegado en toda la red SDN, mientras que el tráfico normal continúa con el proceso de conmutación de paquetes legítimos sin interrupciones. Este proceso asegura una protección efectiva contra los ataques DoS de tipo inundación, manteniendo la operatividad de la red, tal como se muestra en la Figura 55.

Figura 55

Fase de mitigación de la SDN



CAPÍTULO IV: Evaluación de resultados

El presente capítulo presenta los resultados obtenidos de la implementación de los escenarios que conforman el testbed, de modo que las pruebas de funcionamiento abarcan el análisis de las reglas de flujo en cada uno de los entornos durante las fases de ataque y mitigación. Además, se emplean las recomendaciones de la ITU-T Y.1540 para la medición de parámetros de evaluación de la red, con el fin de identificar cómo los ataques y la implementación del mecanismo de seguridad afectan al rendimiento y la eficiencia de la red híbrida SDN.

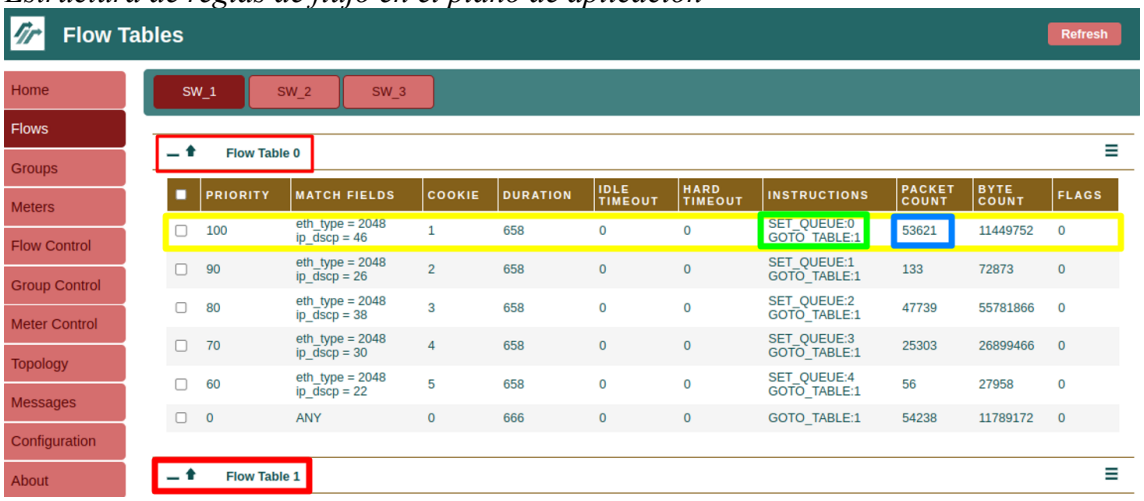
4.1 Red híbrida SDN/MPLS

Las pruebas de funcionamiento en el escenario general de la red SDN/MPLS corresponden en identificar las reglas de flujo de la calidad de servicio aplicada a la red y a la vez la implementación de las funciones MPLS dentro de los conmutadores. Para conseguir el funcionamiento de estas pruebas es necesario utilizar el plano de aplicación de la red SDN mediante la dirección `http://192.168.11.10:8080/home/flows.html`.

La Figura 56 ilustra el contenido de esta sección del plano de la arquitectura SDN. En el recuadro rojo se pueden identificar las tablas que contienen las reglas de flujo para cada uno de los conmutadores asociados al controlador. El recuadro amarillo corresponde a la estructura de una regla perteneciente a la tabla general, junto con las acciones que realizan, las cuales están destacadas en el recuadro celeste. Además, esta funcionalidad del plano de aplicación permite verificar el correcto funcionamiento de las reglas implementadas mediante un contador, identificado en el recuadro azul, el cual muestra el número de paquetes que coinciden con cada regla.

Figura 56

Estructura de reglas de flujo en el plano de aplicación

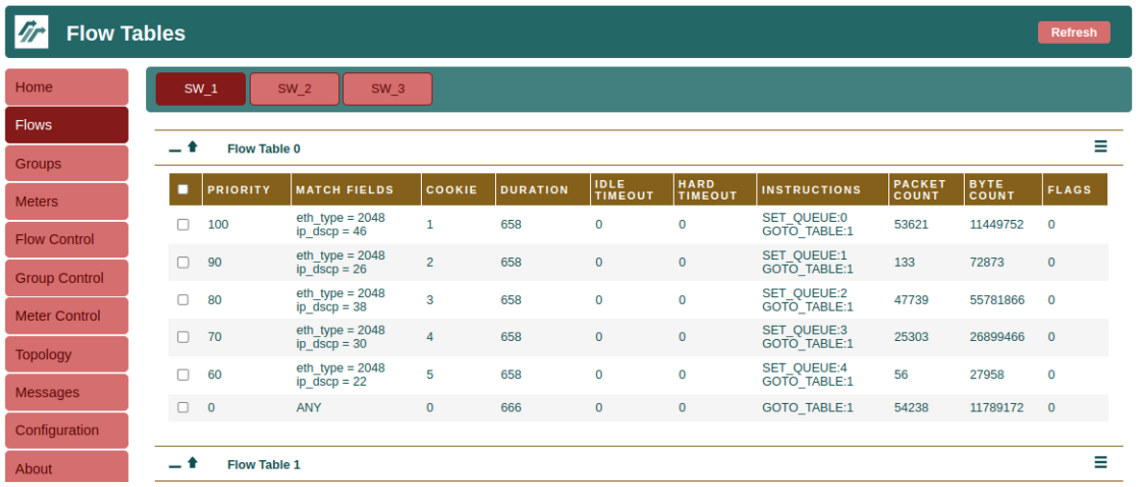


a) Reglas de flujo de calidad de servicio en la red SDN/MPLS

La comprobación de las reglas de flujo se realiza posterior a la implementación de calidad de servicio dentro de la red SDN en la sección 3.2.7. La Figura 57 permite identificar en la Tabla de flujos 0 (Flow Table 0) que cada recuadro que posee representa a una de las reglas implementadas para realizar el encolamiento de los paquetes IP en los conmutadores SDN, corresponden con las siguientes coincidencias:

Figura 57

Reglas de flujo de calidad de servicio

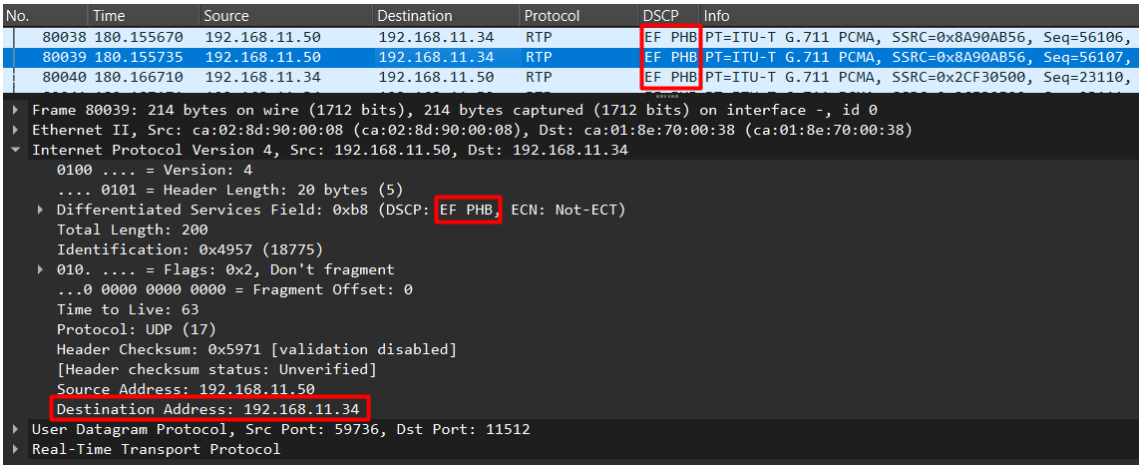


En las siguiente sección se presenta la explicación de cada regla aplicada en la primera tabla de flujos que posee cada conmutador SDN en la red:

- Regla 1: Se aplica a los paquetes IPv4 con el DSCP 46 (EF) y lo asigna a la cola 0, priorizando el tráfico con alta prioridad y baja latencia, lo que influye en la calidad de servicio para este tipo tráfico, y luego redirige estos paquetes a la tabla 1 para un procesamiento adicional correspondiente al proceso de conmutación dentro de la red SDN, no obstante, la información de la tabla 1 es explicada en párrafos posteriores de esta sección. La Figura 58 presenta la información del protocolo SIP que pasa por los conmutadores SDN.

Figura 58

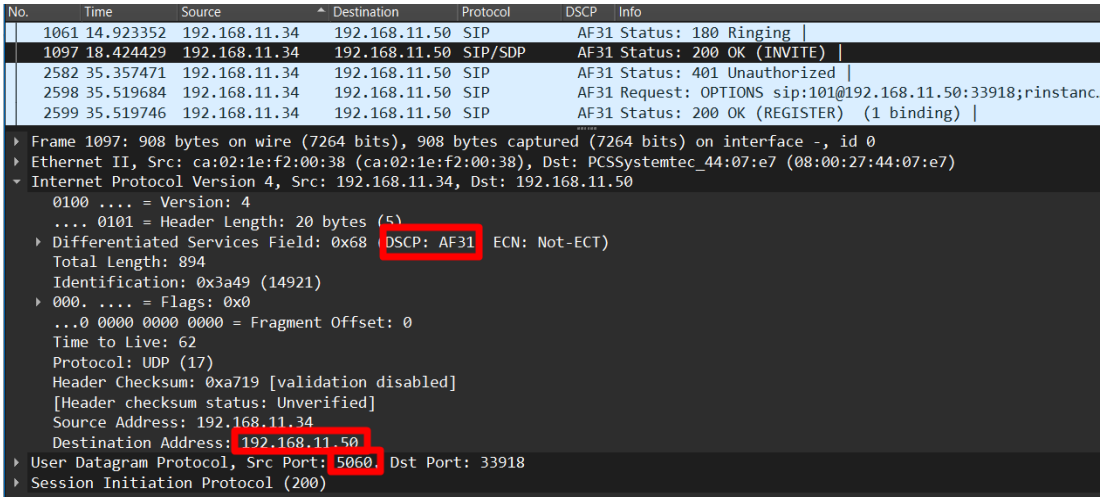
Tráfico RTP en los enlaces de la red SDN



- Regla 2: Se aplica a los paquetes IPv4 que contenga el campo DSCP con el valor de 26 (AF31) y lo asigna a la cola 1, gestionando la señalización de red, lo que influye en la calidad de servicio para este tipo tráfico, para posteriormente redirigir estos paquetes a la tabla 1 para su conmutación . La Figura 59 presenta la información del protocolo SIP que pasa sobre los enlaces de los conmutadores.

Figura 59

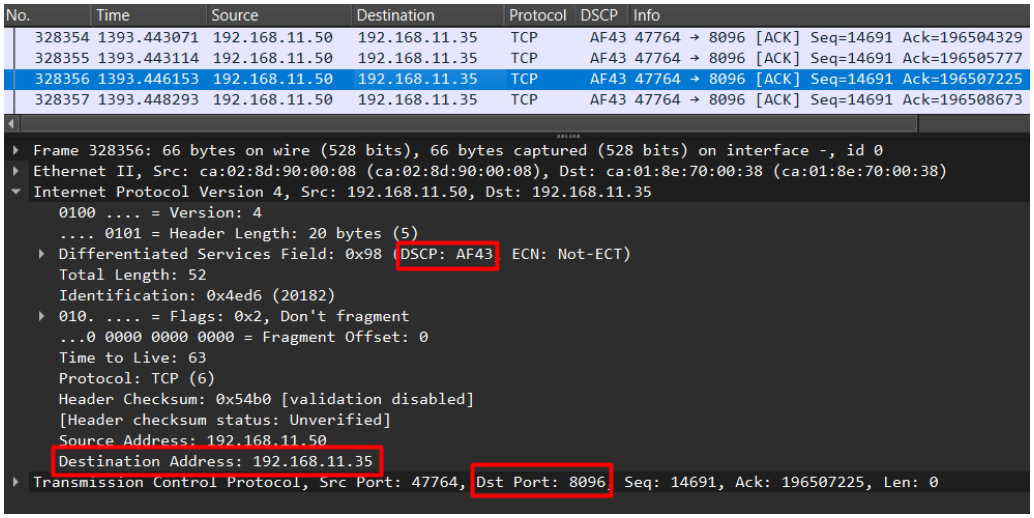
Tráfico SIP en los enlaces de la red SDN



- Regla 3: Se aplica a los paquetes IPv4 que contienen el DSCP 38 correspondiente con AF43 para priorización de QoS y lo asigna a la cola 2 para posteriormenteredirigirlos a la tabla 1 para su conmutación. En la Figura 60 se presenta una captura de paquete mediante Wireshark, en donde se evidencia que el tráfico TCP con puertos de destino 8096 contiene el DSCP (AF43) especificado en las reglas de flujo del conmutador SDN.

Figura 60

Tráfico TCP de VoD en los enlaces de la red SDN

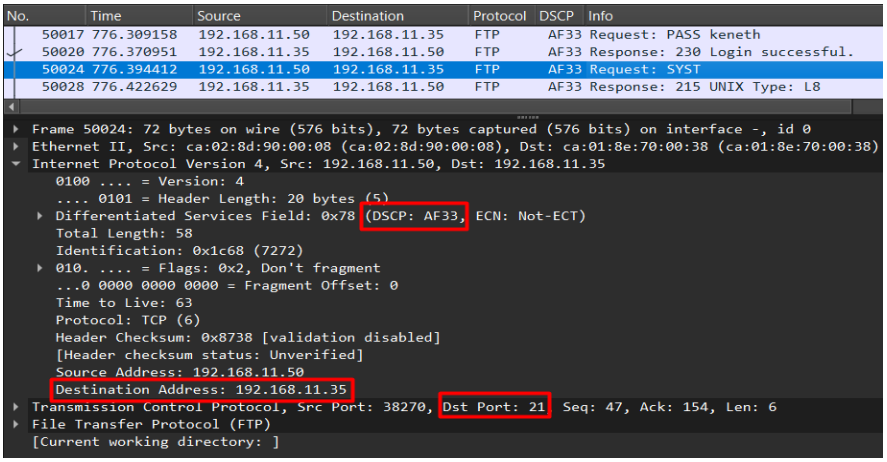


- Regla 4: Regla aplicada a los paquetes IPv4 que posee el campo DSCP con el valor de 30 correspondiente a AF33 y asignarlos a la cola 3, para luego redirigirlos

de igual manera a la tabla 1 para establecer su conmutación hacia el destino. En la Figura 61 se presenta una captura del tráfico FTP, en donde se identifica que todo el tráfico contiene el marcaje AF33 especificado en las reglas de flujo del conmutador SDN.

Figura 61

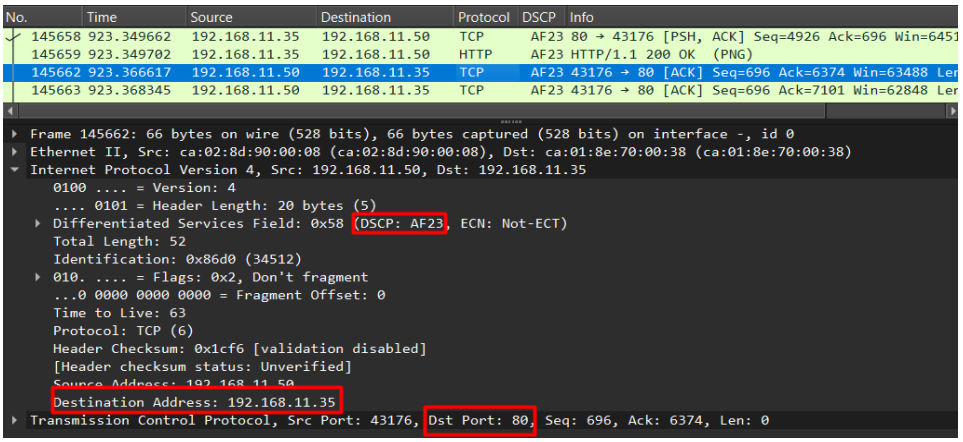
Marcaje del protocolo TCP para FTP en la red SDN



- Regla 5: Esta regla se aplica a los paquetes IPv4 que contienen a campo DSCP a 22 que corresponden con AF23 y asignarlos a la cola 4 para posteriormente redirige el paquete a la tabla 1 para establecer la conmutación del paquete. En la Figura 62 se presenta una captura de paquete mediante Wireshark, en donde se evidencia que el protocolo TCP con puertos de destino 80 contiene el campo DSCP (AF23) .

Figura 62

Marcaje del protocolo TCP para FTP en la red SDN



En lo que respecta al análisis de las tablas de flujo para verificar el funcionamiento de MPLS tras la configuración establecida en la sección 0. Es importante señalar que los Open Vswitch manejan el plano de datos, mientras que el controlador Ryu es responsable de toda la conmutación y distribución de flujos mediante las aplicaciones iniciadas anteriormente.

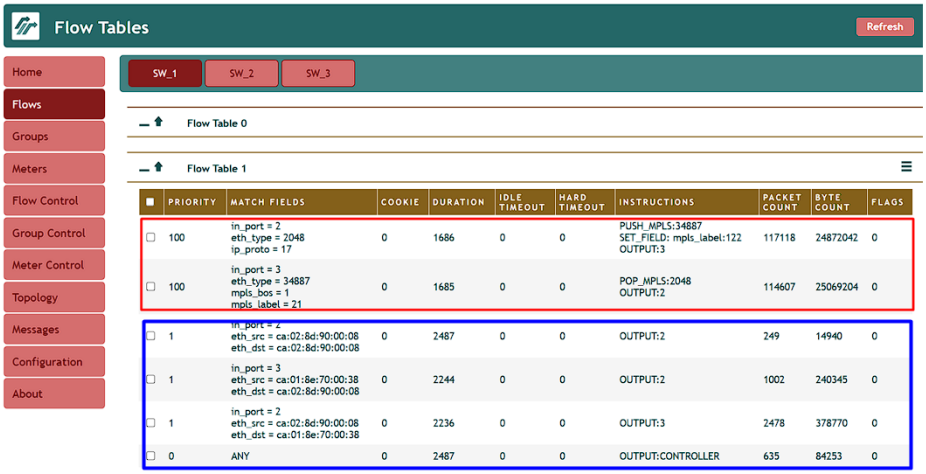
La Figura 63 muestra el contenido de las reglas de flujo de la interfaz del controlador, donde se observan las reglas ubicadas en la Tabla 1 (Flow Table 1) que son identificadas en el recuadro de color rojo y azul. Las reglas identificadas en el recuadro azul gestionan el tráfico basado en direcciones MAC de origen y destino en varios puertos. Estas reglas redirigen paquetes según las coincidencias de direcciones Ethernet, facilitando así la distribución precisa del tráfico a través de los puertos especificados y hacia el controlador cuando no hay coincidencia específica, garantizando una correcta operación y administración del flujo de datos en la red.

Por otro lado, las reglas remarcadas en el cuadro rojo facilitan la implementación de las funciones MPLS en el OpenVswitch1, en este caso el conmutador realizará las funciones de LER dentro de la red SDN/MPLS, en este caso la primera regla se aplica a los paquetes que entran por el puerto 2. Esta regla emplea una acción de "PUSH_MPLS", que agrega una etiqueta MPLS (mpls_label:122) al paquete antes de reenviarlo por el puerto 3.

La segunda regla en cambio afecta a los paquetes que ingresan por el puerto 3 con una etiqueta MPLS específica (mpls_label:21) que es la última en la pila (mpls_bos:1), indicando el final de la pila de etiquetas MPLS. Esta regla realiza una acción de "POP_MPLS", que elimina la etiqueta MPLS del paquete y luego lo envía a través del puerto 2.

Figura 63

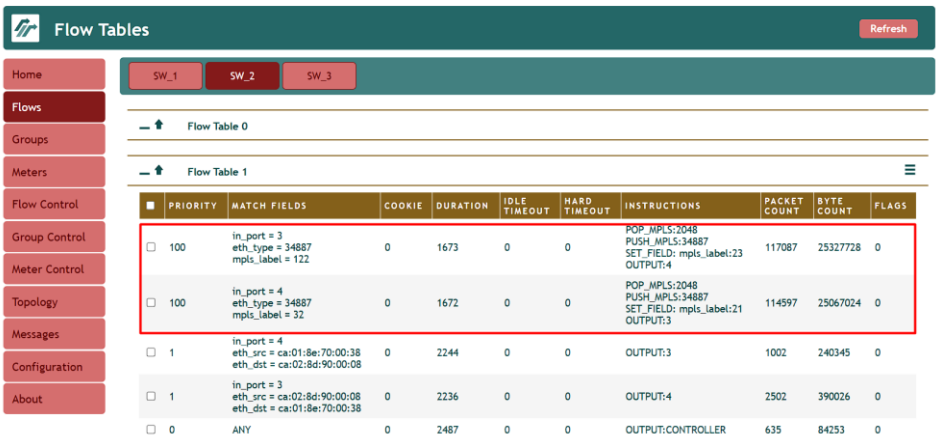
Flujos que realizan las operaciones PUSH y POP en el OpenVswitch1



En la Figura 64 se presentan en el recuadro de color rojo los flujos que permiten realizar la operación SWAP dentro del OpenVswitch2. La primera regla indica que actúa sobre los paquetes que ingresan por el puerto 3 con una etiqueta MPLS de 122. Esta regla remueve la etiqueta MPLS original (POP_MPLS), añade una nueva etiqueta MPLS con el protocolo 34887, cambia la etiqueta MPLS a 23, y reenvía el paquete por el puerto 4. La segunda regla maneja los paquetes que llegan por el puerto 4 con una etiqueta MPLS de 32. Del mismo modo, esta regla remueve la etiqueta MPLS actual, impone una nueva etiqueta con el protocolo 34887, modifica la etiqueta a 21, y dirige el paquete hacia el puerto 3.

Figura 64

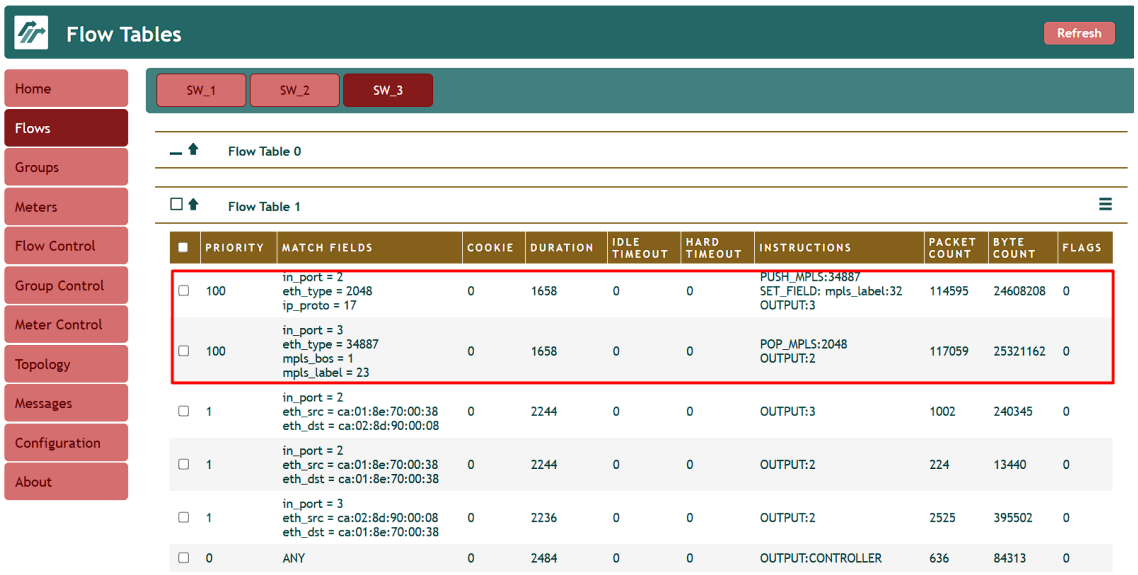
Flujos que realizan las operaciones SWAP en el OpenVswitch1



El análisis de las reglas de flujo implementadas en OpenVswitch3 es identificada en la Figura 65, las reglas identificadas en el cuadro de color rojo identifican las operaciones POP y PUSH que realiza el conmutador ante la llegada de paquetes IPv4 a la red SDN. La primera regla se aplica a los paquetes que entran por el puerto 2 en la cual se emplea una acción de "PUSH_MPLS", que agrega una etiqueta MPLS (mpls_label:32) al paquete antes de reenviarlo por el puerto 3. La segunda regla se emplea a los paquetes que ingresan por el puerto 3 del conmutador con una etiqueta MPLS específica (mpls_label:23) que es la última en la pila (mpls_bos:1), indicando el final de la pila de etiquetas MPLS. Esta regla realiza una acción de "POP_MPLS", que elimina la etiqueta MPLS del paquete y luego lo envía a través del puerto 2.

Figura 65

Flujos que realizan las operaciones PUSH y POP en el OpenVswitch3



PRIORITY	MATCH FIELDS	COOKIE	DURATION	IDLE TIMEOUT	HARD TIMEOUT	INSTRUCTIONS	PACKET COUNT	BYTE COUNT	FLAGS
100	in_port = 2 eth_type = 2048 ip_proto = 17	0	1658	0	0	PUSH_MPLS:34887 SET_FIELD: mpls_label:32 OUTPUT:3	114595	24608208	0
100	in_port = 3 eth_type = 34887 mpls_bos = 1 mpls_label = 23	0	1658	0	0	POP_MPLS:2048 OUTPUT:2	117059	25321162	0
1	in_port = 2 eth_src = ca:01:8e:70:00:38 eth_dst = ca:02:8d:90:00:08	0	2244	0	0	OUTPUT:3	1002	240345	0
1	in_port = 2 eth_src = ca:01:8e:70:00:38 eth_dst = ca:01:8e:70:00:38	0	2244	0	0	OUTPUT:2	224	13440	0
1	in_port = 3 eth_src = ca:02:8d:90:00:08 eth_dst = ca:01:8e:70:00:38	0	2236	0	0	OUTPUT:2	2525	395502	0
0	ANY	0	2484	0	0	OUTPUT:CONTROLLER	636	84313	0

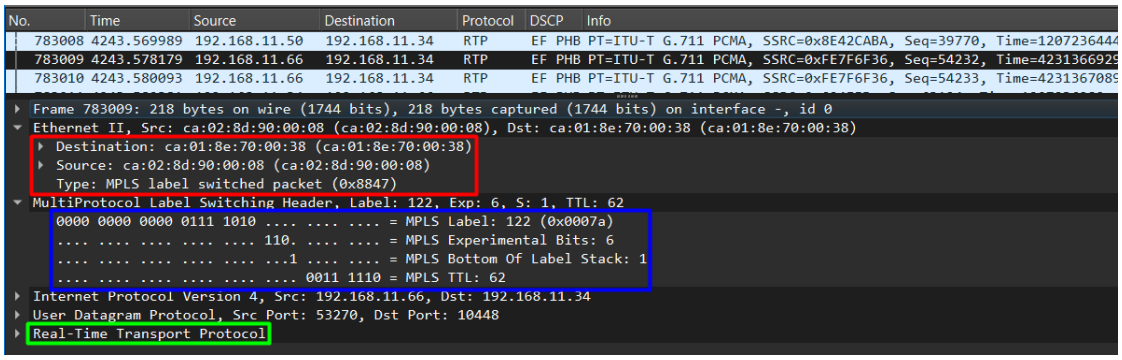
Para verificar el funcionamiento de las tablas de flujo de cada uno de los conmutadores SDN, se utiliza un sniffer como Wireshark para capturar y analizar el tráfico dentro de los enlaces que conforman los conmutadores SDN. La Figura 66 muestra el tráfico RTP que pasa sobre los enlaces de los puertos eth2 de los OpenVswitch 1 y OpenVswitch2 para confirmar que las funciones MPLS y la calidad de servicio está en

operativas dentro de la red SDN. Esta captura específica se realiza durante una llamada telefónica de Cliente1 con la extensión 101 ubicada en la LAN A hacia el Cliente2 con número de extensión 102 localizado en la red LAN B de la topología.

La información presentada en la Figura 66 correspondiente al recuadro rojo describe las direcciones MAC entre lo hosts de origen y destino, a la vez que especifica que existe un etiquetamiento de MPLS aplicado al paquete en el origen. En el recuadro de color azul se identifica que la etiqueta MPLS tiene un valor de 122, un TTL (Time to Live) de 62, y un campo experimental utilizado para indicar que existe calidad de servicio marcado con el valor de 6. Además, se indica que esta etiqueta es la última en la pila de MPLS (BOS, Bottom of Stack, es 1). También se muestra que el paquete es de tipo RTP (Real-Time Transport Protocol), que es comúnmente utilizado para el transporte de audio y video en aplicaciones de comunicación en tiempo real, indicando que el tráfico capturado pertenece a una llamada telefónica VoIP.

Figura 66

Analisis de tráfico RTP dentro de los enlaces de los OpenVswitch 1 y 2

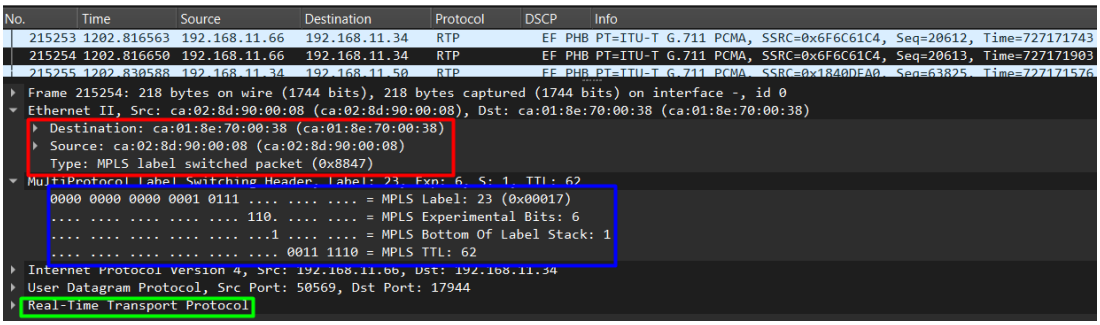


Continuando con el análisis del paquete capturado, este se realiza dentro del enlace los conmutadores OpenVswitch 2 y 3. La información presentada en la Figura 67, correspondiente al recuadro rojo, describe de igual manera que existe un etiquetado MPLS aplicado. Mientras que lo resaltado de color azul identifica que la etiqueta MPLS tiene un valor de 23 debido al intercambio de etiquetas establecidos por las reglas de flujo

del OpenVswitch2, un TTL (Time to Live) de 62, y el campo experimental utilizado para calidad de servicio que se mantiene con el valor 6.

Figura 67

Analisis de tráfico RTP dentro de los enlaces de los OpenVswitch 2 y 3



Finalmente, el análisis de las reglas de flujo en los OpenVswitch y la captura de tráfico en la red SDN demuestran que tanto MPLS como la Calidad de Servicio están funcionando correctamente. Las reglas específicas para operaciones de PUSH, POP y SWAP en los OpenVswitches aseguran una gestión eficiente de las etiquetas MPLS dentro de la red SDN, facilitando así el enrutamiento y la segmentación del tráfico dentro de la red.

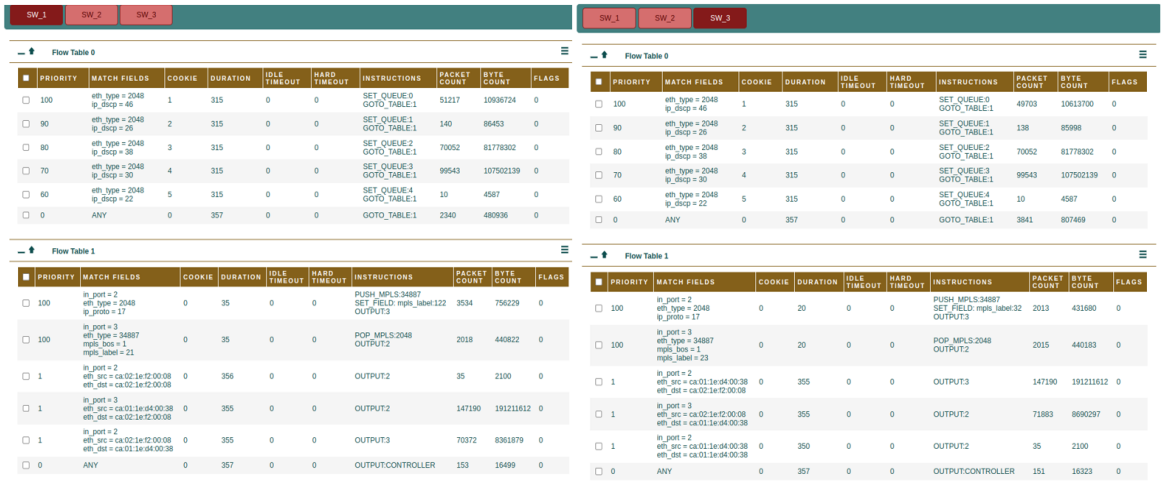
Además, el análisis del tráfico RTP a través de las capturas mediante Wireshark en los enlaces de los OpenVswitch 1, 2 y 3 revela que las etiquetas MPLS están siendo aplicadas y gestionadas adecuadamente, manteniendo el TTL y los campos experimentales para QoS, lo que es crucial para el manejo del tráfico de las aplicaciones utilizados por los hosts clientes dentro de la red SDN/MPLS.

4.2 Mitigación de ataques DoS

En esta sección, se establece inicialmente un entorno en el que la red híbrida SDN opera en condiciones normales, permitiendo un análisis previo de su funcionamiento sin interferencias de seguridad de por medio. Posteriormente, la infraestructura de la red definida por software es sometida a ataques de denegación de servicio para verificar el comportamiento y la eficacia del mecanismo de seguridad implementado ante los ataques que enfrenta la red.

En condiciones normales, la red opera de manera estable en términos de calidad de servicio, etiquetado y encaminamiento a través de MPLS, así como en la conmutación precisa de paquetes entre los conmutadores. Las tablas de flujo, mostradas en la Figura 68, corroboran este comportamiento, evidenciando que cada conmutador aplica políticas de priorización y gestión de tráfico de forma efectiva, lo que conlleva a una administración fluida del flujo de datos y asegura el cumplimiento de los requisitos de rendimiento establecidos para la red.

Figura 68
Reglas de flujo de los conmutadores SDN

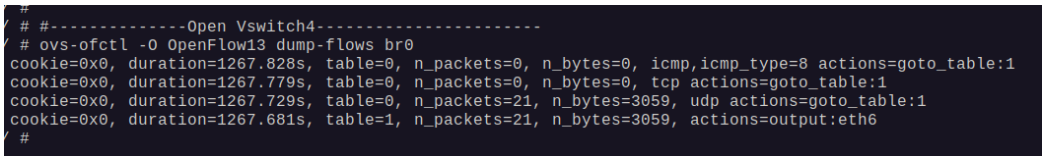


Para garantizar el funcionamiento del mecanismo de seguridad en la red con el fin de evitar bucles debido a la interconexión que poseen los conmutadores, se establecen

reglas de flujo en el conmutador Open vSwitch 4, que tienen como función a redirigir todo el tráfico ICMP, TCP y UDP generado por los ataques hacia el puerto conectado al sistema IDS de Snort. Estas reglas identificadas en la Figura 69 tienen como función que cualquier tipo de tráfico que el controlador identifique como sospechoso se enviado al IDS con el objetivo de inspeccionar el tráfico en busca de actividades sospechosas o patrones de ataque.

Figura 69

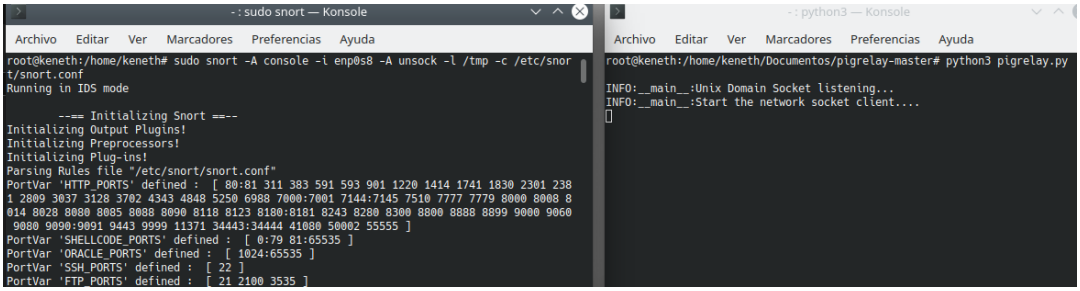
Reglas de flujo de redireccionamiento de tráfico hacia el IDS



Por otra parte, la operación del IDS se basa en una integración conjunta de las capacidades de Snort con scripts de programación de envió de alertas hacia el controlador SDN. El sistema IDS utilizado por Snort mediante sus reglas predefinidas, analiza el tráfico en tiempo real y detecta actividades sospechosas o amenazas potenciales, generando alertas cuando se identifican patrones anómalos; de este modo da paso a las funciones de los scripts que recopilan dicha la información obtenida para posterior enviarla al controlador SDN, el cual puede implementar medidas correctivas, como el bloqueo del tráfico malicioso de los atacantes. La Figura 70 ilustra esta operación integrada, que permite analizar y mitigar ataques DoS en la red SDN, manteniendo un estado de monitoreo activo para prevenir intrusiones en la infraestructura.

Figura 70

Funcionamiento del sistema IDS en la red SDN

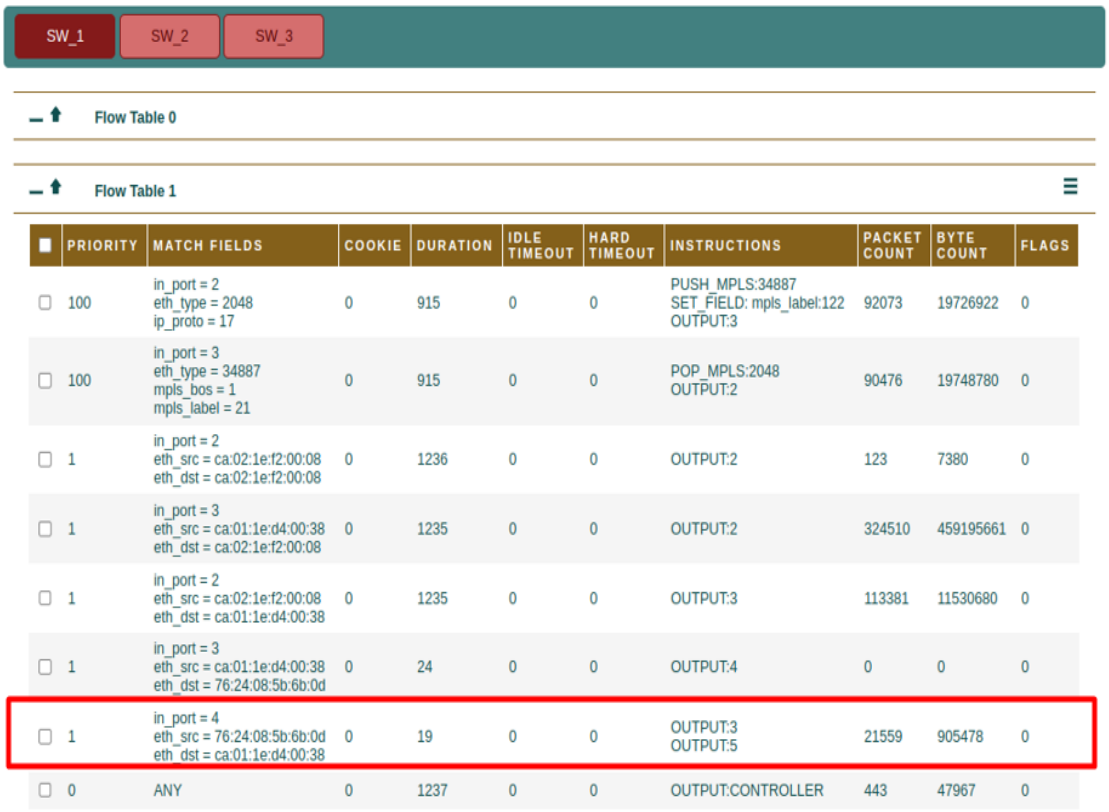


En un entorno donde ocurre un ataque de denegación de servicio contra la infraestructura SDN, como una inundación ICMP, el controlador SDN identifica la llegada masiva de este tipo de paquetes en uno de los puertos de los conmutadores SDN en intervalos cortos. Al detectar este comportamiento anómalo, el controlador redirige el tráfico sospechoso hacia el conmutador OpenVSwitch4, con el objetivo de canalizar este flujo hacia el sistema IDS para su posterior análisis.

La Figura 71 muestra el resultado de esta operación. En el recuadro rojo se destaca la regla de flujo implementada por el controlador en el conmutador OpenVSwitch1, la cual indica que el tráfico originado desde el host conectado al puerto 4 del conmutador sea encaminado a su destino (puerto 3) y, al mismo tiempo, sea duplicado y enviado al puerto 5 que conecta con el sistema del IDS. De esta forma, el tráfico sospechoso es analizado posteriormente por Snort para identificar posibles amenazas.

Figura 71

Regla de redirección de tráfico sospechoso al sistema IDS

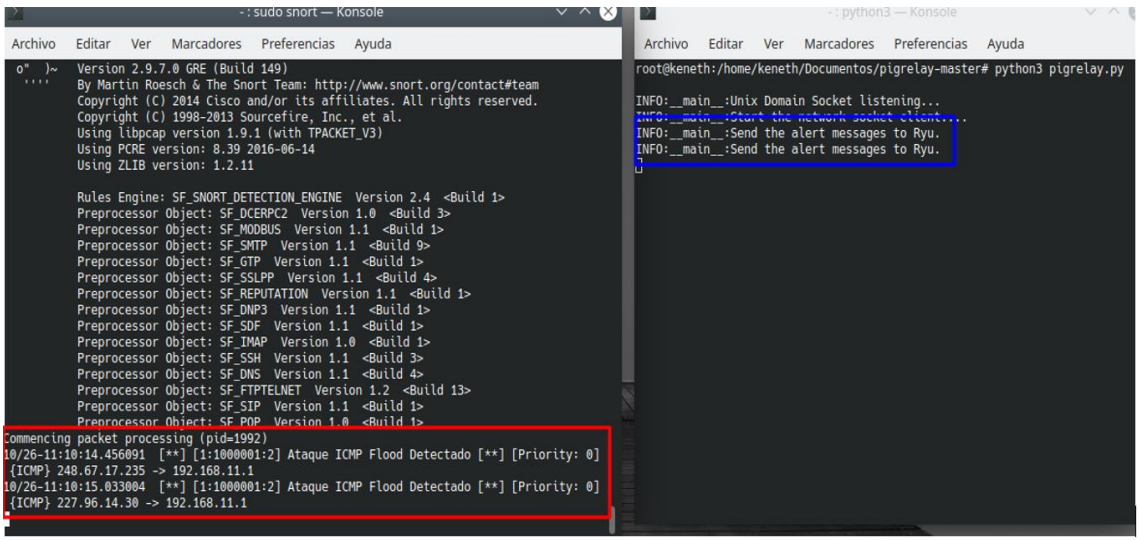


PRIORITY	MATCH FIELDS	COOKIE	DURATION	IDLE TIMEOUT	HARD TIMEOUT	INSTRUCTIONS	PACKET COUNT	BYTE COUNT	FLAGS
100	in_port = 2 eth_type = 2048 ip_proto = 17	0	915	0	0	PUSH_MPLS:34887 SET_FIELD: mpls_label:122 OUTPUT:3	92073	19726922	0
100	in_port = 3 eth_type = 34887 mpls_bos = 1 mpls_label = 21	0	915	0	0	POP_MPLS:2048 OUTPUT:2	90476	19748780	0
1	in_port = 2 eth_src = ca:02:1e:f2:00:08 eth_dst = ca:02:1e:f2:00:08	0	1236	0	0	OUTPUT:2	123	7380	0
1	in_port = 3 eth_src = ca:01:1e:d4:00:38 eth_dst = ca:02:1e:f2:00:08	0	1235	0	0	OUTPUT:2	324510	459195661	0
1	in_port = 2 eth_src = ca:02:1e:f2:00:08 eth_dst = ca:01:1e:d4:00:38	0	1235	0	0	OUTPUT:3	113381	11530680	0
1	in_port = 3 eth_src = ca:01:1e:d4:00:38 eth_dst = 76:24:08:5b:6b:0d	0	24	0	0	OUTPUT:4	0	0	0
1	in_port = 4 eth_src = 76:24:08:5b:6b:0d eth_dst = ca:01:1e:d4:00:38	0	19	0	0	OUTPUT:3 OUTPUT:5	21559	905478	0
0	ANY	0	1237	0	0	OUTPUT:CONTROLLER	443	47967	0

Una vez que el tráfico sospechoso es redirigido hacia el sistema IDS, los resultados del análisis confirman la efectividad de esta estrategia de monitorización, como se evidencia en la Figura #, la cual presenta en el recuadro rojo que en la consola de Snort se detecta el ataque de inundación ICMP, registrando alertas con la etiqueta "Ataque ICMP Flood Detectado". Además, el sistema en la alerta generada recopila la información de los atacantes relacionados con la IP que poseen, dirección MAC y el objetivo del ataque, dando paso a que dicha información sea enviada al controlador SDN para la implementación de la mitigación correspondiente, evidenciado en el recuadro azul de la misma imagen.

Figura 72

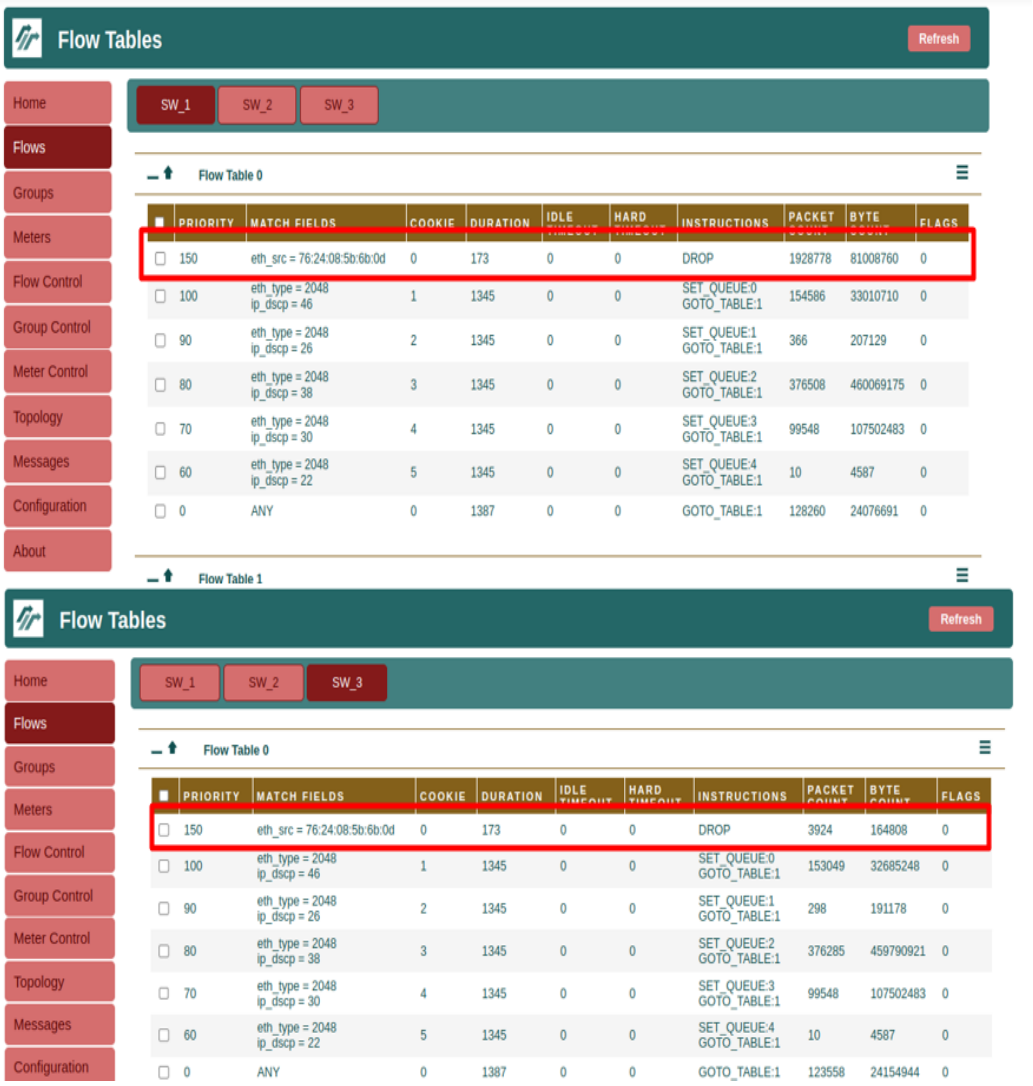
Generación y envió de alertas Snort ante ataque DoS



Finalmente, el mecanismo de mitigación se refleja claramente en la configuración de las tablas de flujo del conmutador SDN. Como se puede apreciar en los recuadros rojos de la **Figura *73***, se han establecido reglas de flujo con un nivel máximo de prioridad de150 en la tabla 0, que específicamente identifica y bloquea todo el tráfico proveniente de la dirección MAC del atacante 76:24:08:5b:6b:0d mediante la acción "DROP". La efectividad de esta medida se evidencia en los contadores de cada regla en los switches, que muestran el número de paquetes que han sido bloqueados desde su implementación, demostrando así que el sistema ha logrado contener eficazmente el ataque sin afectar al resto del tráfico legítimo en la red SDN.

Figura 73

Reglas de flujo de mitigación del ataque en los conmutadores SDN



4.3 Recomendación ITU-T Y.1540

La recomendación ITU-T Y.1540, denominada “Servicio de comunicación de datos con protocolo Internet – Parámetros de calidad de funcionamiento relativos a la disponibilidad y transferencia de paquetes del protocolo Internet”, tiene como finalidad definir un conjunto de parámetros que son utilizados para evaluar la calidad de funcionamiento de los servicios de comunicación de datos basados en el protocolo IP (ITU, 2022).

Los aspectos fundamentales de esta recomendación se centran en la definición de seis parámetros clave para medir la disponibilidad y la eficiencia en la transferencia de paquetes, lo que permite establecer un marco estandarizado para evaluar la calidad de funcionamiento dentro de una red de datos (ITU, 2022). Estos parámetros proporcionan un enfoque integral para garantizar que los servicios IP cumplan con los niveles de calidad esperados en términos de velocidad, exactitud y disponibilidad.

En la presente investigación, se han seleccionado cuatro de estos parámetros que son: la tasa de errores en los paquetes (IPER), la tasa de pérdida de paquetes (IPLR), la variación del retardo de paquetes (IPDV), y la tasa de duplicación de paquetes (IPDR). La elección de estos cinco parámetros se justifica por las siguientes razones :

1. **Tasa de Errores en los Paquetes (IPER)**: Evaluar la tasa de errores es esencial para determinar la integridad de los datos transmitidos, ya que los errores en los paquetes pueden causar retransmisiones, aumentando el tráfico de la red y reduciendo la eficiencia global.
2. **Tasa de Pérdida de Paquetes (IPLR)**: La pérdida de paquetes es un indicador crítico de la calidad de la red, debido a que este parámetro mide la fiabilidad de la transferencia de datos, siendo especialmente importante en aplicaciones de transmisión en vivo donde la pérdida de información puede resultar en una experiencia deficiente.
3. **Variación del Retardo de Paquetes (IPDV)**: Conocido también como jitter, este parámetro mide la variabilidad en el tiempo de llegada de los paquetes, lo que conlleva a que una alta variación en el retardo puede afectar negativamente la sincronización en las aplicaciones, resultando en cortes y deterioro de la calidad.
4. **Tasa de Duplicación de Paquetes (IPDR)**: La duplicación de paquetes puede aumentar innecesariamente la carga de la red y complicar el procesamiento de

datos en el receptor, por ende, la medición de este parámetro es crucial para asegurar que la red no esté generando redundancia que afecte su eficiencia.

Estos parámetros han sido seleccionados debido a su relevancia directa en la evaluación del tráfico que utiliza la infraestructura de redes híbridas SDN, donde la gestión eficiente del tráfico y la minimización de latencia y pérdida de datos son aspectos fundamentales. Al centrarse en estos cinco parámetros, la investigación busca proporcionar un análisis detallado de las condiciones operativas de la red en los dos que escenarios que posee el testbed.

Además, la recomendación describe el proceso de pruebas que deben seguirse para determinar si un servicio IP dentro de la red de datos está en estado de disponibilidad o de indisponibilidad de acuerdo con los parámetros seleccionados anteriormente, de modo los pasos que permiten el despliegue de las pruebas son los siguientes:

- **Paso1: Determinar el origen (SRC) y el destino (DST) de los paquetes**

Esta fase se centra en identificar los dispositivos críticos de la red en donde se realizarán las pruebas, en este caso los SRC y DST serán definidos en base a la topología general de la red híbrida SDN identificada en la Figura 11.

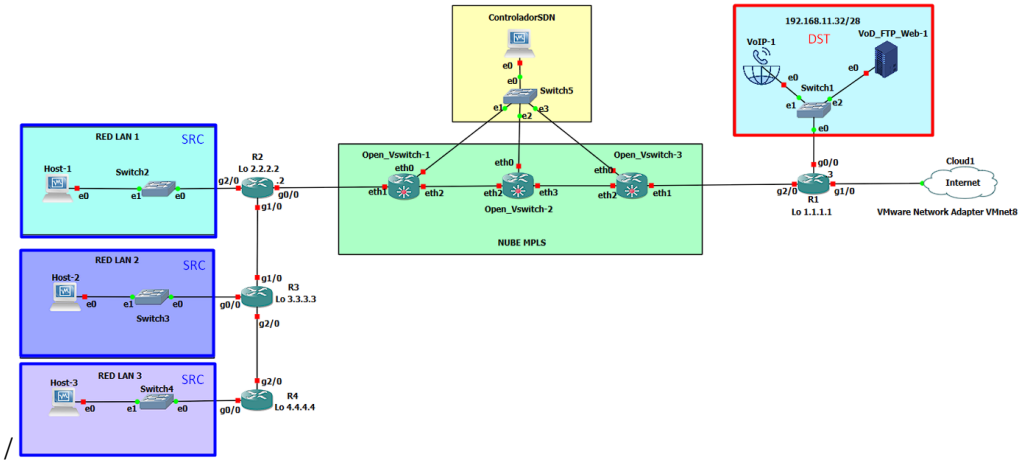
La selección de los dispositivos con el rol de SRC y DST para la evaluación del rendimiento de la red es asignado a los hosts que conforman cada una de las tres redes LAN y los servidores que conforman la red, los cuales son identificados con el recuadro de color azul y rojo respectivamente en la Figura 74. La asignación del rol de SRC a los hosts se basa en el criterio de que estos dispositivos generan diversos tipos de tráfico de diferentes aplicaciones y patrones de uso que se encuentran en un entorno de producción real. Al ser designados con este rol, se evalúa cómo la red maneja diferentes cargas y

tipos de tráfico, proporcionando una visión integral de su desempeño para el registro de los resultados de las pruebas.

Por otro lado, los servidores con el rol de DST son seleccionados por su capacidad para recibir, procesar y responder al tráfico generado por los SRC. Esto permite que el tráfico recibido sea procesado y enviado a su origen, permitiendo así identificar el comportamiento de la red en términos de latencia, capacidad de procesamiento y eficiencia en la retransmisión de datos. Además, la asignación de este rol a los servidores se basa en criterios como su ubicación estratégica dentro del extremo apuesto a los SRC en la red, su capacidad de manejo de grandes volúmenes de tráfico y su compatibilidad con los protocolos de prueba.

Figura 74

Definición de SRCs y DST en la topología de red



- **Paso2: Posicionar los aparatos de prueba o activar los guiones de prueba en los puntos de medición apropiados**

  Esta sección se centra en establecer las herramientas de medición en base a los parámetros seleccionados para medir el rendimiento de la red, a la vez de presentar las secciones de la topología en base a la terminología definida por la ITU-T Y.1540 para los puntos de medición.

La Tabla 42 presenta cómo cada parámetro seleccionado para medir el rendimiento en cada uno de los escenarios está relacionado con una herramienta opensource de medición, las cuales serán utilizadas en las fases de desarrollo de las pruebas. Además, la tabla proporciona una descripción de la funcionalidad de cada de cada herramienta que será utilizada para establecer los resultados requeridos de acuerdo con cada parámetro seleccionado.

Tabla 42
Herramientas de medición para cada parámetro de estimación de rendimiento de la red

Parámetro	Herramienta	Función
IPER	Wireshark	Identifica y analiza paquetes erróneos en la red.
IPLR	Wireshark	Monitorea y conteo la pérdida de paquetes a lo largo del tiempo.
IPDV	Wireshark	Analiza la variación del retardo entre paquetes en las capturas de tráfico.
IPDR	Wireshark	Analiza las capturas de tráfico TCP para identificar duplicaciones de paquetes.

En lo que respecta en el establecimiento de puntos de medición dentro de la red, la recomendación indica en establecer un diagrama de eventos de referencia en base a los términos que corresponde a cada una de las secciones de red que interviene en la transferencia de paquetes IP. Para obtener el diagrama de referencia es necesario establecer los conceptos que la recomendación presenta para definir una terminología identificativa a cada sección y elementos que conforma la red, para conseguir esto la Tabla 43 presenta la información sobre el nombre, abreviaturas y definiciones utilizadas para establecer los componentes del diagrama.

Tabla 43
Terminología para establecer el diagrama de referencia de transferencia de paquetes

Nombre	Abreviatura	Definición
Punto de medición	MP	Ubicaciones específicas dentro de la red donde se pueden observar y registrar los eventos de referencia de transferencia de paquetes IP.

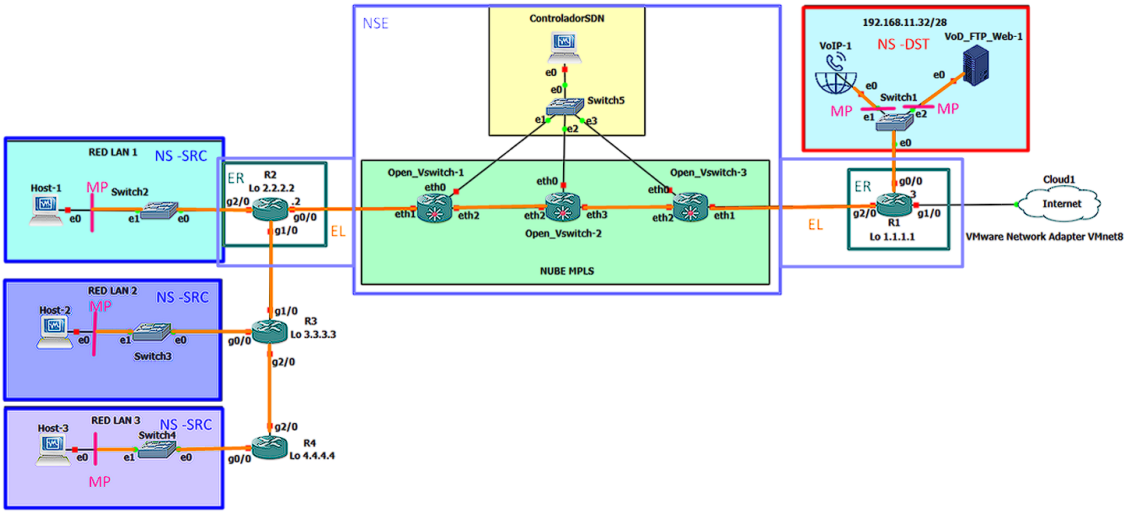
Sección de red	NS	Un conjunto de computadores principales unidos por todos sus enlaces de interconexión que juntos proporcionan una parte del servicio IP.
Sección de red de origen	NS-SRC	La sección de red que contiene el computador principal de origen (SRC) que inicia la transferencia de paquetes IP.
Sección de red de destino	NS-DST	La sección de red que contiene el computador principal de destino (DST) que recibe la transferencia de paquetes IP.
Encaminador	ER	Un computador principal que permite la comunicación entre otros computadores principales reenviando los paquetes IP según su dirección IP.
Enlace central	EL	Un enlace que conecta un computador principal de origen o destino a su encaminador adyacente, o un encaminador de una sección de red a otro.
Conjunto de secciones de red	NSE	Un grupo de secciones de red interconectadas que juntas proporcionan el servicio IP entre un SRC y un DST.

Nota. Adaptado de (ITU, 2022).

Una vez se ha definido cada una de las terminologías necesarias para establecer el diagrama, se establece la ilustración presentada en la Figura 75. Las representaciones de los elementos de red en esta figura se realizan utilizando los conceptos presentados en la Tabla 43, asegurando que cada componente de la red y sus interacciones sean claramente representados conforme a las definiciones establecidas por la Recomendación UIT-T Y.1540.

Figura 75

Diagrama de referencia de transferencia de paquetes IP en la red híbrida SDN



- **Paso3: En un momento determinado previamente, iniciar el envío de paquetes IP distribuidos durante el periodo de tiempo**

Esta sección se centra en realizar las mediciones durante un periodo determinado de tiempo, con la finalidad de obtener una cantidad de datos representativos para poder interpretar de mejor forma los resultados obtenidos de cada uno de los parámetros de evaluación. La Tabla 44 tiene como finalidad proporcionar una guía detallada para la evaluación de la calidad de funcionamiento de la red mediante la especificación de los parámetros clave, el tipo de tráfico adecuado para cada parámetro y el tiempo de prueba recomendado. El establecimiento de los tiempos de prueba asegura que las mediciones sean suficientemente largas para capturar variaciones y comportamientos representativos del tráfico real, garantizando así la obtención de resultados fiables.

Tabla 44
Parámetros de evaluación del rendimiento en la red

Parámetro	Tipo de tráfico	Tiempo de prueba
Tasa de Errores en los Paquetes (IPER)	Tráfico UDP	10-20 minutos
Tasa de Pérdida de Paquetes (IPLR)	Tráfico UDP	10-20 minutos
Variación del Retardo de Paquetes (IPDV)	Tráfico UDP	10-20 minutos
Tasa de Duplicación de Paquetes (IPDR)	Tráfico TCP	10-20 minutos

- **Paso4: : Si el número de resultados de paquetes perdidos es superior, el servicio IP está indisponible durante el intervalo de tiempo**

Esta fase determina la disponibilidad del servicio IP durante el intervalo de tiempo evaluado. Si el número de resultados de paquetes perdidos supera un umbral predefinido, se concluye que el servicio IP está indisponible durante ese periodo. El umbral utilizado

en esta investigación se basa en los valores establecidos por la recomendación UIT-T G.1010, que clasifica la calidad de servicio para usuarios finales de servicios multimedia.

La recomendación tiene como finalidad proporcionar una clasificación amplia de intervalos de calidad de servicio, conforme a la satisfacción del usuario final en una red de datos. Los valores definidos consideran una gama de aplicaciones que transfieren información como voz, video, imagen y texto (ITU, 2001) . Basándose en estos valores, se establece la Tabla 45, que presenta los parámetros de evaluación del rendimiento según la recomendación UIT-T Y.1540 para las aplicaciones que se alojan dentro de los escenarios del testbed, tales como VoIP, VoD, navegación web y FTP.

Esto permite especificar valores preferidos y límites aceptables para métricas como el tiempo de transmisión en un sentido, la variación de retrasos y la pérdida de información. Estos parámetros son fundamentales para determinar los límites aceptables en el rendimiento de la red y verificar mediante los resultados obtenidos como los ataques DoS afectan a estos servicios que utilizan la infraestructura SDN, antes y después de la implementación del IDS/IPS como mecanismo de seguridad.

Tabla 45
Parámetros de rendimiento de la red para las aplicaciones de los ecenarios

Parámetro	**Servidor**	**Aplicación**	**Grado de simetría**	**Tiempo de transmisión en un sentido**	**Variación de retaros**	**Pérdida de información**
IPLR/IPDV	VoIP	Voz en conversación	Dos sentidos	Preferido < 150 ms Límite < 400 ms	< 1 ms	Relación de pérdida de paquete < 3%
IPLR/IPDR /	VoD	Video en un sentido	Un sentido	< 10s	-	Relación de pérdida de paquetes <1 %
IPER/IPLR	Web	Navegación en la web -HTML	Principalmente un sentido	Preferido < 2s/página	N.A.	Nula

				Aceptable < 4s/página		
IPER/IPLR	FTP	Transferencia de gran volumen de datos	Principalmente un sentido	Preferido < 2 s Aceptable < 4 s	N.A	Nula

Nota. Adaptado de (ITU, 2001)

- **Paso 5: Si, de acuerdo con los resultados del paso 4, el servicio IP (sección básica o NSE) no se declara indisponible, ello quiere decir que está disponible durante este intervalo de tiempo.**

La última fase confirmar la disponibilidad del servicio IP durante el intervalo de tiempo evaluado. Si los resultados del Paso 4 no indican una cantidad de paquetes perdidos que exceda el umbral predefinido, se concluye que el servicio IP (ya sea una sección básica o el conjunto de secciones de red, NSE) está disponible durante ese periodo.

4.4 Métodos de medición de parámetros de rendimiento

La evaluación de los escenarios del testbed comienza con la selección de un punto de medición basado en el diagrama de referencia de topología de transferencia de paquetes IP definido en la recomendación UIT Y.1540, presentado en la **Figura *75***. El punto elegido corresponde al host más distante de los servicios de red de la topología híbrida SDN, en este caso, el host de la red LAN C cumple con los requerimientos para esta evaluación.

La elección del host más distante como punto de medición se basa en un inicio a la distancia que posee a los servicios de red, debido a que su ubicación implica la presencia de más saltos en el trayecto de los paquetes, lo que incrementa la posibilidad de congestión y retrasos adicionales, lo que permite evaluar la degradación de la calidad de servicio que presenta la red híbrida SDN en condiciones más desfavorables. En base a estos criterios, el enfoque de esta medición corresponde en evaluar el "peor caso",

asegurando que las mediciones reflejen el comportamiento de la red bajo cargas y distancias máximas, donde el rendimiento es más susceptible de degradarse.

Adicionalmente se establece el tiempo de pruebas para los escenarios, los cuales se desarrollarán durante un intervalo total de 20 minutos, con puntos de control de medición a los 10 y 20 minutos, como se muestra en la **Tabla *44***. Este enfoque permite evaluar tanto el comportamiento inicial de la red como su evolución a lo largo del tiempo, identificando posibles fluctuaciones en el rendimiento del servicio bajo distintas condiciones de carga y uso que cada escenario del testbed posee. Además, al realizar las mediciones en dos momentos clave, se asegura la detección de cualquier degradación progresiva o cambios en el rendimiento durante el tiempo de evaluación.

El tiempo de medición se ha limitado a 20 minutos, ya que extenderlo significativamente implicaría capturar y procesar un volumen masivo de paquetes, lo que sobrecargaría la herramienta de captura y análisis de tráfico de red utilizado. Esto se debe a que un mayor volumen de tráfico no solo demandaría más recursos de procesamiento y almacenamiento, sino que también podría comprometer la precisión del análisis al aumentar la complejidad en la extracción de los parámetros relevantes como la latencia, jitter y pérdida de paquetes.

Por lo tanto, en esta sección se describen los procesos utilizados para obtener los resultados de cada parámetro de medición UIT-T Y.1540 detallado en la Tabla 42, utilizando las herramientas de análisis estadístico de tráfico que ofrece el analizador de paquetes Wireshark. A continuación, se explica el método empleado para calcular cada parámetro, basándose en la información proporcionada por el software:

a) Tasa de pérdida de paquetes (IPLR)

Los resultados del parámetro de tasa de pérdida de paquetes corresponden a mediciones realizadas al tráfico generado en la capa de transporte durante la transmisión de datos entre los hosts, de modo que este parámetro es calculado como el porcentaje de paquetes que no logran llegar a su destino en relación con el total de paquetes enviados, según lo definido en la recomendación UIT Y.1540. Para obtener el resultado de estas mediciones en el testbed, se utiliza el analizador de tráfico Wireshark, que permite identificar y registrar los paquetes perdidos mediante la opción de contadores de paquetes que posee.

En aplicaciones de tiempo real que utilizan el protocolo UDP, como el servicio de VoIP presente en los escenarios, la obtención del parámetro IPLR es mediante el análisis de número de paquetes intercambiados ente los clientes y el servidor. Esto se debe a que protocolos de capa aplicación como RTP son los encargados de garantizar el intercambio de los paquetes de audio desde el emisor hasta el receptor en tiempo real, asegurando que la comunicación sea fluida y con la menor latencia posible.

Para calcular el porcentaje de paquetes perdidos (IPLR) en el tráfico de VoIP se presenta la **Figura *76***, la cual ilustra los pasos necesarios para obtener este valor de medición de rendimiento. El proceso comienza con la captura del tráfico de red generado mediane el software de Wireshark, posteriormente es necesario verificar la existencia de paquetes RTP usando un filtro específico en la sección de filtros del software denominado `rtp`, el cual es el encargado de presentar solo el tráfico RTP capturado.

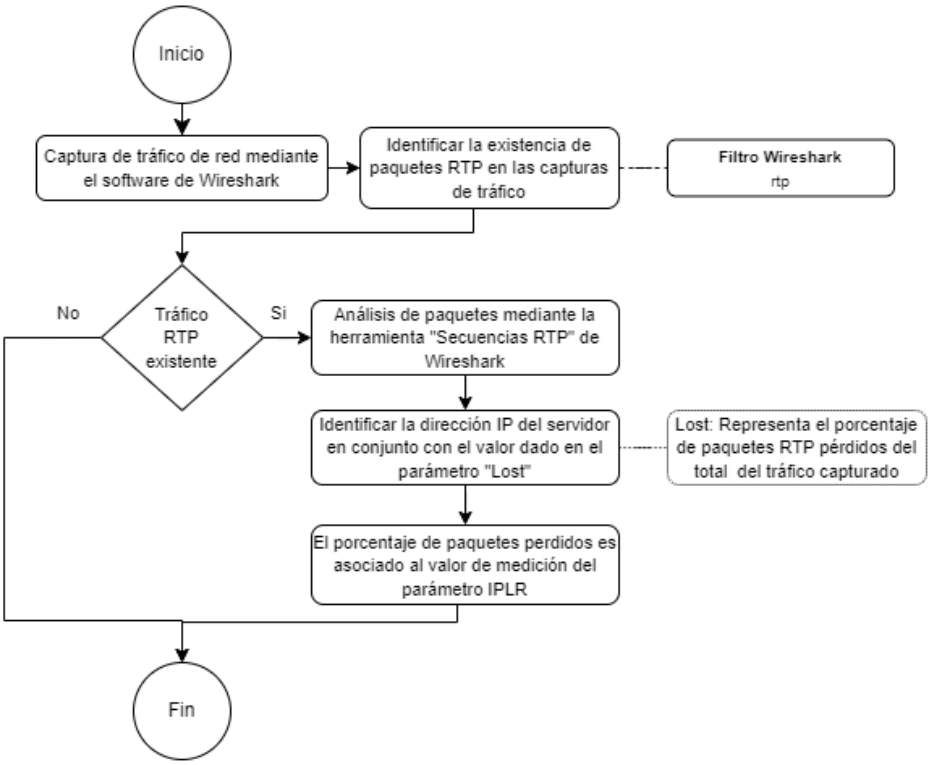
Si este tipo de tráfico es detectado, los paquetes que cumplan con este criterio de selección serán analizados mediante la herramienta “Secuencias RTP” en Wireshark, la cual presenta de forma estadística los valores correspondientes al total de paquetes

transmitidos, paquetes recibidos, porcentaje de paquetes perdidos e información correspondiente al estado de la llamada generada por los hosts.

Una vez que la herramienta muestra los resultados del análisis del tráfico RTP, los valores a tomar en cuenta corresponden a toda la información relacionada con la dirección IP del servidor (VoIP). Esto se debe a que el parámetro IPLR busca reflejar cómo el tráfico enviado por el servidor experimenta pérdidas al atravesar la red antes de llegar al cliente. Por lo tanto, en la herramienta se toma el valor de pérdida de paquetes indicado en la sección "Lost" de la información del servidor, ya que es la encargada de presentar el resultado asociado con el parámetro IPLR correspondiente al porcentaje de paquetes perdidos.

La asociación de IPLR con este resultado se debe a que el software recopila el número de secuencia incluido en el encabezado de cada paquete RTP, un campo de 16 bits que incrementa de forma secuencial con cada paquete enviado, de modo que Wireshark compara estos números de secuencia esperados con los efectivamente capturados para determinar si hay saltos, dando como resultado que se presente el porcentaje de paquetes perdidos del total enviado. Caso contrario, si no se ha detectado tráfico RTP el resultado del parámetro IPLR es considerado como 0%.

Figura 76
Medición de IPLR para tráfico de VoIP



Por otra parte, en lo que respecta a la medición del parámetro IPLR para aplicaciones que utilizan el protocolo TCP el proceso es diferente en comparación a UDP. Esto se debe a que TCP como protocolo de capa de transporte, opera a nivel de segmentos en lugar de paquetes, lo que conlleva a que en esta capa es el encargado de implementar mecanismos de control de flujo, gestión de congestión y retransmisión automática, los cuales garantizan que los segmentos que no lleguen inicialmente a su destino sean reenviados.

Como resultado, la medición del parámetro IPLR para TCP corresponde en identificar los segmentos que no alcanzaron su destino dentro del tiempo esperado y no recibe un acuse de recibo por parte del receptor. Por lo tanto, el resultado de este parámetro se presenta como el porcentaje de segmentos perdidos, reconociendo que esta métrica captura la pérdida a nivel de transporte y no a nivel de paquetes en la red, lo que

conlleva a que este ajuste en la medición asegure que los resultados sean coherentes con las operaciones que realiza el protocolo TCP.

El proceso de medición del IPLR para TCP es ilustrado en la **Figura *77***, la cual indica las acciones y filtros que se requieren aplicar para obtener este valor. La etapa inicial establece la captura del tráfico de red mediante el software Wireshark, para posteriormente aplicar un filtro que permita enfocarse exclusivamente a los paquetes TCP asociados a un número de puerto específico. Esto se debe a que cada puerto o conjunto de puertos están asociados a un servicio en específico (como HTTP, FTP, VoIP), de modo que el filtro configurado en Wireshark con la sintaxis `tcp.port==#` permite aislar el tráfico requerido, donde # es el número del puerto del servicio seleccionado.

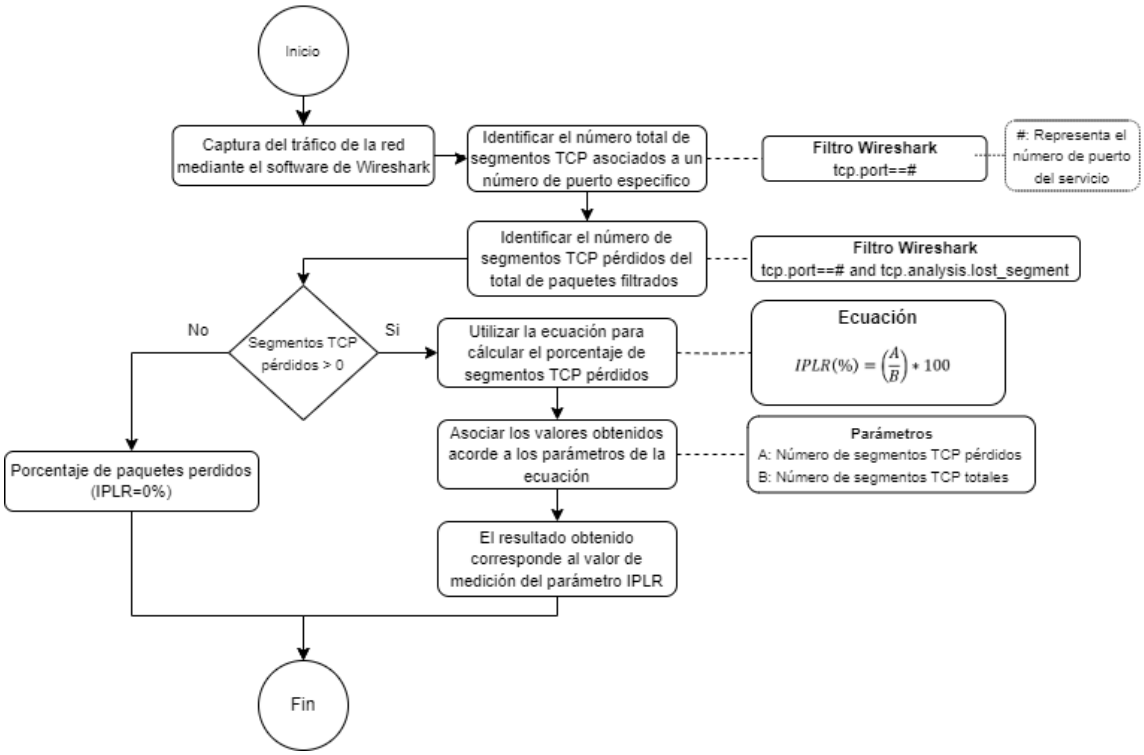
Después de filtrar el tráfico que se requiere analizar, el siguiente proceso consiste en identificar el número de paquetes TCP que se han perdido dentro del tráfico capturado, lo que conlleva en la implementación de un nuevo filtro denominado `tcp.port==# && tcp.analysis.lost_segment`, que tiene como finalidad aislar los segmentos de paquetes que no han llegado a su destino, siendo esta acción esencial para determinar la cantidad de pérdidas que se han realizado en la comunicación TCP.Si el número de paquetes perdidos es mayor a cero, inicia el proceso de cálculo del porcentaje de pérdida de paquetes, lo que resulta en utilizar la (Ec. 1 de razón porcentual. En esta expresión se presentan dos variables, en donde A representa el número de paquetes TCP perdidos y B corresponde al número total de paquetes TCP capturados en el análisis.

$$IPLR = \left(\frac{A}{B}\right) * 100 \qquad \text{(Ec. 1)}$$

Finalmente, se asocia cada valor correspondiente a cada parámetro en la ecuación y el resultado obtenido representa el porcentaje de pérdida de segmentos en la red. No

obstante, si la condición de existencia de segmentos TCP perdidos no se cumple el valor del parámetro IPLR corresponde al 0%, debido a que no se han presentado ninguna pérdida.

Figura 77
Medición de IPLR para tráfico TCP



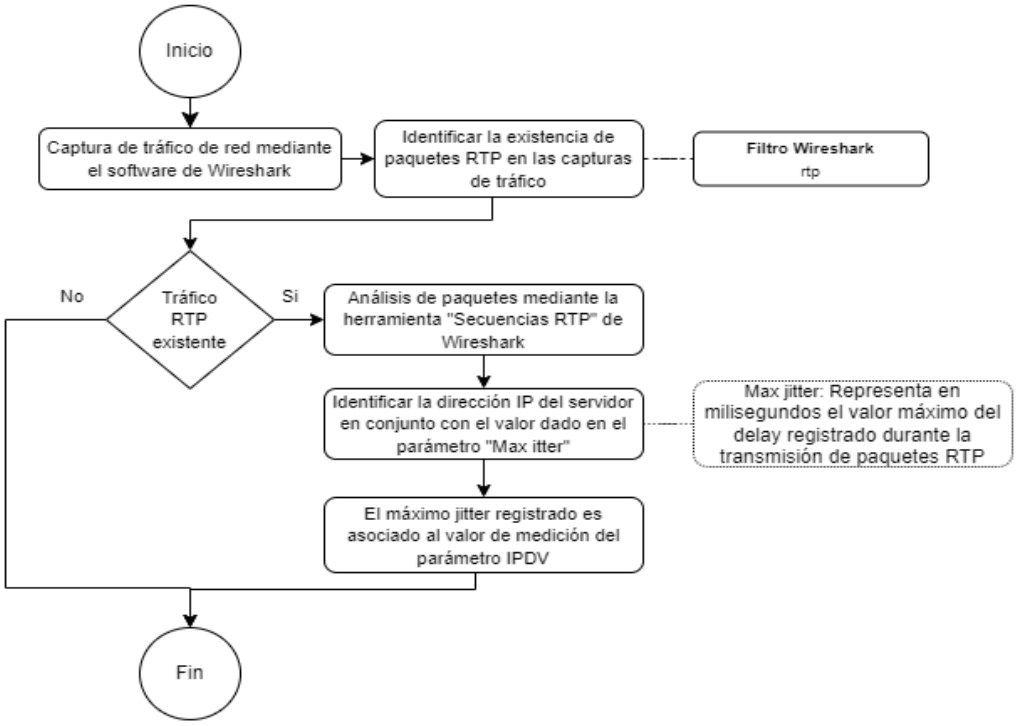
b) Variación del retardo de paquetes (IPDV)

El parámetro de medición de variación del retardo se relaciona de forma directa con la conceptualización del jitter, ya que ambos evalúan las fluctuaciones en el tiempo de entrega de los paquetes a lo largo de una transmisión en la red, de modo que este indicador resulta esencial para determinar la estabilidad del servicio de aplicaciones sensibles al tiempo. En este caso, el IPDV proporciona una medida estadística del rango de variabilidad en el retardo de paquetes dentro de un flujo de datos, y su valor corresponde jitter máximo registrado entre todos los paquetes medidos.

El proceso de medición del parámetro IPDV en aplicaciones de tiempo real, como el servicio de VoIP de los escenarios del testbed, es presentado en la **Figura *78***. La etapa inicial en el proceso de medición consta en identificar en las capturas del tráfico de red la existencia de paquetes RTP mediante filtros de Wireshark, lo que facilita aislar únicamente este tipo de datos relevantes para este análisis. Caso contrario si no se detecta tráfico RTP, el proceso concluye, indicando la ausencia de este tipo de datos en la red.

En caso de que se detecte tráfico RTP, comienza el análisis de los paquetes mediante la herramienta “Secuencias RTP” de Wireshark. Dentro de la información presentada en la herramienta, se identifica la dirección IP del servidor involucrado junto con el valor máximo de jitter registrado, ubicado en el campo "Max jitter", que representa el mayor retraso en la transmisión de los paquetes RTP medido en milisegundos. Por ende, este valor es asociado al parámetro IPDV con la finalidad de proporcionar un indicador sobre la variación del retardo en las transmisiones RTP.

Figura 78
Medición de IPDV para tráfico de VoIP



c) Tasa de Duplicación de Paquetes (IPDR)

El parámetro IPDR está directamente relacionado con el número total de segmentos TCP que han sido duplicados, debido a que este tipo de tráfico es indicativo de problemas en la transmisión de datos, como congestión en la red o errores en los mecanismos de retransmisión. Esto conlleva a que la existencia de duplicación de paquetes genere un impacto negativo en el rendimiento de las aplicaciones, ya que aumenta el tráfico innecesario y la vez provoca retrasos en el procesamiento de datos.

Dado que el IPDR mide la proporción de paquetes duplicados en comparación con el total de paquetes transmitidos, el conteo de número de duplicaciones TCP detectadas junto con el número total de paquetes TCP enviados, permite calcular este parámetro como lo muestran las fases presentes en la **Figura *79***. El proceso inicia con la captura del tráfico de red mediante el software Wireshark, que permite analizar las comunicaciones en la red y registrar los paquetes que se transmiten, lo que conlleva a que vez la captura sea obtenida, se requiera identifica el número total de segmentos TCP asociados a un puerto específico utilizando el filtro `tcp.port==#`.

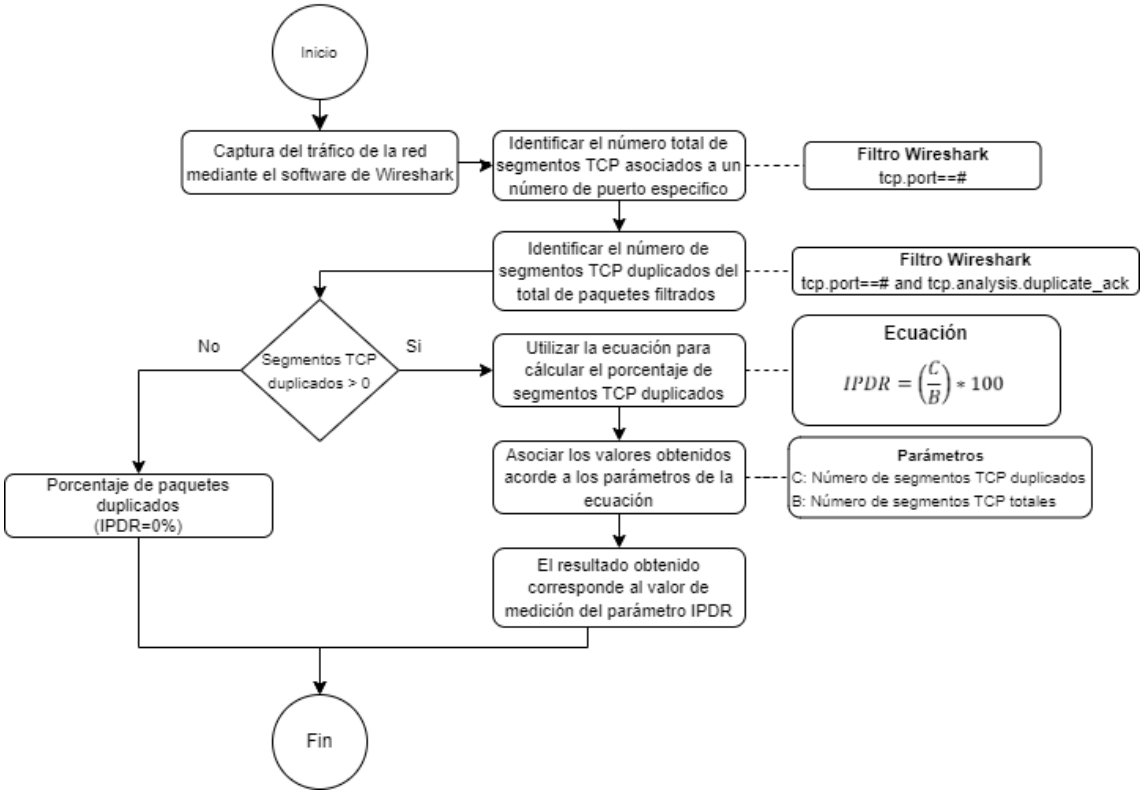
Posteriormente, se identifica el número de segmentos TCP duplicados dentro de los paquetes previamente filtrados, lo que conlleva a la implementación de un filtro adicional como `tcp.port==# and tcp.analysis.duplicate_ack`. Esta combinación de filtros tiene como propósito detectar los ACKs duplicados en las transmisiones, los cuales indican que el remitente no recibió correctamente un segmento previo y está retransmitiendo datos, de modo que su existencia en el tráfico de red es un indicativo de problemas en la comunicación, tales como congestión en la red, pérdidas de paquetes o interrupciones en la estabilidad de la conexión.Si no se detectan segmentos duplicados, el proceso concluye y se determina que el porcentaje de duplicación es igual al 0%. Caso contrario, el cálculo del parámetro IPDR empieza utilizando la ecuación de razón

proporcional (Ec.2) para determinar el porcentaje total de segmentos duplicados del total del tráfico filtrado, donde C corresponde al número de segmentos TCP duplicados y B es el número total de segmentos TCP capturados.

$$IPDR = \left(\frac{C}{B}\right) * 100 \qquad \text{(Ec. 2)}$$

Finalmente, se asocian los valores obtenidos a los parámetros de la ecuación, y el resultado calculado representa el valor de medición del parámetro IPDR, proporcionando un indicador sobre la incidencia de duplicación en la transmisión de los datos.

Figura 79
Medición de IPDR para tráfico TCP



d) Tasa de Errores en los Paquetes (IPER)

La tasa de errores en los paquetes corresponde al cuarto parámetro de medición del rendimiento de la red y está directamente vinculada con los segmentos retransmitidos

en el tráfico TCP, debido a que estos representan una métrica que identifica problemas en la entrega exitosa de los datos durante el intercambio de información entre el cliente y el servidor.

La medición del IPER mediante los segmentos retransmitidos se basa en que cada retransmisión refleja que un segmento enviado no es reconocido por el receptor mediante un acuse de recibo dentro de un tiempo específico o cuando se detectan problemas como alteraciones en los datos, lo cual puede atribuirse a errores en la comunicación, congestión o interferencias en la red. Esto conlleva a que una alta frecuencia de retransmisiones impacte negativamente en el funcionamiento de las aplicaciones, provocando retrasos, pérdida de información e interrupciones en su servicio.

Para ilustrar las fases requeridas para calcular el IPER en aplicaciones que utilizan TCP como protocolo de transporte, se presenta la **Figura *80***, la cual describe de los pasos necesarios para identificar, analizar y calcular este parámetro. La fase inicial de la medición comienza con la captura del tráfico de red mediante un analizador de paquetes como Wireshark, de modo que una vez se ha obtenido el tráfico por analizar, se filtran los segmentos TCP de acuerdo con el número de puerto del servicio, lo que permite identificar el total de paquetes relevantes para el análisis.

Posteriormente, se identifica el número de segmentos retransmitidos mediante el filtro `tcp.port==# and tcp.analysis.retransmission`, lo que resulta en que el software aislé los paquetes que tuvieron que ser reenviados debido a problemas en la transmisión, como pérdida de datos o ausencia de confirmación (ACK) por parte del receptor. Con esta información presentada en el software, se establece una condición en la cual si no existen segmentos retransmitidos el parámetro IPER posee un 0% de

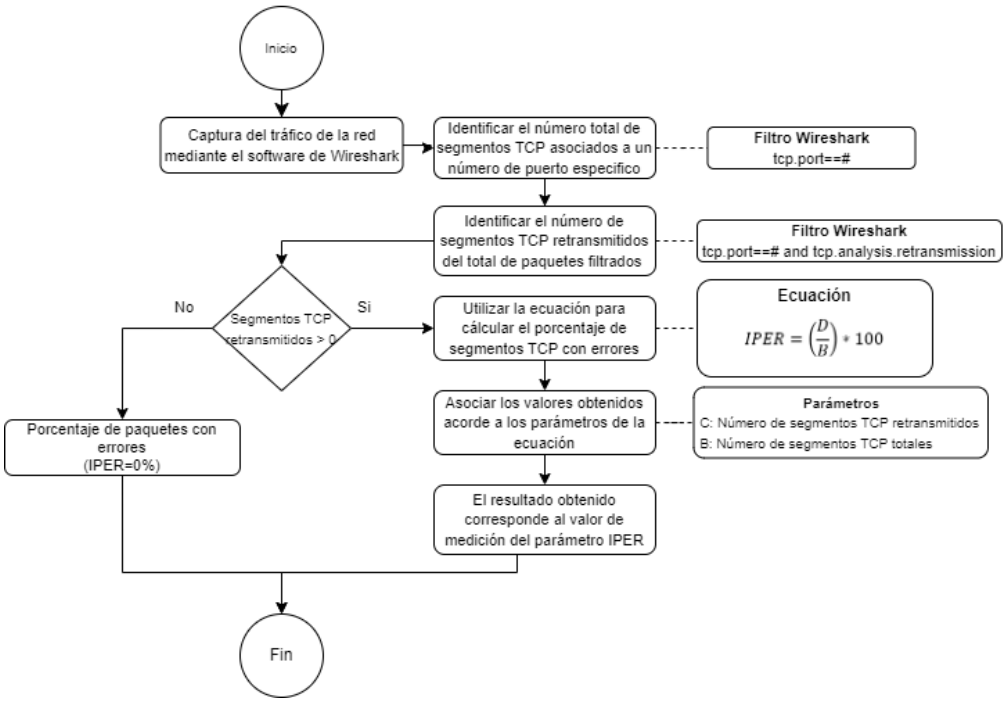
paquetes con errores, caso contrario ante la existencia de retransmisiones inicia el proceso de cálculo.

El cálculo del parámetro es realizado aplicando la ecuación de razón porcentual para errores en los segmentos (Ec.3), en la cual la variable C corresponde al número de segmentos retransmitidos y la variable B indica el número total de segmentos analizados, lo que conlleva a que los valores obtenidos previamente de acuerdo con los filtros expuestos se asignen a sus correspondientes variables dentro de la ecuación, permitiendo obtener el resultado final del cálculo del IPER

$$IPER = \left(\frac{D}{B}\right) * 100 \qquad \text{(Ec. 3)}$$

Finalmente, el resultado del porcentaje obtenido en el cálculo se asocia al parámetro IPER, el cual refleja la proporción de errores en la comunicación TCP evaluada proporcionando este indicador para evaluar la calidad del servicio ofrecido en la red.

Figura 80
Medición de IPER para tráfico TCP



4.5 Evaluación de escenarios

Las mediciones del rendimiento en los escenarios del testbed se realizaron aplicando los métodos descritos en la sección 4.4, las cuales se basan en las recomendaciones de la UIT Y.1540 y el uso de herramientas de análisis como Wireshark, que permitió obtener los resultados de cada parámetro evaluado.

Los parámetros medidos incluyen la tasa de pérdida de paquetes, la variación en el retardo, la duplicación de paquetes y la tasa de errores en los servicios de la red. Esto asegura que los resultados reflejen el comportamiento de la red en tres condiciones: operación ideal, ataques de denegación de servicio y el funcionamiento de la solución de seguridad implementada para mitigar estos ataques.

En las siguientes subsecciones, se presentan a detalle los resultados recopilados del rendimiento de los servicios dentro de la red, destacando cómo cada escenario posee un comportamiento diferente acorde al desempeño que presenta la red.

4.5.1 Evaluación del escenario general

El escenario general corresponde al estado ideal de la red híbrida SDN, donde no existen amenazas de seguridad que afecten a su funcionamiento. Esto conlleva a que las mediciones de cada parámetro de rendimiento en este escenario tengan como propósito establecer los valores de referencia, los cuales representan el comportamiento esperado de los servicios y la red en condiciones normales de operación. Por ende, los resultados recopilados sirven como punto de comparación para evaluar el impacto de situaciones adversas en el escenario posterior (Ataques DoS) y evaluar la eficacia de los mecanismos de seguridad (IDS/IPS) implementados para garantizar el funcionamiento de la red.

La evaluación en el escenario general comienza con el análisis del tráfico de red capturado en el primer intervalo de medición, correspondiente a los primeros 10 minutos,

en los cuales se obtuvieron un total de 784523 paquetes del funcionamiento en paralelo de todos los servicios que la red que posee. A partir de esta captura, se analizaron las diferentes métricas de rendimiento para los servicios evaluados que incluyen a las aplicaciones VoIP, VoD, Web y FTP, los cuales están presentes en la red.

Los resultados del análisis realizado en el intervalo de 10 minutos se resumen en la Tabla 46, donde se presenta una evaluación del rendimiento de diversos servicios de red, en base a los parámetros de IPLR, IPDR, IPER e IPDV. Adicionalmente, la información presente en la tabla también muestra el tipo de tráfico (UDP o TCP) que genera cada servicio, para posteriormente evaluar los parámetros en función del total de paquetes transmitidos y el número de paquetes afectados.

La recopilación de cada uno de los valores obtenidos en este primer intervalo de registro proporciona una visión preliminar del comportamiento de la red bajo condiciones de carga media. Por ejemplo, los resultados muestran que el servicio de VoIP presenta una tasa de pérdida de paquetes del 0.141 % y una variación de retardo promedio de 40.61 ms, lo que refleja un rendimiento aceptable para este tipo de tráfico sensible al retardo. Del mismo modo, el servicio VoD registra un IPLR de 0.706 % y una IPER de 0.534 %, lo que indica una mayor susceptibilidad a pérdidas al ser una aplicación que requiere confirmaciones de entrega de los segmentos debido al uso de TCP en capa transporte.

Por otra parte, los servicios Web y FTP muestran un comportamiento significativamente más estable en este escenario ideal. En el caso del servicio Web, no se reportan pérdidas ni errores con un valor de 0 % respectivamente a cada parámetro de evaluación, lo que refleja un funcionamiento óptimo para un tráfico generalmente menos intensivo. De manera similar, el servicio FTP no registra pérdida ni errores, lo que es

consistente con la baja sensibilidad de este tipo de tráfico a variaciones en las condiciones normales de operación en la red.

Tabla 46
Rendimiento de la red del escenario general en un intervalo de 10 min

Servicio	Trafico	Parámetro	Total de paquetes	Paquetes filtrados	Resultado
VoIP	UDP	IPLR	29639	42	0.141%
		IPDV	29639	-	40.6134 ms
VoD	TCP	IPLR	191562	1353	0.706%
		IPDR	191562	1439	0.751%
		IPER	191532	1023	0.534 %
Web	TCP	IPLR	100	0	0 %
		IPDR	100	0	0 %
		IPER	100	0	0 %
FTP	TCP	IPLR	152	0	0 %
		IPDR	152	0	0 %
		IPER	152	0	0 %

Sin embargo, para observar cómo estas condiciones evolucionan a lo largo de un período mayor, se ha llevado a cabo un análisis adicional en un intervalo de 20 minutos, cuyos resultados se presentan en la Tabla 47. Esta comparación permitirá evaluar si los patrones de pérdida de paquetes, duplicación y errores observados persisten o aumentan con el tiempo, proporcionando una visión más completa del rendimiento de la red bajo cargas prolongadas.

Los resultados obtenidos en el intervalo de 20 minutos indican que el comportamiento de servicios como VoIP y VoD en la red híbrida SND se mantiene dentro de parámetros aceptables, evidenciando ligeras variaciones en las métricas de rendimiento. Por ejemplo, la tasa de pérdida de paquetes en el servicio de VoIP disminuye de 0.141 % a 0.100 %, lo que sugiere que el tráfico en este servicio se estabiliza con el tiempo existiendo una menor pérdida de paquetes. En contraste, el servicio VoD muestra un aumento en los valores de IPLR e IPER, alcanzando 0.811 % y 0.616 %,

respectivamente, lo que atribuye al incremento en la cantidad de paquetes de video procesados en intervalos más largos.

Adicionalmente, los resultados recopilados para los servicios Web y FTP presentan cambios mínimos en la segunda medición, aunque se destaca un ligero aumento en el IPLR del servicio Web, que pasa de 0 % a 0.899 %. Este comportamiento se debe a fluctuaciones en el tráfico o a la baja cantidad de paquetes procesados por este servicio en comparación con VoIP y VoD. No obstante, el servicio FTP mantiene un rendimiento óptimo con tasas de pérdida y error del 0 % en ambos intervalos, reafirmando su estabilidad bajo condiciones de carga prolongada.

Tabla 47
Rendimiento de la red del escenario general en un intervalo de 20 min

Servicio	Trafico	Parámetro	Total de paquetes	Paquetes filtrados	Resultado
VoIP	UDP	IPLR	62973	63	0.100%
		IPDV	62973	-	41.0888 ms
VoD	TCP	IPLR	365904	2969	0.811%
		IPDR	365904	3219	0.879%
		IPER	365904	2255	0.616 %
Web	TCP	IPLR	556	8	0.899 %
		IPDR	556	4	0.719 %
		IPER	556	2	0.359 %
FTP	TCP	IPLR	190	0	0 %
		IPDR	190	0	0 %
		IPER	190	0	0 %

4.5.2 Evaluación del primer escenario

La evaluación del primer escenario del testbed examina el impacto de un ataque de denegación de servicio genera sobre el funcionamiento de la red híbrida SDN, donde se espera que la congestión resultante afecte la infraestructura y la calidad de los servicios que proporciona. Los resultados obtenidos en este escenario tienen como finalidad identificar la degradación de la red mediante los diferentes parámetros de rendimiento

establecidos previamente para cada servicio, proporcionando información para cuantificar la magnitud del impacto adverso y comparar la eficacia de las medidas de mitigación implementadas en el escenario posterior.

En el primer intervalo de medición de 10 minutos, se capturaron un total de 956732 paquetes correspondientes al uso en paralelo de todos los servicios presentes en la red. Los resultados obtenidos en este periodo de medición, detallados en la Tabla 48, muestran un aumento considerable en la retardo, pérdida, duplicación y errores de los paquetes de datos, debido a la congestión causada por el ataque DoS a la red SDN, el cual afecta la capacidad del controlador para procesar el volumen masivo de tráfico generado por el atacante.

En el caso del servicio de VoIP, se registra un IPLR del 8.43 %, lo que implica a que se presenten interrupciones frecuentes y pérdida de calidad en las llamadas de voz, impactando la experiencia del usuario con retrasos, eco o cortes. En VoD, el IPLR registrado de 12.55 % refleja una pérdida significativa de paquetes que puede manifestarse como interrupciones en la reproducción de video, tiempos de carga prolongados o una baja resolución en la transmisión del video solicitado por el usuario.

Además, los resultados preliminares de esta medición indican que los valores de pérdida, duplicación y errores son especialmente elevados en los servicios Web y FTP, con un IPLR superior al 18% y errores de retransmisión cercanos al 17%. Este comportamiento afecta la estabilidad de las aplicaciones basadas en transferencia de archivos y navegación web. En el caso del servicio Web, el incremento en los errores de retransmisión se traduce en tiempos de carga más largos y fallos en la entrega del contenido, mientras que en FTP genera interrupciones en la transferencia de archivos o la necesidad de múltiples intentos para completar un proceso de descarga o carga.

Tabla 48
Rendimiento de la red del primer escenario en un intervalo de 10 min

Servicio	Trafico	Parámetro	Total de paquetes	Paquetes filtrados	Resultado
VoIP	UDP	IPLR	152831	12893	8.43%
		IPDV	152831	-	190.21 ms
VoD	TCP	IPLR	317529	39854	12.55 %
		IPDR	317529	43021	13.55 %
		IPER	317529	35482	11.17 %
Web	TCP	IPLR	86	20	23.25 %
		IPDR	86	17	29.77 %
		IPER	86	15	17.44 %
FTP	TCP	IPLR	114	21	18.42 %
		IPDR	114	18	15.79 %
		IPER	114	19	16.66 %

Con el objetivo de obtener una evaluación más completa y observar la evolución del rendimiento de la red bajo el ataque DoS, se recopilan los resultados en el período de evaluación correspondiente a los 20 minutos. Los resultados obtenidos, presentados en la Tabla 49, muestran una mayor degradación en los parámetros de rendimiento debido al incremento en la congestión de la red y el impacto prolongado del ataque sobre la infraestructura SDN. Esta degradación progresiva resalta la vulnerabilidad de la red frente a ataques sostenidos y la incapacidad del controlador SDN para mitigar eficazmente la congestión sin medidas adicionales de mitigación.

En el servicio de VoIP, el segundo intervalo de medición indica que los servicios de transmisión en tiempo real son particularmente sensibles a las condiciones de congestión prolongadas. Esto se fundamenta la recopilación de 300249 paquetes RTMP pertenecientes al servicio de telefonía IP, en los cuales se presenta un IPLR correspondiente al 8.70% y un retardo promedio (IPDV) de 195.62 ms, lo que refleja la existencia de una degradación progresiva respecto al intervalo de 10 minutos, amplificando los problemas en la comunicación de voz, que se traducen en una notable

disminución de la calidad, interrupciones en las llamadas y retrasos significativos que afectan la experiencia del usuario.

Los resultados para el servicio de VoD, indican que el IPLR alcanzó el 12.65 %, acompañado de un IPDR del 13.45 % y un IPER del 10.99 %. Estos valores obtenidos en la segunda medición de este escenario del testbed, sugieren que las transmisiones de video bajo demanda presentan interrupciones más frecuentes y una disminución en la calidad de visualización a medida que el ataque persiste, lo que impacta la satisfacción del usuario con tiempos de carga más extensos y pausas inesperadas durante la reproducción.

Por su parte, los servicios correspondientes a Web y FTP presentan una notable degradación con un IPLR que alcanza el 16.52 % y 19.60 %, respectivamente, lo que genera que las páginas web tarden más tiempo en cargar o incluso el servicio es denegado completamente, mientras que las transferencias de archivos sufren múltiples errores de retransmisión que comprometen la confiabilidad del servicio. En ambos casos, la persistencia de estas condiciones deteriora la usabilidad general de la red y su capacidad para proporcionar un servicio eficiente.

Tabla 49

Rendimiento de la red del primer escenario en un intervalo de 20 min

Servicio	Trafico	Parámetro	Total de paquetes	Paquetes filtrados	Resultado
VoIP	UDP	152831	300249	26102	8.70%
		IPDV	300249	-	195.62 ms
VoD	TCP	IPLR	648312	82013	12.65 %
		IPDR	648312	87346	13.45 %
		IPER	648312	71230	10.99 %
Web	TCP	IPLR	242	40	16.52 %
		IPDR	242	38	15.70 %
		IPER	242	33	17.44 %
FTP	TCP	IPLR	153	30	19.60 %
		IPDR	153	24	15.68 %
		IPER	153	21	16.66 %

Finalmente, los datos obtenidos en las dos mediciones establecen que el impacto de los ataques DoS no solo se mantiene, sino que aumenta significativamente con el tiempo, generando una acumulación de tráfico malicioso que intensifica la congestión de la red y reduce aún más la capacidad del controlador SDN para priorizar y gestionar el tráfico legítimo. Además, reconocer este tipo de comportamiento en la red destaca la necesidad de implementar estrategias de mitigación que no solo respondan al ataque, sino que también prevengan su escalamiento y minimicen su impacto prolongado en la red.

4.5.3 Evaluación del segundo escenario

La evaluación del segundo escenario tiene como propósito analizar el impacto del mecanismo de seguridad integrado a las funciones del controlador SDN sobre el rendimiento de la red durante un ataque de denegación de servicio. La recopilación de los resultados en este escenario busca validar la capacidad del IDS/IPS implementado para detectar y mitigar el trafico malicioso en la red, minimizando la congestión en la infraestructura y garantizando una calidad de servicio aceptable para los usuarios finales, de modo que, a través de esta evaluación, se cuantifica el nivel de mejora en los parámetros de rendimiento de la red respecto al escenario anterior con la finalidad de presentar resultados similares al escenario de funcionamiento ideal.

Durante el primer intervalo de medición, correspondiente a los primeros 10 minutos de la evaluación, se obtuvieron un total de 892524 paquetes del funcionamiento en paralelo de todos los servicios que la red que posee en conjunto con el despliegue de los ataques y el mecanismo de seguridad. Los datos registrados, detallados en la Tabla 50, muestran una reducción significativa en los indicadores de degradación del tráfico como el retardo, pérdida, duplicación y errores de los paquetes de datos, en los servicios evaluados.

En el servicio de telefonía IP, el funcionamiento del mecanismo de seguridad implementado a las funciones del controlador SDN permitió mantener el IPLR en un 0.152 %, lo que conlleva a un nivel imperceptible de interrupciones en las llamadas al no poseer un alto porcentaje de pérdidas de paquetes RTP, de igual manera el retardo promedio (IPDV) fue de 40.7538 ms, manteniéndose dentro de los umbrales recomendados para aplicaciones en tiempo real, lo que indica una mejora en la estabilidad del servicio bajo ataque.

Para el servicio de video bajo demanda, los datos recopilados evidencian una mejora en los parámetros de rendimiento, ya que se ha registrado un IPLR de 0.716 %, acompañado de un IPDR de 0.760 % y un IPER de 0.545 %. Los valores reflejados en estas métricas establecen que el IDS/IPS implementado es capaz de denegar los flujos de tráfico ilegítimos para no ser procesados en la red SDN, y a la vez priorizar los flujos de tráfico legítimo para garantizar en este servicio tiempos de carga mínimos y una reproducción continua del contenido.

En los servicios Web y FTP, la intervención del IDS/IPS resultó en un rendimiento prácticamente óptimo, debido a que el servicio Web no registró pérdida de paquetes con un IPLR del 0 %, mientras que en FTP el IPLR se limitó a 0.67 %. Estos resultados destacan la eficiencia del sistema para mitigar las afectaciones generadas al basado en conexiones, como transferencias de archivos y navegación web, sin que se generen errores de retransmisión significativos.

Tabla 50

Rendimiento de la red del segundo escenario en un intervalo de 10 min

Servicio	Trafico	Parámetro	Total de paquetes	Paquetes Filtrados	Resultado
VoIP	UDP	IPLR	29637	45	0.152%
		IPDV	29637	-	40.7538 ms
VoD	TCP	IPLR	191560	1372	0.716%
		IPDR	191560	1456	0.760%
		IPER	191560	1045	0.545 %
Web	TCP	IPLR	101	0	0 %
		IPDR	101	1	0.99 %
		IPER	101	0	0 %
FTP	TCP	IPLR	150	1	0.67 %
		IPDR	150	0	0 %
		IPER	150	0	0 %

Para ofrecer una visión más completa del impacto del mecanismo de seguridad en la red, se establece el segundo intervalo de análisis correspondiente a los 20 minutos. Los resultados, presentados en la Tabla 51, muestran una mejora sostenida en todos los indicadores de rendimiento, a pesar de la prolongación del ataque.

En el caso del servicio de telefonía IP, el parámetro IPLR se redujo aún más a 0.103 % de paquetes perdidos, mientras que el IPDV se mantuvo estable en 41.188 ms, demostrando que el IDS/IPS puede manejar un volumen creciente de tráfico malicioso sin comprometer el rendimiento de aplicaciones sensibles al tiempo. Por otro lado, en VoD, el IPLR fue de 0.806 %, lo que refuerza la capacidad del sistema para garantizar la calidad del servicio en transmisiones multimedia durante periodos prolongados de los ataques hacia la red.

Los datos obtenidos en los servicios Web y FTP poseen el mismo comportamiento, con un valor del parámetro de IPLR inferior al 1 % en ambos casos y la eliminación de errores de retransmisión en FTP. Esto establece que la integración del

IDS/IPS no solo protege la infraestructura, sino que también optimiza el procesamiento del tráfico legítimo, minimizando las afectaciones en la experiencia del usuario.

Tabla 51
Rendimiento de la red del segundo escenario en un intervalo de 20 min

Servicio	Trafico	Parámetro	Total de paquetes	Paquetes filtrados	Resultado
VoIP	UDP	IPLR	62971	65	0.103%
		IPDV	62971	-	41.188 ms
VoD	TCP	IPLR	365900	2950	0.806%
		IPDR	365900	3205	0.876%
		IPER	365900	2220	0.607 %
Web	TCP	IPLR	558	5	0.897 %
		IPDR	558	5	0.896 %
		IPER	558	2	0.358 %
FTP	TCP	IPLR	189	0	0 %
		IPDR	189	0	0 %
		IPER	189	0	0 %

En resumen, los resultados demuestran que el mecanismo de seguridad integrado en el controlador ha reducido significativamente el impacto de los ataques DoS, manteniendo un rendimiento estable incluso durante periodos prolongados de ataque. Los valores obtenidos confirman la efectividad del sistema IDS/IPS como una herramienta esencial para mitigar ataques de tipo inundación en redes SDN, al reducir la carga sobre el controlador, estabilizar los parámetros de rendimiento y garantizar la continuidad operativa de los servicios.

4.6 Comparativa de resultados

En esta sección se presentan los resultados obtenidos de la evaluación de los diferentes bancos de pruebas del testbed analizados correspondiente al escenario general, escenario bajo ataque DoS y escenario de mitigación, de modo que los valores obtenidos en cada intervalo de medición permiten recopilar y analizar el rendimiento de la red híbrida SDN y las aplicaciones en distintas condiciones de uso a lo largo del tiempo.

Comenzando con servicios críticos como la telefonía IP, que es utilizada en la infraestructura SDN, los resultados de su rendimiento bajo los escenarios del testbed están recopilados en la Tabla 52. Los resultados obtenidos indican que, en el escenario general, los valores de **p**érdida de paquetes se mantuvieron bajos tanto en 10 minutos (0.141%) como en 20 minutos (0.100%). Sin embargo, bajo ataque DoS, se observa un aumento significativo en la pérdida de paquetes, alcanzando el 8.43% en 10 minutos y 8.70% en 20 minutos, lo que evidencia un deterioro en la calidad de este servicio debido a este tipo de anomalías presentes en la red. No obstante, en el escenario de mitigación los valores de IPLR disminuyen considerablemente, casi igualando los niveles del escenario general con valores aproximados de 0.152% en la primera medición y 0.103% en la segunda medición realizada.

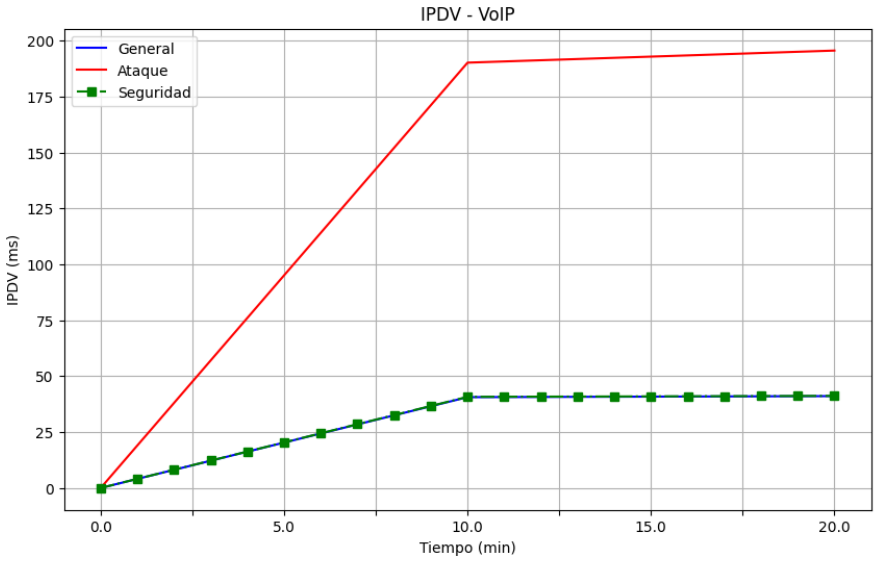
Respecto a la variación del retardo de los paquetes, el escenario general muestra una latencia aceptable correspondiente a 40.6134 ms en la primera evaluación de rendimiento y 41.0888 ms en la segunda. Durante el ataque DoS a la infraestructura SDN, el retardo aumenta drásticamente, alcanzando los 190.21 ms en 10 minutos y 195.62 ms en 20 minutos, siendo valores muy altos para considerarse el servicio en términos adecuados de funcionamiento. Esto conlleva a que la mitigación por parte del mecanismo de seguridad minimice su impacto, generando que la latencia se estabiliza en valores similares a los del escenario general correspondientes a los 40.7538 ms y 41.188 ms para cada medición respectivamente.

Tabla 52
Comparativa de rendimiento del servicio VoIP

Parámetro	Escenario General (10 min)	Escenario General (20 min)	Ataque DoS (10 min)	Ataque DoS (20 min)	Mitigación (10 min)	Mitigación (20 min)
IPLR	0.141%	0.100%	8.43%	8.70%	0.152%	0.103%
IPDV	40.613 ms	41.088 ms	190.2 ms	195.6 ms	40.753 ms	41.188 ms

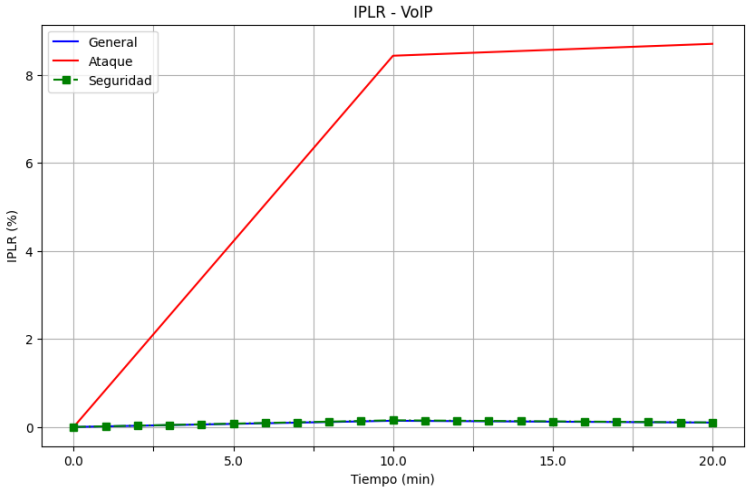
La Figura 81 presenta los datos de la Tabla 52 de forma gráfica, lo que permite visualizar de manera más clara el comportamiento del rendimiento del servicio VoIP bajo los tres escenarios evaluados. En la gráfica de IPDV, se observa cómo el retardo de paquetes permanece bajo y constante en los escenarios general y de mitigación, manteniéndose en valores cercanos a los 40 ms. Sin embargo, durante el ataque DoS, el retardo incrementa de manera abrupta, alcanzando valores superiores a los 190 ms, lo cual corrobora el impacto significativo de este tipo de ataque en la calidad de este servicio.

Figura 81
Gráfica de IPDV del servicio de VoIP



Por otro lado, la gráfica presentada en la Figura 82 de IPLR muestra una tendencia similar, con pérdidas de paquetes mínimas en los escenarios general y de mitigación, mientras que en el escenario de ataque DoS se evidencia un aumento exponencial, superando el 8% de pérdida de paquetes del total analizado, de modo que estas visualizaciones refuerzan los resultados numéricos presentados en la tabla, destacando la eficacia del mecanismo de mitigación al devolver los valores de rendimiento a niveles aceptables para la operación del servicio.

Figura 82
Gráfica de IPLR del servicio de VoIP



En lo que respecta a los resultados del rendimiento del servicio de VoD, la Tabla 53 presenta los valores recopilados de las mediciones en cada uno de los casos de estudio del testbed. Este servicio experimenta un comportamiento similar al servicio de VoIP, ya que el IPLR en el escenario general es moderado correspondiente al 0.706% y 0.811% del total de paquetes en cada uno de los dos puntos de medición. Sin embargo, el ataque DoS provoca un incremento sustancial, con pérdidas de hasta 12.65% de todos los paquetes transmitidos, de modo que, tras la implementación de las medidas de mitigación, el IPLR se reduce notablemente de 0.716% en la primera recopilación de datos y de 0.806% en la segunda recopilación, lo que demuestra la eficacia del sistema de seguridad en la mitigación de los ataques.

En cuanto a la duplicación de paquetes, el escenario DoS incrementa significativamente este valor (13.55% y 13.45%), mientras que en el escenario de mitigación mediante sus mecanismos de mitigación vuelve los valores a niveles aceptables de operación correspondiente a los 0.760% y 0.876%, asegurando que los mecanismos implementados, como la redirección de tráfico y filtrado de paquetes, reduzcan el impacto de los ataques DoS y restauren el rendimiento de los servicios de

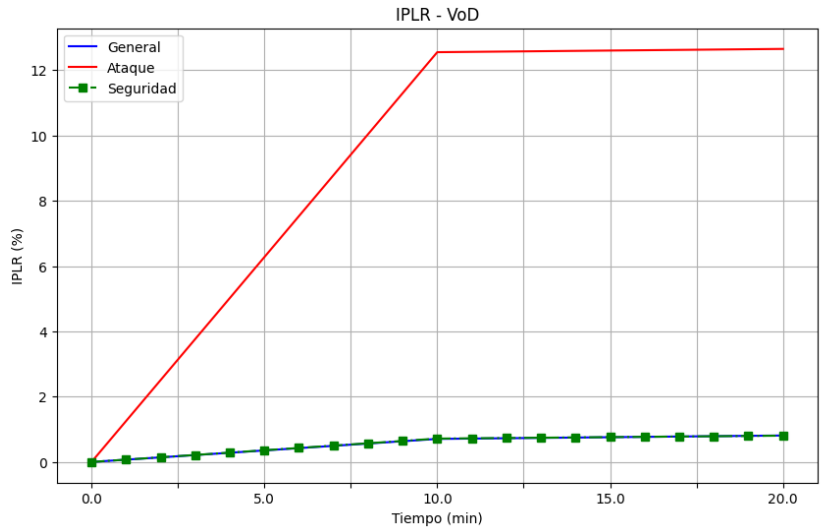
VoD. Esta reducción en la duplicación de paquetes mejora la calidad del servicio, disminuyendo la tasa de pérdida de paquetes y evitando la congestión excesiva de la red, lo que a su vez optimiza la experiencia de usuario.

Tabla 53
Comparativa de rendimiento del servicio de VoD

Parámetro	Escenario General (10 min)	Escenario General (20 min)	Ataque DoS (10 min)	Ataque DoS (20 min)	Mitigación (10 min)	Mitigación (20 min)
IPLR	0.706%	0.811%	12.55%	12.65%	0.716%	0.806%
IPDR	0.751%	0.879%	13.55%	13.45%	0.760%	0.876%
IPER	0.534%	0.616%	11.17%	10.99%	0.545%	0.607%

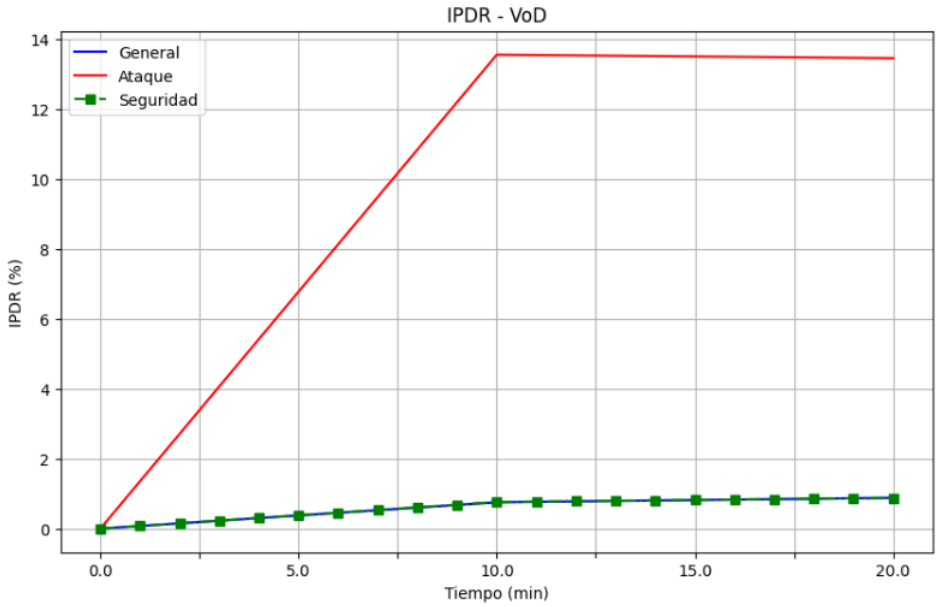
La Figura 83 presenta de manera gráfica los datos recopilados en la Tabla 53, lo que permite observar con mayor claridad el comportamiento del rendimiento del servicio VoD bajo los tres escenarios evaluados. En la gráfica de IPLR, se evidencia cómo las pérdidas de paquetes se mantienen en niveles bajos y constantes durante los escenarios general y de mitigación, con valores cercanos al 0.8%. No obstante, durante el ataque DoS, estas pérdidas aumentan drásticamente, alcanzando valores superiores al 12.6%, lo que confirma el impacto severo de este tipo de ataque en la estabilidad del servicio.

Figura 83
Gráfica de IPLR del servicio de VoD



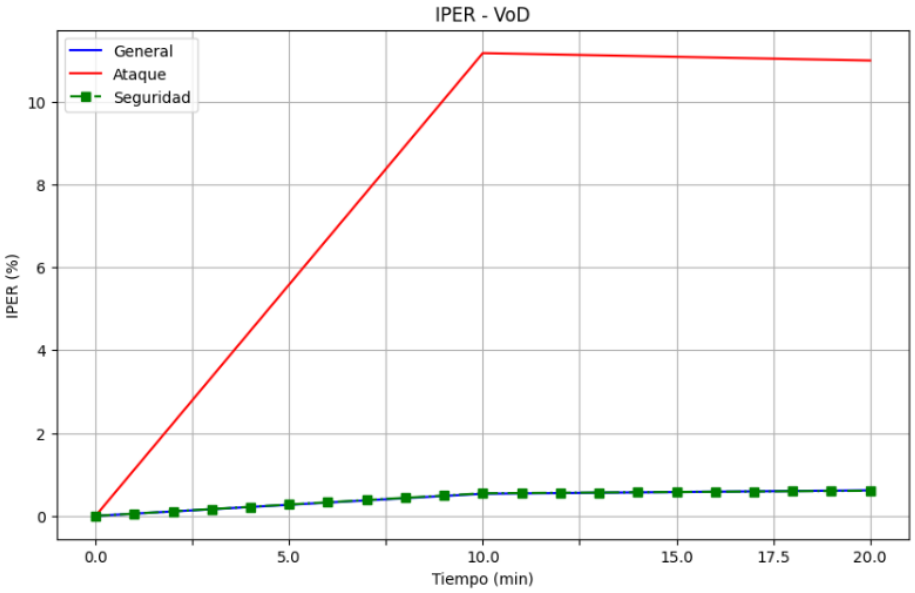
De manera similar, en la gráfica de la Figura 84 correspondiente al parámetro IPDR del servicio de VoD se observa cómo la duplicación de paquetes sigue un patrón comparable: mientras que en los escenarios general y de mitigación los valores permanecen bajos en torno al 0.8%, de modo que cuando el ataque DoS es generado a la red, provoca un incremento significativo, superando el 13%.

Figura 84
Gráfica de IPDR del servicio de VoD



Por último, la gráfica de IPER presentada en la Figura 85 muestra un comportamiento congruente con las métricas anteriores, donde los errores de retransmisión se disparan durante el ataque y se reducen considerablemente tras la implementación de las medidas de seguridad.

Figura 85
Gráfica de IPDR del servicio de VoD



Los resultados obtenidos para identificar el desempeño del servicio web están recopilados en la Tabla 54 donde los valores del parámetro IPLR destacan los impactos severos del ataque DoS en el funcionamiento del servidor web. Esto se debe a que el porcentaje de paquetes perdidos alcanza niveles críticos, lo que afecta directamente la calidad del servicio. En la primera fase del ataque, el IPLR llega a un 23.25% en los primeros 10 minutos. Sin embargo, después de 20 minutos, este valor disminuye al 16.52%, lo que podría deberse a una estabilización del tráfico malicioso. Este descenso relativo sugiere que, aunque el ataque sigue activo, el impacto se atenúa ligeramente a medida que el tráfico malicioso y legítimo interactúan.

Con la mitigación implementada, los resultados muestran una mejora significativa, logrando que el IPLR se reduzca a niveles casi imperceptibles, acercándose a los valores observados en condiciones normales sin ataque, lo que confirma la eficacia de las contramedidas aplicadas. Además, otros parámetros como el IPDR e IPER también muestran incrementos pronunciados durante el ataque, alcanzando un 29.77% y 17.44%

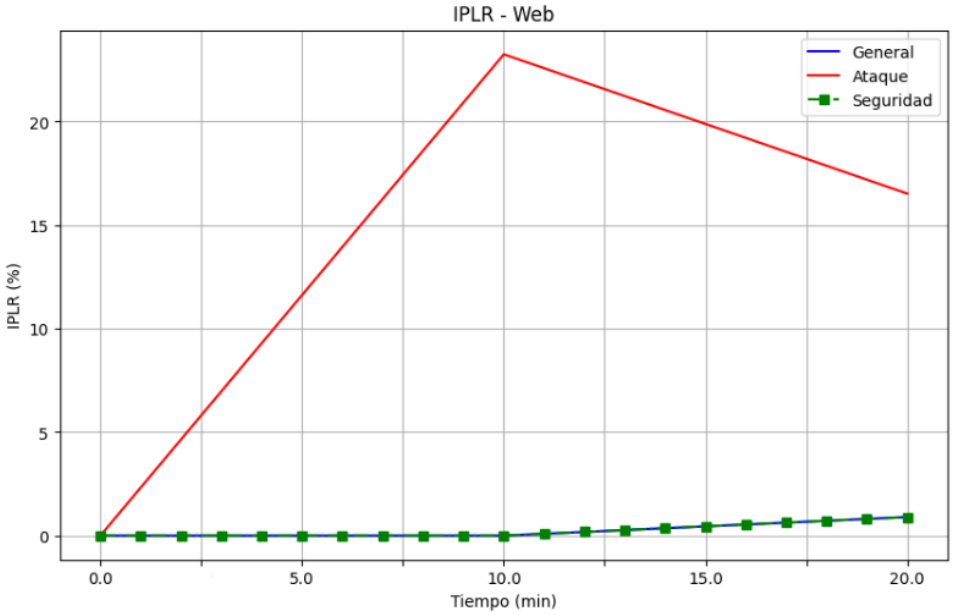
respectivamente en cada medición. Sin embargo, una vez aplicada la mitigación, estos valores disminuyen considerablemente, restaurando el rendimiento del servicio web a niveles aceptables y estables, lo que refuerza la importancia del sistema de seguridad.

Tabla 54
Comparativa de rendimiento del servicio web

Parámetro	Escenario General (10 min)	Escenario General (20 min)	Ataque DoS (10 min)	Ataque DoS (20 min)	Mitigación (10 min)	Mitigación (20 min)
IPLR	0%	0.899%	23.25%	16.52%	0%	0.897%
IPDR	0%	0.719%	29.77%	15.70%	0.99%	0.896%
IPER	0%	0.359%	17.44%	17.44%	0%	0.358%

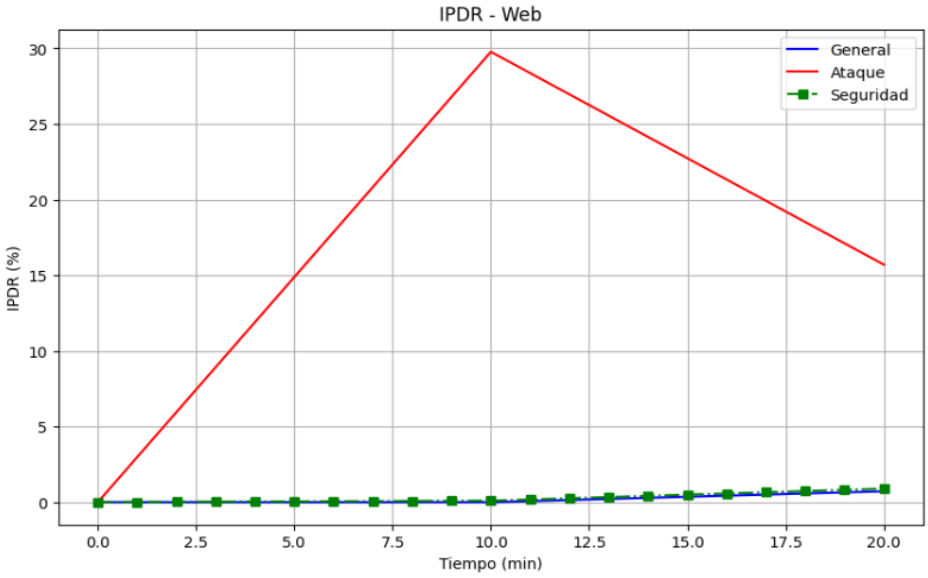
La Figura 86 ilustra los resultados recopilados para el servicio Web, destacando el comportamiento de las métricas de rendimiento en los tres escenarios evaluados. En la gráfica de IPLR, se observa un aumento drástico en las pérdidas de paquetes durante el ataque DoS, alcanzando su punto máximo de más del 20% alrededor de los 10 minutos. Este comportamiento puede atribuirse al colapso temporal de los recursos de red debido a la saturación generada por el ataque. Posteriormente, la caída en las pérdidas podría explicarse por una posible estabilización momentánea del servicio, no obstante, cuenta con altos porcentajes de pérdidas del total de paquetes analizados lo que le indica que la red aún sigue comprometida.

Figura 86
Gráfica de IPLR del servicio de Web



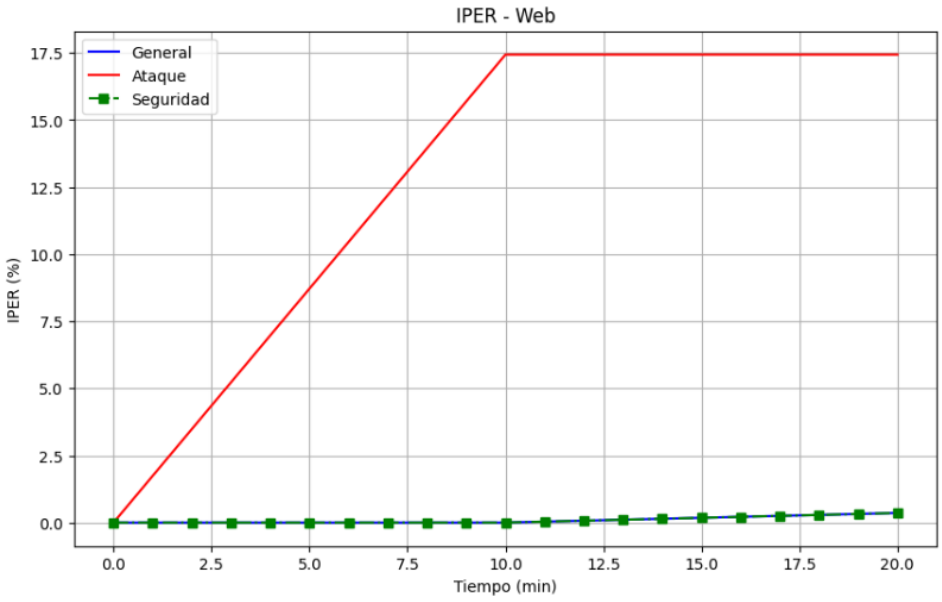
De manera similar, en la gráfica presentada en la Figura 87 correspondiente al parámetro de IPDR, la duplicación de paquetes muestra una tendencia abrupta de aumento, alcanzando su pico de más del 30% en el mismo intervalo de tiempo. Esta duplicación excesiva podría ser el resultado de retransmisiones automáticas generadas por el congestionamiento de la red y el mal manejo de paquetes durante el ataque. La caída posterior en la duplicación, al igual que en el caso del IPLR, refleja una posible recuperación parcial de la red, pero con índices de porcentajes de duplicación de paquetes aún elevados que comprometen su funcionamiento.

Figura 87
Gráfica de IPDR del servicio de Web



Por último, la gráfica de IPER para el servicio web, presentada en la Figura 88, muestra un comportamiento congruente con las métricas anteriores, donde los errores de retransmisión se disparan durante el ataque y se reducen considerablemente tras la implementación de las medidas de seguridad a niveles óptimos de operación.

Figura 88
Gráfica de IPER del servicio de Web



En el caso del servicio FTP los datos obtenidos en los escenarios de pruebas y recopilados en la Tabla 55 indican que, durante el ataque DoS, el parámetro IPLR que identifica la tasa de pérdida de paquetes se incrementa a un 18.42% del total de paquetes identificados, lo que indica una severa interrupción en la transmisión de datos debido a la saturación de la red por tráfico malicioso. Esta situación empeora ligeramente a un 19.60% en la segunda medición, debido a que el ataque persiste y genera congestión constante en el tráfico de este servicio. Sin embargo, cuando se implementan las medidas de mitigación, los valores del IPLR disminuyen drásticamente, pasando a un 0.67% en los primeros 10 minutos y llegando a **0%** en los 20 minutos, por ende, este descenso tan pronunciado demuestra la efectividad del mecanismo de seguridad, ya que logra filtrar el tráfico malicioso y restaurar el flujo normal de datos.

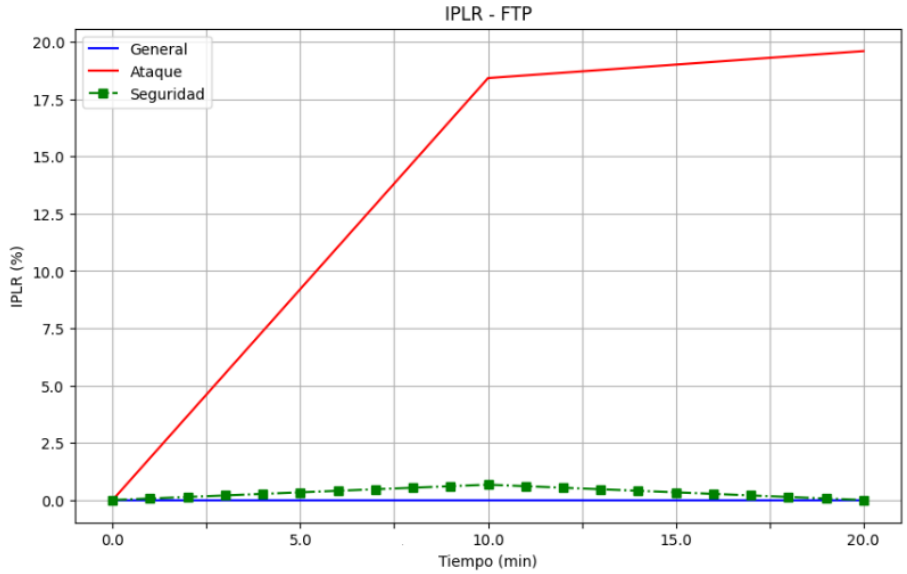
Además, la duplicación de paquetes y los errores de retransmisión, que también sufren incrementos durante el ataque, reflejan un entorno de red ineficiente donde los paquetes se retrasan o se retransmiten erróneamente debido a la sobrecarga que posee la infraestructura SDN frente al ataque. Con la mitigación implementada, estos problemas disminuyen de forma notable, lo que indica que la red puede manejar el tráfico de forma más efectiva, garantizando una mayor fiabilidad en la transferencia de datos, lo que conlleva a que la reducción de estos problemas mejora el rendimiento general del servicio FTP, devolviéndolo a niveles de operación normal.

Tabla 55
Comparativa de rendimiento del servicio FTP

Parámetro	Escenario General (10 min)	Escenario General (20 min)	Ataque DoS (10 min)	Ataque DoS (20 min)	Mitigación (10 min)	Mitigación (20 min)
IPLR	0%	0%	18.42%	19.60%	0.67%	0%
IPDR	0%	0%	15.79%	15.68%	0%	0%
IPER	0%	0%	16.66%	16.66%	0%	0%

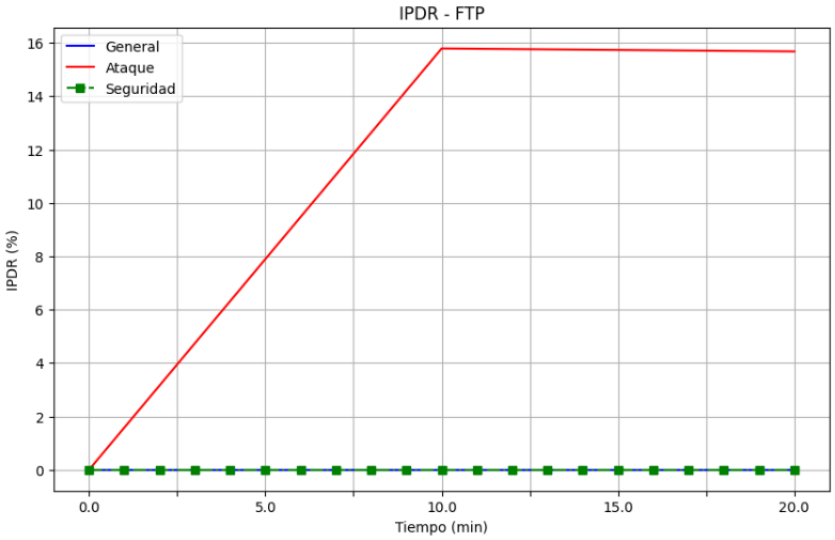
La Figura ilustra los resultados recopilados para el servicio FTP, destacando el comportamiento de las métricas de rendimiento en los tres escenarios evaluados. En la gráfica correspondiente a IPLR, se observa un aumento drástico en las pérdidas de paquetes durante el ataque DoS, alcanzando su punto máximo de más del 18% alrededor de los 10 minutos y alcanzando un tope máximo de 19.5%, indicando la existencia de problemas en la operación del servicio al transferir archivos. No obstante, las mediciones en el escenario de mitigación poseen valores de operación similar al escenario ideal de la red.

Figura 89
Gráfica de IPLR del servicio FTP



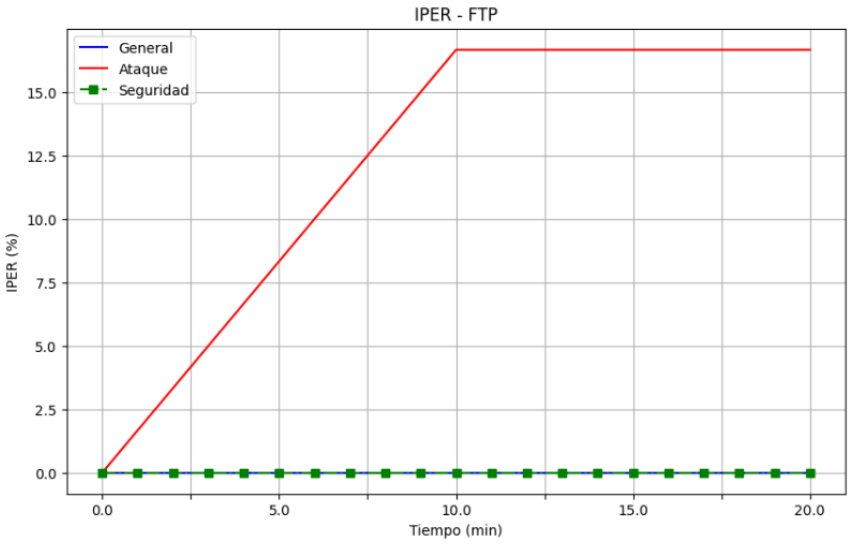
En lo que respecta a la duplicación de paquetes, la gráfica de la Figura # muestra una tendencia abrupta de aumento, alcanzando su pico de más del 15% en el mismo intervalo de tiempo de 10 minutos, alcanzando un máximo del 16% en la segunda medición. El patrón de mitigación sigue igual, en donde se identifica una reducción en este parámetro, obteniendo valores similares al escenario general indicando que el servicio posee una adecuada operación.

Figura 90
Gráfica de IPDR del servicio FTP



Finalmente, el parámetro de errores en el servicio FTP presentado en la Figura # para los tres escenarios del testbed, establece comportamiento similar indicando un máximo del 16 % de los paquetes con errores identificados en el escenario de ataques DoS. En el escenario general y mitigación se presentan valores similares de menos del 1% de paquetes con errores cuando el servicio se encuentra en la red con el mecanismo de seguridad implementado.

Figura 91
Gráfica de IPER del servicio FTP



Al analizar los resultados obtenidos en el testbed, se evidencia que el mecanismo de seguridad implementado en la red SDN es una solución efectiva frente a los ataques DoS de tipo inundación. Esto se debe, en primer lugar, a la reducción significativa de la pérdida de paquetes (IPLR) en todos los servicios analizados, debido a que, durante los ataques, los valores de IPLR aumentaron de manera drástica, llegando hasta un 23% de paquetes perdidos del total de paquetes transferidos de cada servicio. Sin embargo, tras la implementación de la mitigación, los valores se redujeron considerablemente, acercándose a los niveles observados en condiciones normales, lo que conlleva a que el mecanismo de seguridad es eficaz en filtrar el tráfico malicioso, preservando la calidad del servicio para los usuarios legítimos.

Además, el análisis del retardo y la variación del retardo (IPDV) muestra que el mecanismo de mitigación estabiliza el comportamiento de la red bajo este tipo de amenazas de seguridad. Durante los ataques DoS hacia la infraestructura SDN, el servicio de VoIP presentó un aumento significativo en la variación del retardo, alcanzando 195.62 ms máximo de jitter, dejando totalmente inaccesible al servicio a los usuarios legítimos. No obstante, con la mitigación activa, este valor volvió a estabilizarse en niveles cercanos al escenario sin ataque, lo que demuestra la capacidad del mecanismo de seguridad para gestionar la congestión y mantener un tiempo de respuesta óptimo en condiciones adversas.

Finalmente, otro aspecto relevante de la solución de seguridad implementada en la infraestructura SDN es el control sobre la duplicación de paquetes y los errores de retransmisión (IPDR e IPER), debido a que durante los ataques estos parámetros se incrementaron considerablemente en todos los servicios, con picos del 29.77% en duplicación de paquetes en el servicio Web y del 17.44% en errores de retransmisión en

el servicio FTP. Pese a ello, con la implementación de la mitigación, estos valores volvieron a niveles normales, lo que demuestra que la red SDN equipada con medidas de seguridad puede restaurar la eficiencia de la transmisión de datos y optimizar el uso de los recursos de red.

Conclusiones Y Recomendaciones

Conclusiones

La integración de un sistema IDS/IPS con el controlador SDN, utilizando herramientas de software libre como Snort y Ryu, demostró ser una solución efectiva para mitigar ataques de denegación de servicio (DoS) en redes híbridas SDN, de modo que este enfoque combinó la capacidad del controlador de gestionar el tráfico de manera dinámica a través del protocolo OpenFlow con la detección y respuesta rápida ante amenazas generadas por hosts maliciosos identificadas por parte del mecanismo de seguridad. Por ende, la sinergia entre el IDS/IPS y el controlador permitió no solo encaminar eficientemente el tráfico legítimo, sino también tomar medidas correctivas ante posibles ataques a la infraestructura de red, asegurando así funcionamiento y repercusiones.

Los resultados de las pruebas en los dos escenarios evaluados confirman esta mejora. En el primer escenario, una red SDN sin mecanismos de seguridad mostró una degradación significativa bajo un ataque DoS, lo que evidenció la vulnerabilidad de los servicios como VoIP, VoD, Web y FTP ante la congestión provocada por estos ataques. Sin embargo, en el segundo escenario, con la implementación del IDS/IPS, la red mostró una notable capacidad para resistir y recuperarse de los ataques generados, con una mejora visible en la pérdida de paquetes, duplicación y errores de retransmisión, además de una reducción en los tiempos de retardo.

Las políticas de bloqueo de tráfico por direcciones MAC contribuyeron a establecer reglas de flujo que se distribuyeron de manera efectiva entre los conmutadores SDN, proporcionando una protección proactiva contra ataques futuros. Además, las pruebas demostraron que el IDS/IPS optimizó el uso de recursos del sistema, mejorando el rendimiento general del controlador SDN al reducir la carga en la CPU al no manejar

cantidades masivas de tráfico ilegitimo y a la vez acelerar la respuesta a las alertas de seguridad.

Finalmente, la integración de un IDS/IPS con el controlador SDN no solo reforzó la seguridad ante ataques DoS, sino que también aumentó la eficiencia operativa de la red. Estos resultados respaldan la adopción de mecanismos de seguridad basados en software libre como una solución viable y eficiente para garantizar la continuidad y el rendimiento de las redes SDN frente a amenazas de seguridad.

.

Recomendaciones

Como resultado de esta investigación, se recomienda la implementación generalizada de sistemas IDS/IPS en redes SDN, aprovechando herramientas de software libre como Snort y el controlador Ryu. Esto se debe a que esta operación en conjunto ha demostrado ser una solución eficaz para mitigar ataques de denegación de servicio (DoS) sin comprometer el rendimiento general de la red, de modo que integrar estas tecnologías puede fortalecer significativamente la seguridad de las redes, especialmente en entornos donde la dinámica del tráfico requiere una gestión flexible y eficiente a través del controlador SDN.

Además, se sugiere continuar optimizando las políticas de bloqueo de tráfico malicioso, específicamente mediante el uso de direcciones MAC para establecer reglas de flujo que se distribuyan de manera efectiva entre los conmutadores SDN. Esta estrategia demostró ser proactiva para proteger la red de futuros ataques de los mismos hosts maliciosos, no obstante, su mejora continua podría brindar aún más seguridad y una mayor capacidad de respuesta frente a nuevas amenazas correspondientes a otros tipos de ataques.

Es importante destacar que algunos servicios, como VoIP y FTP, fueron más vulnerables a los ataques DoS durante las pruebas realizadas. Por ello, se recomienda investigar métodos adicionales para reducir el impacto de estos ataques en diferentes tipos de servicios y encontrar soluciones que puedan mitigar las interrupciones sin afectar la calidad del servicio ofrecido por la red.

También es fundamental monitorear de manera continua el uso de los recursos del sistema, como la CPU del controlador SDN, y ajustar el funcionamiento del IDS/IPS para mantener un equilibrio óptimo entre seguridad y eficiencia. La integración de este

mecanismo de seguridad ha demostrado mejorar el rendimiento del controlador al reducir la carga de procesamiento, por lo que implementar herramientas de supervisión regular permitirá mantener una operación eficiente e identificar anomalías en su funcionamiento.

Además, se recomienda explorar nuevas tecnologías complementarias, como el uso de inteligencia artificial o aprendizaje automático, para mejorar la detección de amenazas avanzadas en tiempo real. En este contexto, una de las primeras aplicaciones sería la creación de datasets específicos que recopilen características detalladas de distintos tipos de ataques, incluyendo patrones de tráfico, firmas de comportamiento anómalo y correlaciones temporales. Estos datasets permitirían entrenar modelos de aprendizaje automático y usarlos en el controlador para ser capaz de identificar ataques como DoS, DDoS e intrusiones más complejas en base a la información recopilada.

Otra función clave sería la integración de sistemas de respuesta automatizada, donde la IA, integrada a las funciones del controlador SDN, pueda tomar decisiones en tiempo real, como reconfigurar reglas de flujo, bloquear direcciones IP maliciosas o aislar segmentos específicos de la red. Adicionalmente, los sistemas basados en análisis predictivo podrían identificar actividades sospechosas antes de que se materialicen como ataques, empleando algoritmos como árboles de decisión, máquinas de soporte vectorial (SVM) o redes neuronales profundas, que puedan clasificar los flujos de tráfico en categorías como normal, sospechoso o malicioso.

Por último, algoritmos de clustering (como K-means o DBSCAN) podrían ser utilizados para agrupan datos similares para distinguir entre comportamientos típicos o anómalos en la red, mientras que redes neuronales profundas, diseñadas para adaptarse a redes con alta variabilidad de tráfico, fortalecerían la capacidad de detección. Estas tecnologías, en conjunto, no solo mejorarían la seguridad de las redes SDN frente a

amenazas, sino también optimizarían su operación al reducir la carga del sistema y asegurar la continuidad de los servicios.

Referencias Bibliográficas

Ahmad, I., Namal, S., Ylianttila, M., & Gurtov, A. (2015). Security in Software Defined Networks: A Survey. *IEEE Communications Surveys & Tutorials*, *17*(4), 2317–2346. https://doi.org/10.1109/COMST.2015.2474118

Ahmed, U., Raza, I., Hussain, S. A., Ali, A., Iqbal, M., & Wang, X. (2015). Modelling cyber security for software-defined networks those grow strong when exposed to threats: Analysis and propositions. *Journal of Reliable Intelligent Environments*, *1*(2–4), 123–146. https://doi.org/10.1007/S40860-015-0008-0/TABLES/3

Ali, J., Lee, G. M., Roh, B. H., Ryu, D. K., & Park, G. (2020). Software-Defined Networking Approaches for Link Failure Recovery: A Survey. *Sustainability 2020, Vol. 12, Page 4255*, *12*(10), 4255. https://doi.org/10.3390/SU12104255

Apache Software Foundation. (2011). *Apache Karaf Version 3.0.5 Apache Karaf Users' Guide*. http://www.princexml.com

Bone, M., Rodríguéz, J., Sosa, S., & Núñez, L. (2021). Vista de Perspectivas de aplicación e investigación en Software Defined Networking SDN. *Conciencia Digital*, *4*(1). https://cienciadigital.org/revistacienciadigital2/index.php/ConcienciaDigital/article/view/1538/3890

Bujedo, L., Perazzi, E., Borja, A., & Ortin, E. (2022). *Software Defined Networks*. https://rdu.iua.edu.ar/bitstream/123456789/1486/1/Seguridad%20y%20rendimiento%20en%20redes%20hibridas%20SDN.pdf

Cisco. (2023a). *Apertura de la ruta más corta primero (OSPF): guía de diseño - Cisco*. https://www.cisco.com/c/es_mx/support/docs/ip/open-shortest-path-first-ospf/7039-1.html

Cisco. (2023b). *Configuración de ejemplo para la autenticación en RIPv2 - Cisco*. https://www.cisco.com/c/es_mx/support/docs/ip/routing-information-protocol-rip/13719-50.html

Cisco. (2023c). *Configure la adyacencia IS-IS y los tipos de área. - Cisco*. https://www.cisco.com/c/es_mx/support/docs/ip/integrated-intermediate-system-to-intermediate-system-is-is/200293-IS-IS-Adjacency-and-Area-Types.html

Cisco. (2023d). *¿Qué es la distancia administrativa? - Cisco*. https://www.cisco.com/c/es_mx/support/docs/ip/border-gateway-protocol-bgp/15986-admin-distance.html

Cuenca Pérez, G., & Flores Marín, M. (2015). Redes definidas por software: Solución para servicios portadores del Ecuador. *Investigatio*, *6*. https://doi.org/10.31095/investigatio.2015.6.2

Dargahi, T., Caponi, A., Ambrosin, M., Bianchi, G., & Conti, M. (2017). A Survey on the Security of Stateful SDN Data Planes. *IEEE Communications Surveys & Tutorials*, *19*(3), 1701–1725. https://doi.org/10.1109/COMST.2017.2689819

Farahmandian, S., & Hoang, D. B. (2016). *SECURITY FOR SOFTWARE-DEFINED (CLOUD, SDN AND NFV) INFRASTRUCTURES-ISSUES AND CHALLENGES*. 13–24. https://doi.org/10.5121/csit.2016.61502

Fares, A. A. Y. R., De Caldas Filho, F. L., Giozza, W. F., Canedo, E. D., De Mendonca, F. L. L., & Nze, G. D. A. (2019). DoS attack prevention on IPS SDN networks. *WCNPS 2019 - Workshop on Communication Networks and Power Systems*. https://doi.org/10.1109/WCNPS.2019.8896233

Fernández, M., & Ulloa, R. (2016). Despliegue de una red SDN aplicando el protocolo MPLS y generando políticas de QoS para servicios de telefonía IP. *UNIVERSIDAD POLITÉCNICA SALESIANA*. https://dspace.ups.edu.ec/bitstream/123456789/12855/1/UPS-CT006705.pdf

Fuchsberger, A. (2005). Intrusion Detection Systems and Intrusion Prevention Systems. *Information Security Technical Report*, *10*(3), 134–139. https://doi.org/10.1016/J.ISTR.2005.08.001

Georgiev, M., Iyengar, S., Jana, S., Anubhai, R., Boneh, D., & Shmatikov, V. (2012). The most dangerous code in the world: Validating SSL certificates in non-browser software. *Proceedings of the ACM Conference on Computer and Communications Security*, 38–49. https://doi.org/10.1145/2382196.2382204

Gillis, A. (2023). *What is an Intrusion Prevention System (IPS)?* https://www.techtarget.com/searchsecurity/definition/intrusion-prevention

GNS3. (2023). *Software | GNS3*. https://www.gns3.com/software

GNS3. (2024). *Software | GNS3*. https://www.gns3.com/software

Huawei. (2023). *What Is IPS? How Does IPS Work?* . https://info.support.huawei.com/info-finder/encyclopedia/en/IPS.html

IETF. (2001a). *RFC 3032: MPLS Label Stack Encoding*. https://www.rfc-editor.org/rfc/rfc3032.html

IETF. (2001b, January). *Multiprotocol Label Switching Architecture*. Request for Comments: 3031.

INCIBE. (2020). *¿Qué son y para que sirven los IDS e IPS?* https://www.incibe.es/empresas/blog/son-y-sirven-los-siem-ids-e-ips

IRTF. (2015). *RFC 7426: Redes definidas por software (SDN): terminología de capas y arquitectura*. https://www.rfc-editor.org/rfc/rfc7426.html

ISO 25000. (2022). *ISO 25010*. https://iso25000.com/index.php/normas-iso-25000/iso-25010

ISO/IEC. (2011). *ISO 25010*.

Kandoi, R., & Antikainen, M. (2015). Denial-of-service attacks in OpenFlow SDN networks. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, 1322–1326. https://doi.org/10.1109/INM.2015.7140489

Kivachuk, V. (2018). *Estudio de la plataforma ONOS para la Gestión de asociaciones de seguridad IPsec en entornos SDN*.

Klöti, R., Kotronis, V., & Smith, P. (2013). OpenFlow: A security analysis. *Proceedings - International Conference on Network Protocols, ICNP*. https://doi.org/10.1109/ICNP.2013.6733671

Kreutz, D., Ramos, F. M. V, & Verissimo, P. (2019). *Towards Secure and Dependable Software-Defined Networks*.

López, J. (2020). *Emulación de una red SD-WAN (Software-Defined Wide Area Network) utilizando tecnología Fortinet y el software GNS3*.

Lores, J. (2011). *Configuración y pruebas de funcionamiento de la interconexión de redes heterogéneas con troncal MPLS*.

Lozada, J. (2022). *IMPLEMENTACIÓN DE UNA RED HÍBRIDA SDN/MPLS PARA LA GESTIÓN DE SERVICIOS DIFERENCIADOS MEDIANTE SIMULACIÓN DE TIEMPO REAL GNS3*. http://repositorio.utn.edu.ec/handle/123456789/13203

Mejia, L. (2018). *ANÁLISIS DE SEGURIDAD EN REDES SDN (REDES DEFINIDAS POR SOFTWARE)*.

MINTEL. (2022). *Agenda de Transformación Digital del Ecuador 2022-2025*. https://aportecivico.gobiernoelectronico.gob.ec/system/documents/attachments/000/000/098/original/ade31653435a0820a7b8b252953dabba6e3ec71b.pdf

Modenesi, M., Fraire, J. A., Pablo, L., & Ventura, G. (2022). *Omnetpy: Integración del Lenguaje Python en el Entorno de Simulación OMNeT++*.

Moscoso, E. (2016). *Desarrollo de una aplicación para la implementación de calidad de servicio por priorización de tráfico sobre una red definida por software (SDN)*.

Nasereddin, H. H. O., & Abdelkarim, A. A. (2011). *INTRUSION PREVENTION SYSTEM*. www.ijar.lit.az

Okokpujie, K., Shobayo, O., Noma-Osaghae, E., Okokpujie, I. P., & Okoyeigbo, O. (2018). Performance of MPLS-based virtual private networks and classic virtual private networks using advanced metrics. *Telkomnika (Telecommunication Computing Electronics and Control)*, *16*(5). https://doi.org/10.12928/TELKOMNIKA.v16i5.7326

OMNeT. (2021). *OMNeT++ Simulation Manual*.

ONF. (2013). *OpenFlow Switch Specification*. http://www.opennetworking.org

ONF. (2016). *Security Foundation Requirements for SDN Controllers*. www.opennetworking.org

ONOS. (2024). *The ONOS Web GUI - ONOS - Wiki*. https://wiki.onosproject.org/display/ONOS/The+ONOS+Web+GUI

Open vSwitch. (2024). *Open vSwitch Manual*. https://www.openvswitch.org/support/dist-docs/ovs-vswitchd.conf.db.5.html

Prokopets, M. (2023). *Intrusion Detection Systems (IDS) Types: The Complete Guide*. https://nira.com/intrusion-detection-systems-ids-types/

Ramachandran, S., & Shanmugam, V. (2017). Impact of DoS Attack in Software Defined Network for Virtual Network. *Wireless Personal Communications*, *94*(4), 2189–2202. https://doi.org/10.1007/S11277-016-3370-1/METRICS

Ruipérez, J. (2021). *Seguridad en Redes definidas por software (SDN)*. www.etsit.upv.es

Santos Kumar, B., Sekhara, T. C., Raju, P., Ratnakar, M., Dawood Baba, S., & Sudhakar, N. (2013). *Intrusion Detection System-Types and Prevention*. www.ijcsit.com

Sharma, K. (2023). *Intrusion Detection System (IDS): Types, Techniques, and Applications*. https://www.knowledgehut.com/blog/security/intrusion-detection-system#frequently-asked-questions

Shirokova, A. (2019). *Security of software-defined networks: Denial-of-service attack detection and mitigation*.

Shu, Z., Wan, J., Li, D., Lin, J., Vasilakos, A. V., & Imran, M. (2016). Security in Software-Defined Networking: Threats and Countermeasures. *Mobile Networks and Applications*, *21*(5), 764–776. https://doi.org/10.1007/S11036-016-0676-X/METRICS

Tapia, J. C. (2022). *Testbed para el estudio de la tecnología de redes definidas por software para la carrera de Telecomunicaciones de la Universidad Técnica del Norte*.

Ujcich, B., Sanders, W. H., Bates, A., Gunter, C. A., & Iyer, R. K. (2020). *SECURING THE SOFTWARE-DEFINED NETWORKING CONTROL PLANE BY USING CONTROL AND DATA DEPENDENCY TECHNIQUES*.

Yanko, I., Marín, A., Félix, C. I., & Paliza, Á. (2016). Seguridad En Redes Futuras . Ims and Sdn Collaboration . an Architecture To Mitigate Security Problems in Future Networks . *Soil& Tillage Research Soil & Tillage ResearchSoil & Tillage Research*, *41*(41).

Yue, M., Wang, H., Liu, L., & Wu, Z. (2020). Detecting DoS Attacks Based on Multi-Features in SDN. *IEEE Access*, *8*, 104688–104700. https://doi.org/10.1109/ACCESS.2020.2999668

Zheng, G., Xu, X., & Yan, J. (2020). SD-CRF: A DoS Attack Detection Method for SDN. *International Conference on Communication Technology Proceedings, ICCT*, *2020-October*, 1116–1120. https://doi.org/10.1109/ICCT50939.2020.9295801

Anexos

Anexo 1. Selección de herramientas de simulación en base a la ISO/IEC 25010

La evaluación de las plataformas de diseño de redes se basa en una revisión de las funciones y capacidades que presenta cada software. El criterio de ponderación prioriza las características de eficiencia de desempeño, compatibilidad y capacidad de interacción, las cuales están definidas en la norma ISO/IEC 25010. Estas características se consideran como elementos clave para establecer los requisitos de uso de la herramienta de simulación de redes, con el fin de realizar una evaluación y selección posterior. De esta manera, se garantiza que el software seleccionado cumpla con los requerimientos establecidos para el desarrollo del presente proyecto.

La (ISO/IEC, 2011) establece la definición las características seleccionadas y cada uno de sus subcomponentes, siendo estos los siguientes:

a) Eficiencia de desempeño

Refleja el rendimiento de un producto de software al llevar a cabo sus funciones dentro de los límites de rendimiento establecidos, mediante el uso eficiente de recursos como la CPU, la memoria y el almacenamiento (ISO/IEC, 2011). A su vez, esta característica se divide en las siguientes subcaracterísticas, las cuales se emplearán como criterios de evaluación en el presente proyecto:

- Uso de recursos: medida en la cual la cantidad y variedad de recursos empleados por el producto al ejecutar sus funciones bajo ciertas condiciones no supera los límites establecidos.
- Capacidad: nivel en que el producto satisface los requisitos relacionados con los límites máximos para un determinado parámetro, como la cantidad

de ítems almacenados, usuarios concurrentes, ancho de banda de comunicaciones,

b) Compatibilidad

Habilidad de un producto para compartir información con otros productos y/o ejecutar sus funciones requeridas cuando comparten un entorno y recursos comunes (ISO/IEC, 2011). Esta característica se desglosa además en las siguientes subcaracterísticas:

- Coexistencia: capacidad del producto para operar de manera conjunta con otro software independiente en un entorno compartido, utilizando recursos comunes sin afectar su funcionamiento.
- Interoperabilidad: habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información compartida,

C) Capacidad de interacción

Habilidad del software para permitir que el usuario interactúe a través de su interfaz (ISO/IEC, 2011); de modo que las subcaracterísticas tomadas en cuenta que se adaptan al proceso de selección de una plataforma de simulación de redes son:

- Operabilidad: capacidad del producto de software que permite al usuario operar y controlar las funciones que realiza con facilidad.

El proceso de evaluación en un inicio requiere fijar distintas abreviaturas, debido a que cada una de ellas definirá su asociación a cada métrica establecida según las características seleccionadas de la norma. Este proceso es realizado con el objetivo de que se permita una mejor comprensión de la información presentada en siguientes secciones de este apartado, de modo que la **Tabla *56*** presenta la nomenclatura utilizada

por las métricas en base a las características de eficiencia de desempeño, compatibilidad y capacidad de interacción.

Tabla 56
Abreviaturas de las métricas utilizadas

Descripción	Abreviatura
Métrica de requerimientos de eficiencia de desempeño	MtRED
Métrica de requerimientos de compatibilidad	MtRC
Métrica de requerimientos de capacidad de interacción	MtRCI

a) Métricas de requerimientos de eficiencia de desempeño

Las métricas de requerimientos de eficiencia de desempeño se centran en la evaluación del rendimiento de las plataformas de diseño de redes. Esta evaluación implica identificar cómo los recursos informáticos del hardware que alojan a las plataformas ejecutan sus funciones en condiciones normales, en relación con los requisitos mínimos de uso de CPU, RAM y almacenamiento. Además, estas métricas abarcan la capacidad de acceso a los recursos y la presentación de información sobre el consumo del sistema por parte del software. Todos estos aspectos están detallados en las métricas de requerimientos recopiladas en la **Tabla *57***.

Tabla 57
Requerimientos de eficiencia de desempeño

Abreviatura	Descripción del requerimiento	Prioridad		
		Alta	Media	Baja
Métricas de utilización de recursos				
MtRED1	La plataforma de simulación debe utilizar la CPU de manera eficiente, asegurando que no exceda el límite especificado de uso de recursos de la CPU bajo condiciones normales de funcionamiento.	X		
MtRED2	El software de simulación debe gestionar eficientemente la memoria RAM, asegurando que no utilice más memoria de la especificada.	X		

MtRED3	La herramienta de simulación debe utilizar eficientemente el almacenamiento disponible.		X	
Métricas de capacidad				
MtRED4	El software debe ser capaz de admitir el acceso a múltiples usuarios, asegurando que puedan acceder a los recursos de simulación que posee.		X	
MtRED5	Función de procesar una cantidad específica de datos o tareas, asegurándose de presentar la información del uso de los recursos del sistema al usuario.	X		
MtRED6	Permite la continuidad de sus funciones incluso cuando el sistema se aproxima a los límites máximos de utilización de recursos		X	

b) Métricas de requerimientos de compatibilidad

Las métricas de compatibilidad se centran en garantizar que la plataforma pueda interactuar de manera efectiva con otras herramientas, como software de virtualización, sistemas operativos de fabricantes de dispositivos de red y herramientas de análisis de tráfico. Por otro lado, las métricas de interoperabilidad se enfocan en asegurar que la plataforma pueda integrarse sin problemas con diferentes entornos de red y softwares, permitiendo compartir recursos del sistema y simular comunicaciones mediante diversas tecnologías de red de datos. Estas métricas se presentan en la **Tabla *58*** y son esenciales para garantizar la flexibilidad, compatibilidad y funcionalidad de la plataforma seleccionada en el contexto del proyecto.

Tabla 58
Requerimientos de compatibilidad

Abreviatura	**Descripción del requerimiento**	**Prioridad**		
		Alta	**Media**	**Baja**
Métricas de coexistencia				
MtRC1	La plataforma de simulación posee funciones que permiten implementar máquinas virtuales mediante softwares de virtualización y contendores Docker.	X		

MtRC2	La herramienta habilita la integración de los sistemas operativos de dispositivos de red tradicionales y redes definidas por software.	X		
MtRC3	El software permite que herramientas de análisis de tráfico que recopilen información de red las simulaciones.		X	
Métricas de interoperabilidad				
MtRC4	La herramienta de simulación permite la convergencia entre los softwares de virtualización con sus funciones propias para simular la comunicación de mediante diferentes tecnologías de red.	X		
MtRC5	El software posibilita que el uso de recursos del sistema sea compartido para su operación.			X

c) Métricas de requerimientos de capacidad de interacción

Las métricas de capacidad de interacción se centran en garantizar que la plataforma ofrezca una interfaz gráfica intuitiva para facilitar la implementación y conexión de los elementos de red, así como una interfaz de línea de comandos para la configuración de dispositivos. Estas métricas definidas en la **Tabla *59*** se consideran esenciales para asegurar la usabilidad y la eficacia del software de simulación de redes.

Tabla 59
Requerimientos de capacidad de interacción

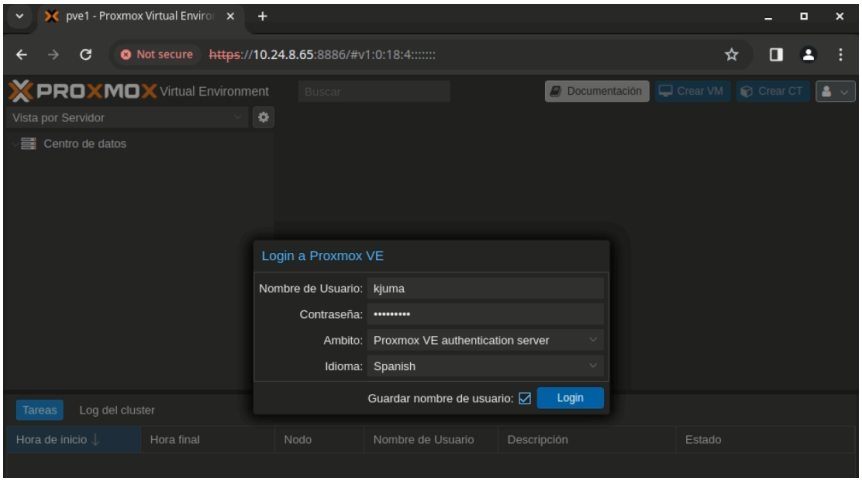
Abreviatura	**Descripción del requerimiento**	**Prioridad**		
		Alta	**Media**	**Baja**
MtRCI1	El software posee una interfaz gráfica que facilita la implementación y conexión de los elementos de red.	X		
MtRCI2	La herramienta de simulación posee una interfaz de línea de comandos que permita la configuración de los dispositivos de red.			X
MtRCI3	La plataforma de simulación permite la captura de tráfico de red mediante el uso de su interfaz gráfica		X	

Anexo 2. Máquina virtual alojada en el servidor Proxmox

Uno de los procesos iniciales correspondientes al desarrollo del testbed es poseer un host virtualizado que aloje los dos escenarios en el servidor Proxmox de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte. El acceso a este host solo puede ser realizado desde la red (Eduroam) de la UTN mediante la dirección url de Proxmox `https://10.24.8.65:8886/`.

Al acceder, el servidor requiere la autentificación de usuario y contraseña asignados por el administrador del servidor, que son `Usuario: kjuma`, `Contraseña: kjuma123#` y `Ámbito: Proxmox VE authentication server`; como puede ser identificar en la **Figura *92***.

Figura 92
Inicio de sesión con credenciales asignadas en el servidor Proxmox

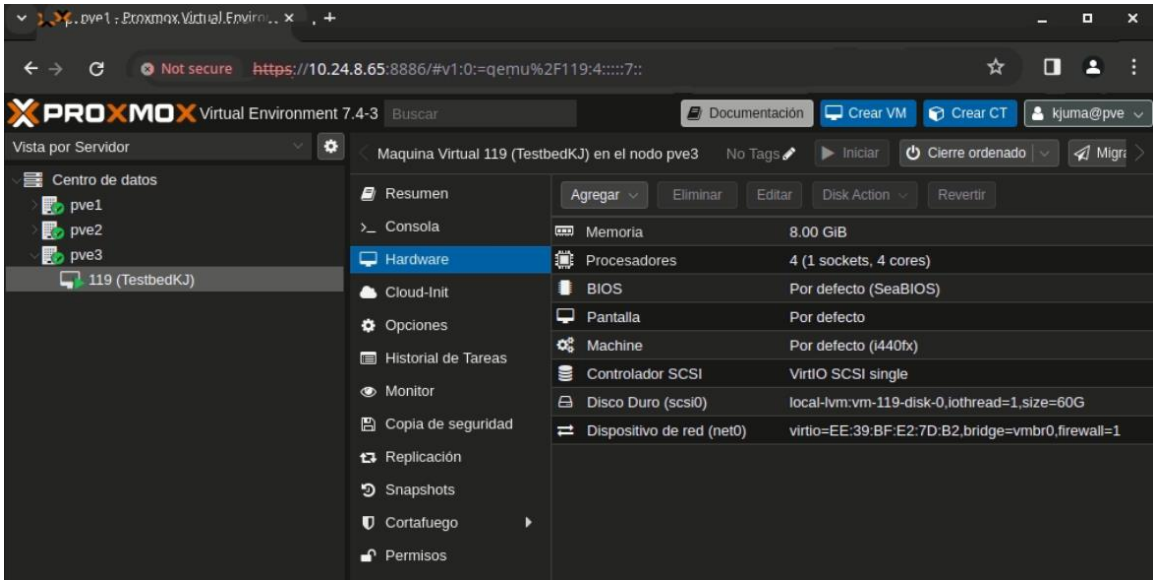


Una vez que el servidor ha autenticado de forma exitosa el inicio de sesión, la máquina virtual asignada se encuentra ubicado en el Tercer Entorno Virtual Proxmox (pve3) con el identificador 119, correspondiente al host asignado para la presente investigación con las características de hardware y sistema operativo identificadas en la **Tabla *60*** y evidenciados en la **Figura *93***.

Tabla 60
Características del host anfitrión

Características	Descripción
Procesador	1 socket (4 núcleos)
RAM	8 gigabytes
Almacenamiento	60 gigabytes
Sistema Operativo	Debian 12

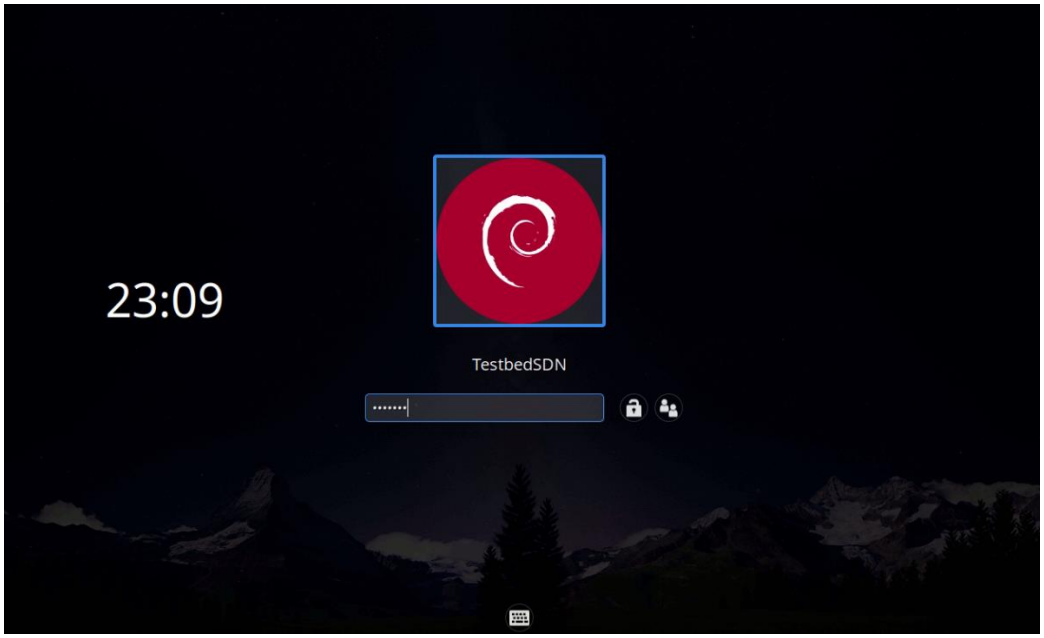
Figura 93
Características de la máquina virtual en el servidor Proxmox de la UTN



Finalmente, el usuario puede acceder a la consola de la máquina virtual que un inicio la ventana de inicio de sesión de Debian 12 requerirá las credenciales para acceder a la interfaz y las funciones del sistema. Las credenciales que posee la máquina son `Usuario: testbedsdn` y `Contraseña: t3stb3d`, como puede ser mostrado en la **Figura *94***.

Figura 94

Acceso a la interfaz de Debian 12



Anexo 3. Requisitos e instalación de GNS3 en Debian 12

La implementación del software requiere un proceso de identificación de los requerimientos de hardware y software que los hosts anfitriones deberán poseer con el propósito de obtener un correcto funcionamiento de la plataforma, de modo que (GNS3, 2023) indica en su página web la lista de especificaciones que permiten su funcionamiento en sistemas Linux o Windows. Basándose en estos requisitos, la Tabla 61 enumera los elementos mínimos y las especificaciones técnicas que el host anfitrión debe satisfacer para garantizar el funcionamiento adecuado de la plataforma.

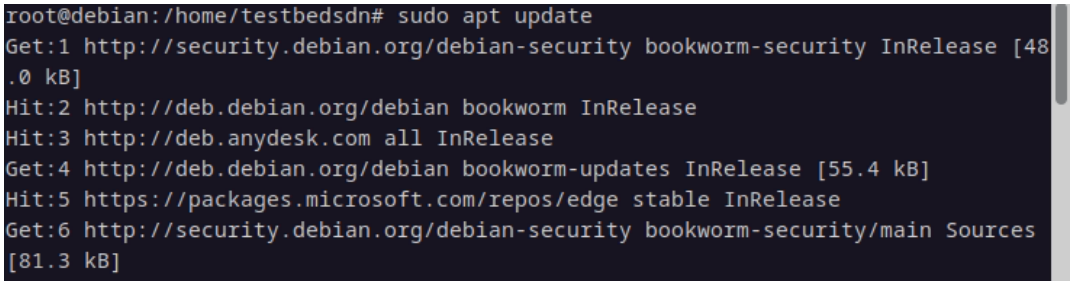
Tabla 61
Requisitos de instalación de GNS3

Requerimiento	Prioridad	Descripción
Sistema Operativo	Baja	Plataformas Windows y Linux (Debian/Ubuntu)
Procesador	Alta	Mínimo 4 núcleos lógicos: Serie AMD-V/RVI o Intel VT-X/EPT
Virtualización	Media	Requiere virtualización
Memoria RAM	Alta	Mínimo 8 GB de RAM
Almacenamiento	Alta	Unidad de estado sólido (SDD) de 35 GB de espacio disponible

Nota. Adaptado de (GNS3, 2023)

La instalación del software de simulación de redes GNS3 en el sistema operativo Debian 12 inicia actualizando todos los paquetes de los repositorios de Debian mediante el comando `sudo apt update`. Esto es realizado con el objetivo de poseer una lista actualizada de las más recientes versiones de los paquetes que necesita el software para su instalación y este proceso puede ser evidenciado en la **Figura *95***.

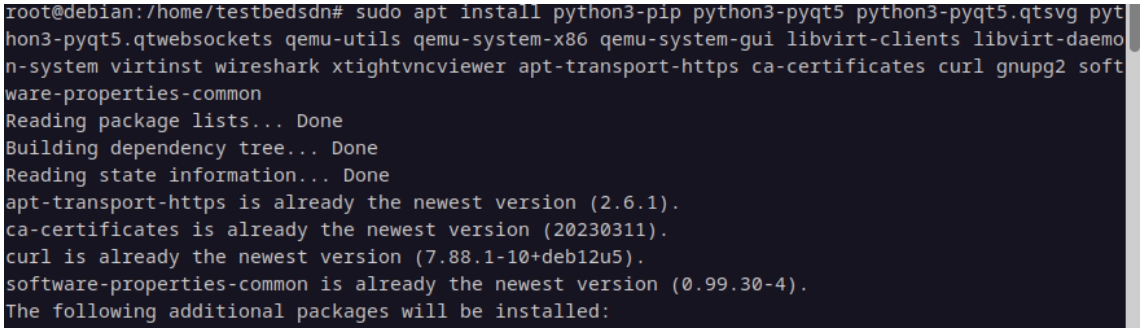
Figura 95
Actualización de repositorios de Debian 12



Una vez que los paquetes han sido actualizados, el siguiente paso consiste en instalar las diversas herramientas y paquetes para desarrollo y virtualización, como el intérprete de Python, PyQt5 para interfaces gráficas, utilidades y componentes esenciales para la virtualización con QEMU y libvirt, herramientas de red como Wireshark, un visor VNC, y paquetes adicionales necesarios para la instalación de software y configuraciones como se evidencia en la **Figura *96***. Este conjunto de comandos proporciona un entorno básico y necesario para tareas de desarrollo y virtualización en sistemas Debian que utiliza GNS3.

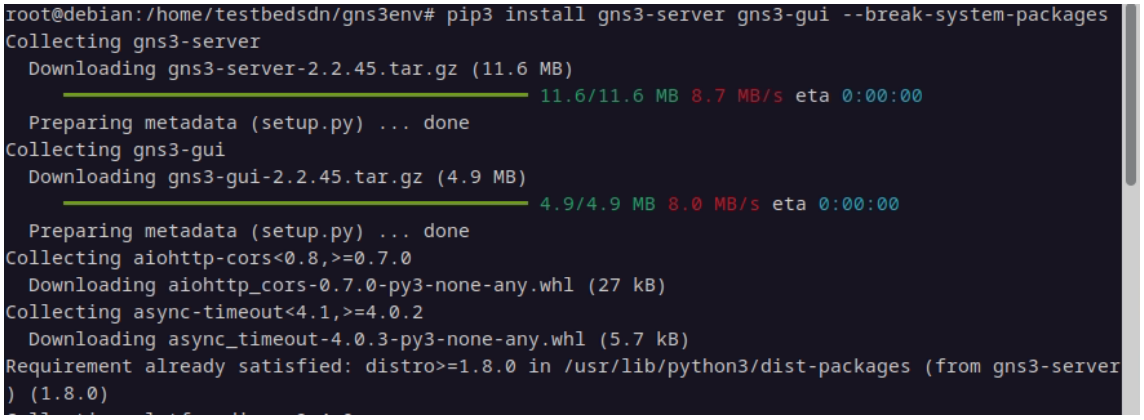
La instalación de todas las herramientas mencionadas anteriormente que utiliza GNS3 para su instalación son: `sudo apt install python3-pip python3-pyqt5 python3-pyqt5.qtsvg python3-pyqt5.qtwebsockets qemu-utils qemu-system-x86 qemu-system-gui libvirt-clients libvirt-daemon-system virtinst wireshark xtightvncviewer apt-transport-https ca-certificates curl gnupg2 software-properties-common`

Figura 96
Instalación de herramientas para GNS3



Establecidas las herramientas previas que necesita GNS3 para su funcionamiento en sistemas Debian, el siguiente paso consiste en instalar y configurar GNS3 correspondiente a sus componentes de interfaz de línea de comandos, interfaz gráfica y servidor remoto mediante el comando `pip3 install gns3-server gns3-gui --break-system-packages,` que son evidenciados en la **Figura *97***.

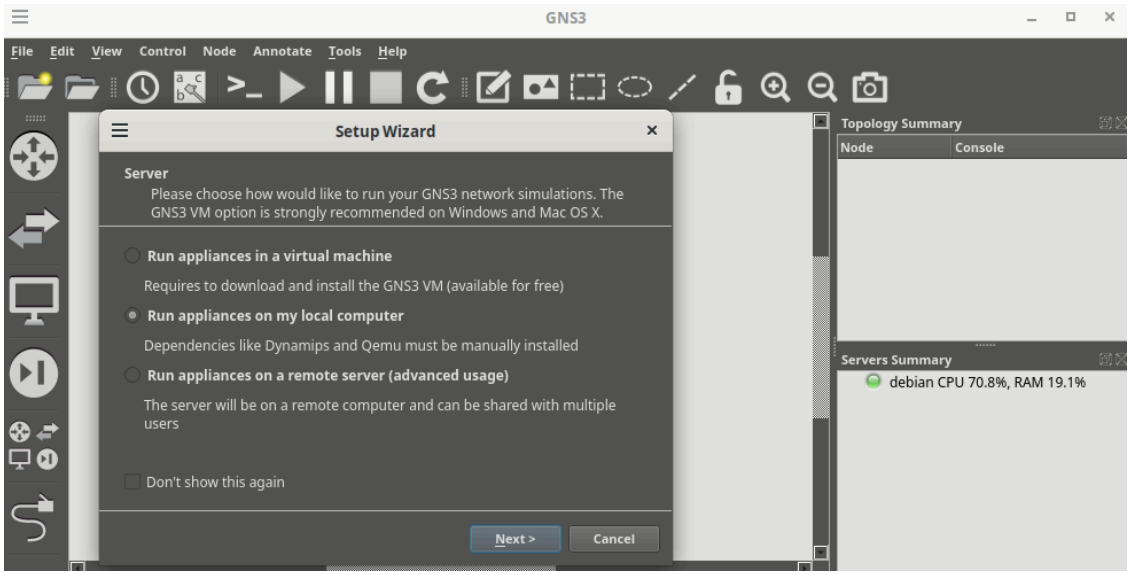
Figura 97
Instalación de GNS3 y sus complementos



Al finalizar la instalación mediante la línea de comandos, el usuario podrá acceder a la interfaz gráfica de GNS3 con el objetivo de realizar las últimas configuraciones relacionadas a que todos los procesos se ejecuten dentro del sistema operativo que aloja este software y configuraciones de conectividad al servidor web que en este caso se mantendrán predeterminadas como se muestra en la **Figura *97***.

Figura 98

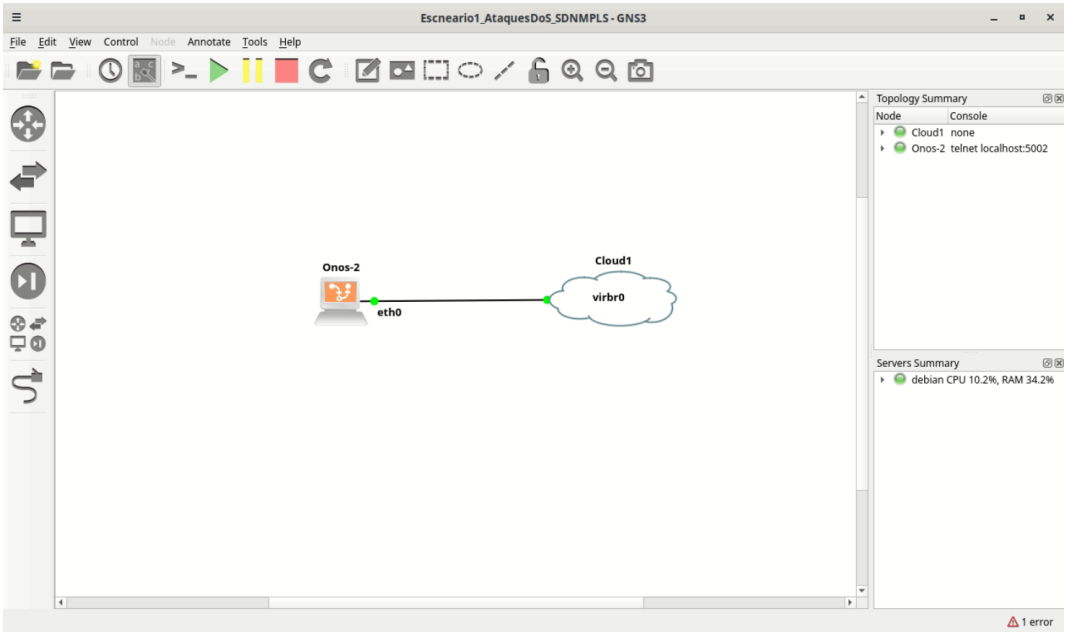
Selección de Debian como host que aloja funciones de GNS3



Finalmente, con las configuraciones finales mencionadas previamente el usuario puede acceder a la interfaz gráfica para desarrollar los escenarios planteados.

Figura 99

Funcionamiento de la GUI de GNS3



Anexo 4. Protocolo de enrutamiento utilizado en la red

La elección y uso de un protocolo de enrutamiento es fundamental para el funcionamiento de la red híbrida SDN/MPLS ya que son requeridas las funciones como cálculo de la mejor ruta hacia las diferentes redes que conforman la topología, intercambio de información de enrutamiento entre los enrutadores referente a las subredes alcanzables en conjunto con los caminos preferidos hacia las mismas y toma de decisiones sobre qué ruta seguir para llegar a una red de destino específica.

Las funcionalidades que poseen los protocolos de enrutamiento permiten a tecnologías como MPLS funcionar en conjunto con protocolos de distribución de etiquetas para asignar y distribuirlas a lo largo de la ruta definida por el protocolo de enrutamiento creando los label paths utilizados en esta tecnología para la conmutación de paquetes. En el caso de SDN los protocolos de enrutamiento son fundamentales para que el controlador SDN pueda tomar decisiones informadas sobre cómo dirigir el tráfico de la red, ajustando dinámicamente las rutas según las necesidades cambiantes del tráfico y las políticas de red implementadas.

Por ende, es necesario una comparativa detallada entre los diferentes protocolos de enrutamiento de gateway interior (IGP) como lo son OSPF, IS-IS y RIPv2 recopilando sus características principales en la **Tabla *62*** con el propósito de elegir el protocolo más adecuado para la infraestructura de la red híbrida SDN/MPLS.

Tabla 62
Comparativa de protocolos de enrutamiento OSPF, IS-IS y RIPv2

Característica	OSPF	IS-IS	RIPv2
Tipo de protocolo	Estado del enlace	Estado del enlace	Vector distancia
Métrica	Costo	Costo	Numero de saltos
Escalabilidad	Utiliza áreas jerárquicas en los routers	Utiliza niveles jerárquicos en los routers	Limitado al número de saltos
Actualización de rutas	Eventual ante cambios en la topología	Eventual ante cambios en la topología	Periódica cada 30 segundos
Límite de saltos permitidos	Sin límite	Sin límite	15 saltos
Distancia administrativa	110	115	120
Construcción de la tabla de enrutamiento	Algoritmo SPF sobre LSDB utilizando LSAs	Algoritmo SPF sobre LSDB utilizando LSPs	Actualizaciones de vecinos
Seguridad	Autentificación por MD5 – SHA	Autentificación por MD5	No

Nota. Adaptado de (Cisco, 2023d, 2023b, 2023c, 2023a).

La Tabla 62 a su vez evidencia las ventajas que presenta OSPF en comparación a IS-IS y RIPv2, lo que permite justificarla elección de OSPF como protocolo de enrutamiento para el diseño de una red híbrida SDN/MPLS se basa en las siguientes ventajas que presentada OSPF en comparación a los protocolos de enrutamiento mencionados anteriormente:

- El protocolo OSPF permite una alta escalabilidad y soporte a grandes redes, la cual es una característica requerida en redes SDN/MPLS que interconectan gran cantidad de dispositivos tradicionales y conmutadores SDN.
- Posee una rápida convergencia al utilizar el algoritmo del camino más corto conocido como Shortest Path First (SPF) que se fundamenta en el algoritmo de Dijkstra para calcular las rutas más cortas desde el router de origen a todos los demás routers y actualizar dichas rutas si se presentan cambios de topología

mediante actualizaciones de los LSAs que maneja cada router, de modo que esta característica es esencial en redes que requieren conectividad como lo son las SDN/MPLS.

- El protocolo de enrutamiento OSPF presenta una distancia administrativa de 110, de modo que es un valor numérico que se utiliza para seleccionar la mejor ruta dando confianza en el enrutamiento en comparación a IS-IS con una distancia de 115 o RIPv2 que posee el valor de 120.

El protocolo OSPF basa su funcionamiento en el algoritmo de la ruta más corta (SPF) intercambiando mensajes del estado del enlace, de modo que los routers que manejan el protocolo OSPF necesitan establecer una relación de vecindad antes de intercambiar actualizaciones de enrutamiento. Una vez establecida la vecindad, los routers intercambian información de estado del enlace (LSA) para construir y mantener sus bases de datos de enlace de estado (LSDB), detallando el estado de los enlaces en la red. Utilizando el algoritmo SPF, cada router calcula las mejores rutas hacia todos los destinos, generando su tabla de enrutamiento asegurando que los routers posean la información precisa sobre la topología de la red, facilitando la elección de la ruta del tráfico hacia su destino final.

Las configuraciones relacionadas con el protocolo OSPF inician estableciendo los identificadores de cada router OSPF en la topología también conocido como el Router ID que es considerada como una herramienta utilizada para definir el Router Designado (DR) y el Router Designado de Respaldo (BDR) en un área OSPF que son necesarios para la sincronización de bases de datos y la elección del DR. Las configuraciones de cada enrutador se centran en establecer el número de proceso, declarar la dirección IP de cada red en conjunto con la máscara de wildcard, establecer el área OSPF a la que pertenece y finalmente el establecimiento del Router ID toma la dirección IP más alta que

corresponde a las interfaces de Loopback que posee cada router, por ende, la **Tabla *63*** especifica cada uno de los Router ID configurados en base a la dirección Loopback que se le asigna a cadanrutador.

Tabla 63
Interfaces Loopback y Router ID de los enrutadores

Enrutador	Loopback (IPv4)	Máscara de red	Router ID
R1	1.1.1.1	255.255.255.255	1.1.1.1
R2	2.2.2.2	255.255.255.255	2.2.2.2
R3	3.3.3.3	255.255.255.255	3.3.3.3
R4	4.4.4.4	255.255.255.255	4.4.4.4

Anexo 5. Instalación del controlador SDN RYU

La instalación del controlador SDN Ryu requiere configurar un entorno de desarrollo de aplicaciones que requieren de Python 3.9 y bibliotecas específicas, que en este caso son las que necesita el controlador Ryu para poder operar. Cabe destacar que la instalación del controlador es realizada en un entorno Ubuntu 22.04, de modo que los comandos utilizados para la instalación del controlador son para este sistema operativo y se presentan en la Tabla 64, la cual especifica el comando utilizado y su función en la instalación de Ryu.

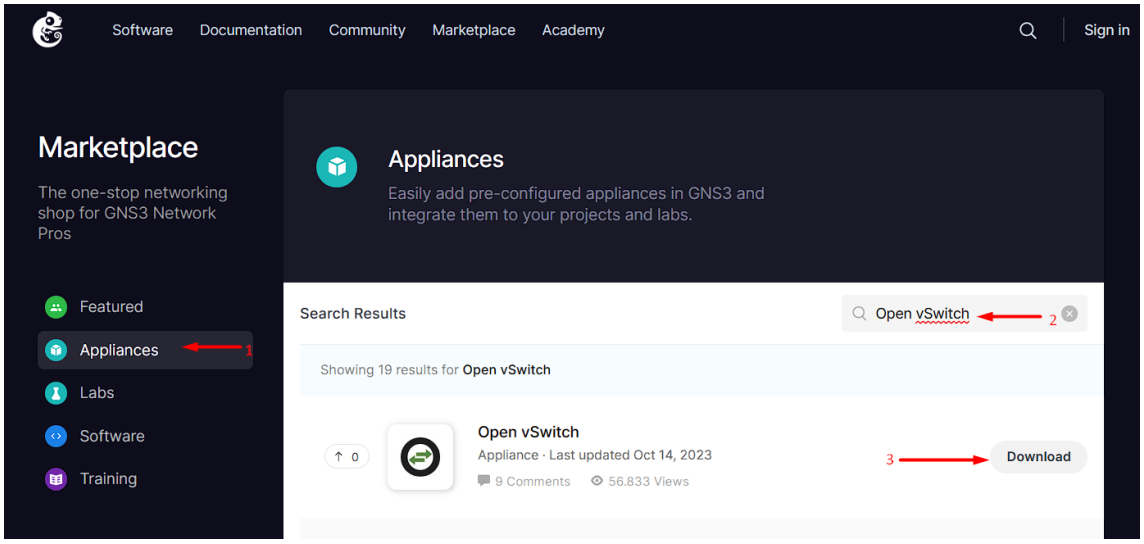
Tabla 64
Comandos de instalación del controlador Ryu en Ubuntu 22.04

Comando	Función
`sudo apt update`	Actualiza la lista de paquetes disponibles en los repositorios configurados en el sistema, lo que conlleva a que se trabajará con las versiones más recientes de los paquetes.
`sudo apt install software-properties-common`	Instala un paquete para gestionar repositorios PPA, permitiendo la instalación de software adicional no incluido en los repositorios estándar.
`sudo add-apt-repository ppa:deadsnakes/ppa`	Agrega el repositorio "deadsnakes", que incluye versiones más recientes de Python que no están disponibles en los repositorios oficiales de Ubuntu.
`sudo apt update`	Actualiza nuevamente la lista de paquetes para incluir el nuevo repositorio que fue añadido.
`sudo apt install python3.9 python3.9-dev python3.9-distutils`	Instala Python 3.9 junto con las herramientas: python3.9-dev encargado de archivos de desarrollo y python3.9-distutils que corresponde a utilidades para manejar módulos y extensiones.
`curl https://bootstrap.pypa.io/get-pip.py -o /home/keneth/get-pip.py`	Descarga el script oficial para instalar pip, la herramienta de gestión de paquetes de Python, en una ruta especificada del sistema.
`sudo python3.9 get-pip.py`	Ejecuta el script para instalar pip en el sistema utilizando Python 3.9.
`sudo pip3 install ryu eventlet==0.30.2`	Instala el controlador Ryu y la biblioteca eventlet en su versión 0.30.2 mediante pip, siendo este proceso necesario para garantizar la compatibilidad con el contolador Ryu.

Anexo 6. Instalación de Open vSwitch

La instalación de los OpenVswitch inicia con la descarga desde el Marketplace oficial de GNS3 que posee la url `https://gns3.com/marketplace/appliances`. Una vez el usuario se encuentra en la tienda de aplicaciones de GNS3 deberá realizar una serie de pasos para obtener el contenedor de los switches SDN. La **¡Error! No se encuentra el origen de la referencia.** muestra el proceso a seguir, en un inicio el usuario seleccionará el apartado de Appliances de la marketplace, posteriormente en la barra de búsqueda será necesario ingresar el nombre de OpenVswitch y finalmente descargar el container requerido.

Figura 100
Descarga del container de OpenVswitch



Una vez que el contenedor se ha cargado y GNS3 esté inicado, el usuario deberá seleccionar la opción ***Import appliance*** en la sección ***File***, siguiendo la indicación de la **¡Error! No se encuentra el origen de la referencia.**. Al acceder a esta opción, se elige l a aplicación previamente descargada, como se ilustra en la **Figura *101*** y **Figura *102***, para dar continuidad al proceso de instalación.

Figura 101

Sección que permite la importación de containers a GNS3

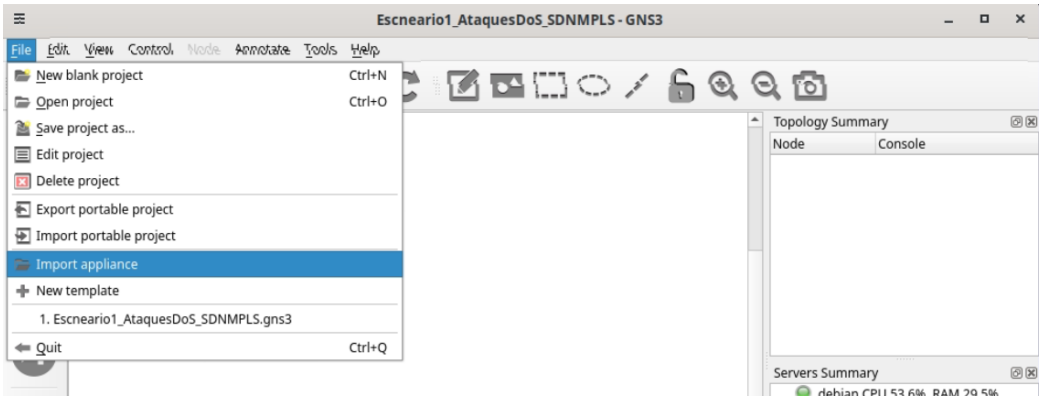
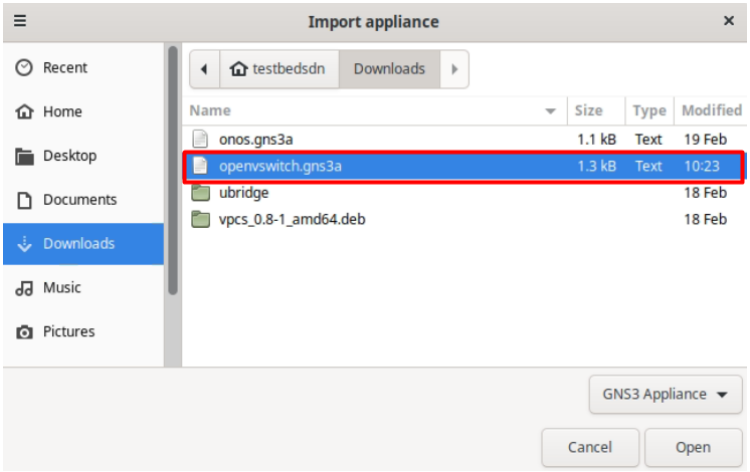


Figura 102

Selección del container de Open Vswitch

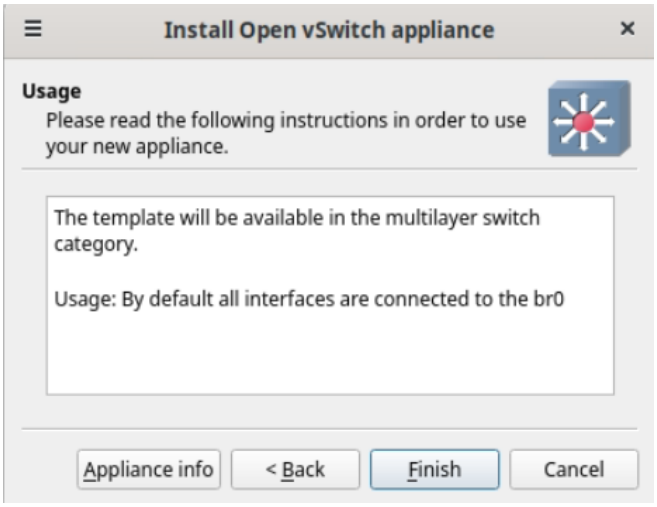


Siguiendo con el proceso de importación del container, en la **Figura *103*** se especifica el servidor que se debe seleccionar, recomendándose en este caso el servidor propio de GNS3.

Figura 103
Selección del servidor de GNS3 para alojar el container de Open Vswitch

Completado el proceso de selección del servidor, los Open Vswitch serán añadidos exitosamente como se muestra en la **Figura *104*** y permite que este elemento sea añadido a la lista de componentes de red de GNS3 que pueden ser usados.

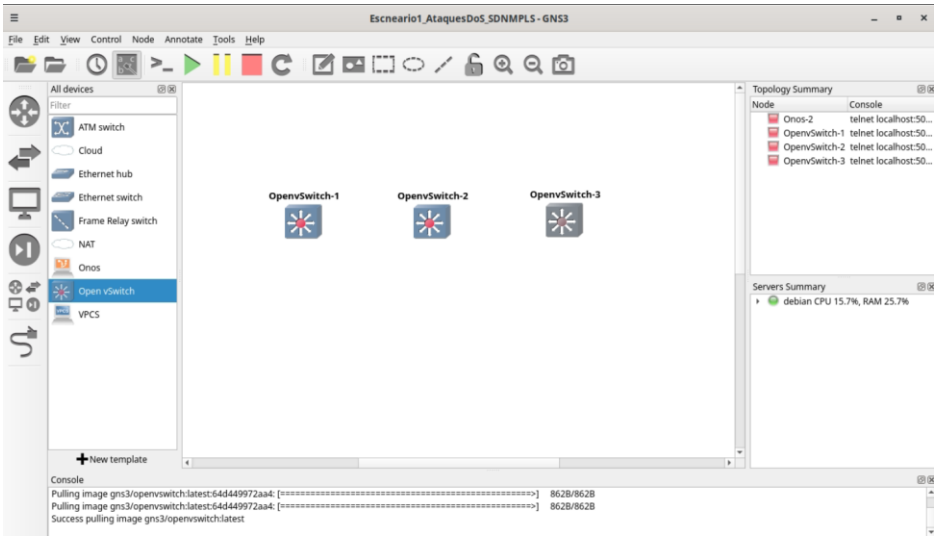
Figura 104
Container de Open Vswitch añadido exitosamente



Finalmente, como comprobación de que el container que aloja a los Open Vswitch no presenta ningún tipo de error este puede ser seleccionado en un espacio de trabajo de GNS3 y comenzar con la descarga para su posterior funcionamiento, como se muestra en la **Figura *105***.

Figura 105

Conmutadores SDN disponibles en los elementos de red de GNS3



Anexo 7. Repositorio de Github del proyecto

https://github.com/KenethJuma1999/testbed_dos_sdn.git