PROYECTO DE TITULACIÓN

Carrera de Ingeniería en Tecnologías de la Información

TEMA: Diseño de una Red Definida por Software de alta velocidad utilizando la tecnología Segment Routing

AUTOR: Benalcazar Vallejo, Erika Alexandra

TUTOR: Ing. Núñez Agurto, Alberto Daniel, Mgtr.

Santo Domingo, 01 de Marzo del 2024

Reporte de verificación de contenido

Plagiarism and AI Content Detection Report

SDN_SR_Tesis_EB.pdf

Scan details

Scan time:	Total Pages:	Total Words:
March 5th, 2024 at 12:5 UTC	51	12716

Plagiarism Detection

0.8%

Types of plagiarism		Words
Identical	0.4%	49
Minor Changes	0%	0
Paraphrased	0.3%	40
Omitted Words	14.9%	1897

AI Content Detection

0%

Text coverage		Words
AI text	0%	0
Human text	100%	10819

Learn more

Alerts: (1)

Cross Language: Same Document Language

2/5 Severity

Submitted language and cross-language text are the same language. No credits were used.

Plagiarism Results: (4)

Chafloque_mj.pdf.txt;jsessionid=F80767499F529787877FA42583BCAF5B?s... 0.3%

https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/10017/chafloque_mj.pdf.txt;jsessionid=f80...

Universidad Nacional Mayor de San Marcos Universidad del PerÃo. Decana de AmÃ©rica Facultad de IngenierÃa ElectrÃ3nica y ElÃ©ctrica Escu...

Cisco IOS XRv Router Installation and Configuration Guide - Hypervisor R... 0.2%

https://www.cisco.com/en/us/docs/ios_xr_sw/ios_xrv/install_config/b_xrvr_432_chapter_010.html

Cisco IOS XRv Router Installation and Configuration Guide Hypervi...

CasilimasFajardoCarlosAlexis2022.pdf?sequence=2&isAllowed=y 0.2%

https://repository.udistrital.edu.co/bitstream/handle/11349/29727/casilimasfajardocarlosalexis2022.pdf?seq...

PAPELES Y CORRUGADOS ANDINA

1 Arquitectura y funcionamiento de redes definidas por software (SDN) Universidad Distrital Francisco Jose de Caldas. Especialización Te...

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Certificación

Certifico que el trabajo de integración curricular: "**Diseño de una Red Definida por Software de alta velocidad utilizando la tecnología Segment Routing**" fue realizado por la señorita **Benalcazar Vallejo, Erika Alexandra,** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Santo Domingo, 1 de marzo del 2024

Firma:

Firmado electrónicamente por:
ALBERTO DANIEL
NUNEZ AGURTO

.....

Núñez Agurto, Alberto Daniel

C. C: 1716572548

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Responsabilidad de Autoría

Yo, **Benalcazar Vallejo, Erika Alexandra**, con cédula de ciudadanía n° 0401967013, declaro/declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Diseño de una Red Definida por Software de alta velocidad utilizando la tecnología Segment Routing** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Santo Domingo, 1 de marzo del 2024

Firma:

.....

Benalcazar Vallejo, Erika Alexandra

C.C.: 0401967013

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Autorización de Publicación

Yo **Benalcazar Vallejo, Erika Alexandra**, con cédula de ciudadanía n° 0401967013, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Diseño de una Red Definida por Software de alta velocidad utilizando la tecnología Segment Routing** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Santo Domingo, 1 de marzo del 2024

Firma:

.....

Benalcazar Vallejo, Erika Alexandra

C.C.: 0401967013

DEDICATORIA

Dedico este trabajo a mis padres, que siempre me apoyaron desde el día uno al iniciar mi etapa académica; a mis padrinos, que me brindaron apoyo incondicional y siempre me motivaron a salir adelante; a mis hermanos, quienes con sus enseñanzas y regaños fueron una parte importante para mi desarrollo personal y académico. Y, sobre todo, a mí mismo, ya que logré culminar con éxito mis estudios universitarios, a pesar de muchas circunstancias.

Benalcazar Vallejo, Erika Alexandra

AGRADECIMIENTO

Agradezco a Dios por darme sabiduría y perseverancia para afrontar desafíos y poder culminar con esta etapa.

A mis padres, padrinos y hermanos, quienes me brindaron una palabra de aliento para seguir, los mismos que juegan un papel importante en mi vida. Sé que sin ellos no hubiera sido posible alcanzar este logro.

A mi tutor de tesis, quien con su paciencia me ayudó a culminar este proceso, y especialmente nunca dejó que me rindiera.

A la universidad y los docentes, quienes me brindaron conocimientos y experiencias importantes para mi vida. También a los amigos y amigas con quienes compartí experiencias únicas en mi vida universitaria.

Gracias a todas las personas que fueron parte de este proceso que hoy se culmina.

Benalcazar Vallejo, Erika Alexandra

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Las telecomunicaciones enfrentan constantemente nuevos desafíos, especialmente en términos de tráfico de datos, ancho de banda y latencia, particularmente en redes a gran escala como las de los proveedores de servicios. El tráfico de datos está en constante evolución debido a la variedad de servicios disponibles, lo que afecta considerablemente a las redes tradicionales. En este contexto, las redes definidas por software han surgido como una solución, ofreciendo flexibilidad y escalabilidad para minimizar la congestión, la pérdida de paquetes y los problemas de encolamiento de tráfico. La monitorización de la red a través de software se presenta como una solución eficiente para gestionar el tráfico de datos, ya que estas redes permiten un control sobre grandes volúmenes de tráfico. Por lo tanto, el propósito de esta investigación busca emular una red definida por software implementando la tecnología de Segment Routing, así permitiendo optimizar el rendimiento eficaz a los proveedores de servicios. Se trabaja en un entorno virtual de simulación GNS3, utilizando dispositivos Cisco IOS XR y el controlador Opendaylight como parte de la infraestructura de una red definida por software y la tecnología de Segment Routing. Para el cumplimiento del mismo, se utiliza la metodología PPDIOO. Como resultado, se logra la integración de dos tecnologías que ofrecen un mejor rendimiento en el manejo del tráfico de datos. A través de esta integración, se instaura un escenario de pruebas para verificar el funcionamiento, rendimiento y características con la herramienta Wireshark.

Palabras clave: Redes Definidas por Software, Segment Routing, NETCONF.

ABSTRACT

Telecommunications are constantly facing new challenges, especially in terms of data traffic, bandwidth and latency, particularly in large-scale networks such as those of service providers. Data traffic is constantly evolving due to the variety of services available, which significantly affects traditional networks. In this context, software-defined networks have emerged as a solution, offering flexibility and scalability to minimize congestion, packet loss and traffic queuing problems. Network monitoring through software is presented as an efficient solution to manage data traffic, as these networks allow control over large volumes of traffic. Therefore, the purpose of this research is to emulate a software-defined network by implementing Segment Routing technology, thus allowing service providers to optimize efficient performance. We work in a virtual simulation environment GNS3, using Cisco IOS XR devices and the Opendaylight controller as part of the infrastructure of a software-defined network and Segment Routing technology. The PPDIOO methodology is used to achieve this goal. As a result, the integration of two technologies that offer better performance in data traffic management is achieved. Through this integration, a test scenario is set up to verify the operation, performance and characteristics with the Wireshark tool.

Keywords: Software-Defined Networking, Segment Routing, NETCONF

I. Introducción y estado del arte

A. *Introducción*

En la actualidad, las redes de comunicación enfrentan desafíos significativos en la gestión eficiente del tráfico, lo que ha generado la necesidad de administrar flujos de datos a velocidades cada vez más altas. Los Proveedores de Servicios de Internet (ISPs) que gestionan redes de alta velocidad a menudo se enfrentan a congestiones y dificultades en la gestión efectiva del tráfico, lo que puede resultar en un rendimiento deficiente para los usuarios finales [1].

A medida que las redes se vuelven más complejas y las aplicaciones más exigentes en términos de ancho de banda y latencia, ha surgido la necesidad de adoptar un enfoque más ágil y centralizado para la administración de redes [2]. La transición de las redes tradicionales basadas en hardware hacia las Redes Definidas por Software (SDN) se ha acelerado para abordar estas demandas. Las SDN abordan problemas críticos en el ámbito de las redes, como la virtualización y la complejidad en la gestión de los centros de datos.

Uno de los problemas centrales identificados es la dificultad para garantizar una gestión eficiente del tráfico en redes de alta velocidad [3]. La velocidad y la constante carga de datos ejercen una presión significativa sobre la infraestructura de red, lo que aumenta la complejidad de la toma de decisiones en términos de enrutamiento y asignación de recursos.

La implementación de las SDN ha demostrado su capacidad para mejorar la flexibilidad y la gestión de las redes de comunicación. Sin embargo, la aplicación de las redes definidad por software, especialmente haciendo uso de tecnologías como Segment Routing (SR), en entornos de alta velocidad presenta desafíos que requieren una atención más detenida [4]. La evaluación proporcionará información crítica sobre cómo el Segment Routing influye en el rendimiento y la eficiencia de la red.

Tradicionalmente, las redes corporativas se han basado en el control centralizado, el enrutamiento y la seguridad. Implementar una red definida por software con Segment Routing puede facilitar la gestión dinámica del tráfico, permitiendo rutas eficientes y adaptativas basadas en la carga de la red [5]. El constante aumento en la demanda de ancho de banda, impulsado por aplicaciones multimedia y servicios de misión crítica, exige soluciones de red que puedan adaptarse y escalar dinámicamente para garantizar un rendimiento óptimo.

Las SDN y el Segment Routing permiten una gestión centralizada y programable de los recursos de red, lo que facilita la adaptación dinámica a cambios en la carga y mejora la escalabilidad. Una SDN en Segment Routing puede adaptarse más fácilmente a los requisitos específicos de tecnologías emergentes.

B. Estado del arte

Hace unos años, las SDN representaban el mayor impacto en el sector de las redes de comunicación, dado que las redes tradicionales se utilizaban para dirigir el tráfico de red a través de routers. Inicialmente, con la llegada de SDN, se implementaron para la gestión del plano de control y el plano de datos para el flujo de tráfico [6].

Este estudio de investigación se enfoca en la clasificación del tráfico en las SDN utilizando técnicas de Aprendizaje Automático, lo cual es relevante para tu tema de tesis sobre el Diseño de una SDN utilizando Segment Routing. Destaca la importancia de la clasificación del tráfico para gestionar eficientemente los recursos de red de acuerdo con los requisitos de calidad de servicio y seguridad. Se menciona que las SDN, con su controlador centralizado, ofrecen una visión global de la red que facilita el análisis del tráfico y proporciona capacidades de programación directa, lo cual se alinea con la idea de diseñar una SDN utilizando Segment Routing. La integración de técnicas de Aprendizaje Automático en las SDN permiten agregar inteligencia a las redes, optimizarlas y mejorar su gestión y mantenimiento, lo cual puede ser beneficioso para este trabajo sobre el diseño de una SDN con Segment Routing. Además, se identifican nuevos desafíos y direcciones futuras de investigación en la clasificación del tráfico en las redes definidas por software, lo cual puede ser relevante para mejorar la eficiencia y la seguridad de la red diseñada [7].

Actualmente, las SDN se enfocan en diversos ámbitos como ejemplo son redes amplias de qué forma se conectan las empresas hacia sus sucursales. Por ello se tiene la implementación de controladores centralizados para la gestión eficaz del tráfico de infraestructuras de red brindando una solución eficiente y flexible [8].

Segment Routing permite el transporte de tráfico a través de rutas específicas. Los principales problemas que aborda en la tecnología MPLS, son las siglas de Multiprotocol Label Switching (conmutación de etiquetas multiprotocolo) son la escalabilidad y la facilidad de ope-

ración. El autor [3], menciona que el Segment Routing es un nuevo modelo de negocio que permite que las aplicaciones dirijan el comportamiento de la red a través de cualquier flujo topológico. Además, su implementación es sencilla y simple de configurar. Para la ingeniería de tráfico, existía el protocolo de reserva de recursos RSVP-TE, el cual requería una etiqueta para cada ruta. Sin embargo, el Segment Routing ofrece una solución sin necesidad de implementar este protocolo. No requiere un protocolo de distribución de etiquetas como LDP; estas etiquetas se distribuyen mediante el protocolo de puerta de enlace interior (IGP).

El objetivo principal planteado es una investigación sobre el Segment Routing y su aplicabilidad a través de protocolos subyacentes como BGP, IS-IS y PCEP en una red virtual de routers Cisco. Los resultados esperados son los beneficios en términos de escalabilidad y rendimiento al utilizar Segment Routing.

Una parte importante del Segment Routing tiene como objetivo el reenvío de paquetes sin requerir estado de flujo. Aprovecha el concepto de enrutamiento de origen, el cual busca una ruta óptima para un paquete basándose en su origen. En este contexto, permite que los routers tomen decisiones considerando la carga de tráfico, evitando la saturación de la red y brindando una distribución eficiente del tráfico de red.

La ingeniería de tráfico es un claro ejemplo de Segment Routing, el cual opera solo en el borde de la red [9]. Mediante el avance significativo de aplicaciones y servicios que realizan transferencia de datos mayormente elevada, se evalúa como un gran desafío implementar redes de alta velocidad.

Una metodología para la aplicación es Segment Routing el cual permite dirigir el tráfico de red por segmentos denominados identificadores [10]. En este contexto Segment Routing resalta en la simplicidad, escalabilidad y eficiencia en la manipulación de flujos de tráfico, en el cual permite una solución para la demanda de ancho de banda, optimización en la transmisión de datos, aquí entra un concepto latencia, el cual minimiza el retraso de comunicación en la red [11].

El enrutamiento de segmentos fue diseñado para la era SDN, Segment Routing y SDN (SDN-SR) son una combinación muy poderosa y presentan una propuesta ganadora para los proveedores de servicios proporcionando un rendimiento de red a los hosts finales, se debe lograr el ancho de banda disponible [12].

C. Objetivos

1. Objetivo General

Diseñar e implementar de Red Definida por Software (SDN) de alta velocidad utilizando la tecnología de Segment Routing.

2. Objetivos Específicos

- Investigar la tecnología de Segment Routing y su integración en Redes Definidas por Software.
- Diseñar e implementar en un entorno de emulación una Red Definida por Software de alta velocidad con la tecnología Segment Routing.
- Evaluar el impacto de Segment Routing en una Red Definida por Software en términos de rendimiento y eficiencia de la red de transporte.
- Documentar el proceso de diseño, implementación y pruebas, y evaluar la viabilidad de la solución para redes empresariales.

D. Alcance

Mediante la investigación e implementación de Red Definida por Software y Segment Routing se utiliza GNS3 como emulador de redes creando un entorno virtual que emula una red SDNe de alta velocidad. En el cual se implementa la tecnología Segment Routing como diseño específico aplicado en dispositivos Cisco IOS XRv ya que permite un enrutamiento flexible que se ejecuta en infraestructuras de red de alto rendimiento. Lo cual permitirá realizar pruebas para evaluar la aplicabilidad en entornos empresariales.

II. Marco teórico/Marco conceptual

A. *Redes Definidas por Software*

Las SDN inicialmente virtualizan la red separando el plano de control, que gestiona la red, del plano de datos por el que fluye el tráfico [13]. Hoy en día, las SDN van más allá y se utilizan para el control de redes amplias en la interconexión de redes empresariales. Además, se presentan como una solución para segmentar el tráfico de red aplicado a la seguridad. A medida que se utiliza el software para controlar la red, esta se vuelve más ágil y fácil de gestionar. Las SDN introduce una división clara entre el plano de control, donde residen las decisiones de enrutamiento y políticas de tráfico, y el plano de datos, donde los dispositivos de red ejecutan las instrucciones del controlador [14].

1. Arquitectura SDN

Los principales componentes de una red definida por software incluyen:

- **Capa de Aplicación:** En esta capa se maneja un conjunto de herramientas para administrar el comportamiento del plano de datos, donde las aplicaciones comunican las instrucciones de red al controlador.
- **Capa de Control:** Esta capa permite el uso del controlador SDN, el cual conecta la capa de aplicaciones a la capa de infraestructura. Aquí se procesan las peticiones enviadas por la capa de aplicación a través de una API de comunicación. Además, es responsable de la lógica de control y la toma de decisiones de enrutamiento.
- **Capa de Infraestructura:** Esta capa contiene los dispositivos de red (switches y routers) que reenvían el tráfico de acuerdo con las políticas definidas por el controlador. También procesa los datos para luego enviarlos a la capa de control.

Dentro de la arquitectura SDN, observamos en la Fig. 1 cómo se compone cada capa [14]. El funcionamiento se realiza mediante una API de comunicación. En este caso, puede ser Northbound entre el plano de control y el plano de aplicación, y Southbound entre el plano de control y la infraestructura o el plano de datos. Además, dentro del plano de control, puede manejarse westbound y eastbound [15].

- **API Northbound**: Esta API permite la comunicación entre el plano de control y el plano de aplicación. Mediante esta API se gestiona la infraestructura de red mediante RESTful.

Facilita la monitorización y automatización de redes de forma eficiente.

- **API Southbound**: Esta API permite la comunicación entre el plano de control y el plano de datos. Se comunica mediante el controlador OpenFlow. Permite obtener comunicación entre los dispositivos de red como switches y routers, y acceder y almacenar información acerca del tráfico y el estado de la red.

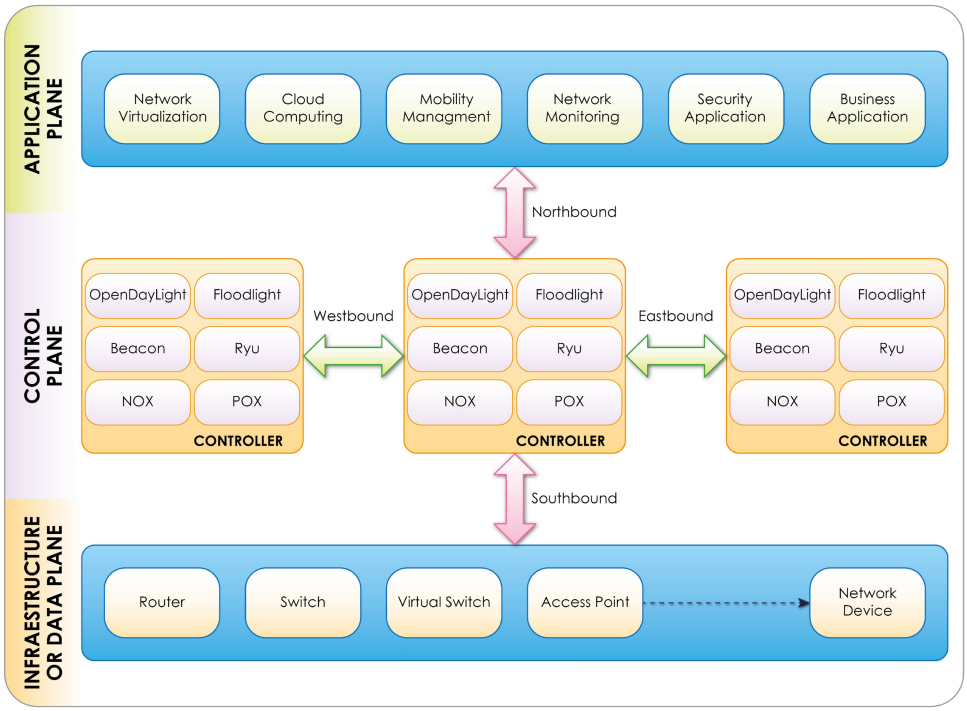


Fig. 1. Arquitectura funcional de SDN [7].

2. *Controladores y Tecnologías SDN*

En la arquitectura, existen componentes indispensables para un correcto funcionamiento de la tecnología SDN. Uno de ellos es OpenFlow, el cual está encargado de la comunicación entre el plano de control y el plano de datos, permitiendo el envío de paquetes y la gestión de tráfico [14].

Para trabajar en el controlador SDN, existen numerosos controladores, uno de los cuales es OpenDaylight, el cual gestiona el comportamiento de la red.

OpenFlow, como se mencionó anteriormente, es un protocolo encargado de la comunicación, permite el flujo de tráfico de red y toma decisiones de enrutamiento [16]. Además, permite configurar el comportamiento de los dispositivos que se encuentran en la capa de infraestructura.

Puede implementarse dentro de un proveedor de servicios para una mejor gestión de datos [17].

NETCONF (Network Configuration Protocol), Es un protocolo de gestión que permite configurar el estado de los dispositivos de red utilizando un modelo de datos YANG. Se basa en el transporte seguro de XML sobre SSH y proporciona mensajes de eventos y errores.

RESTCONF (Representational State Transfer), Este protocolo, al igual que NETCONF, permite la configuración de dispositivos. Lo que lo diferencia es el uso de operaciones HTTP y el transporte seguro de XML sobre HTTP, utilizado para el diseño web RESTful.

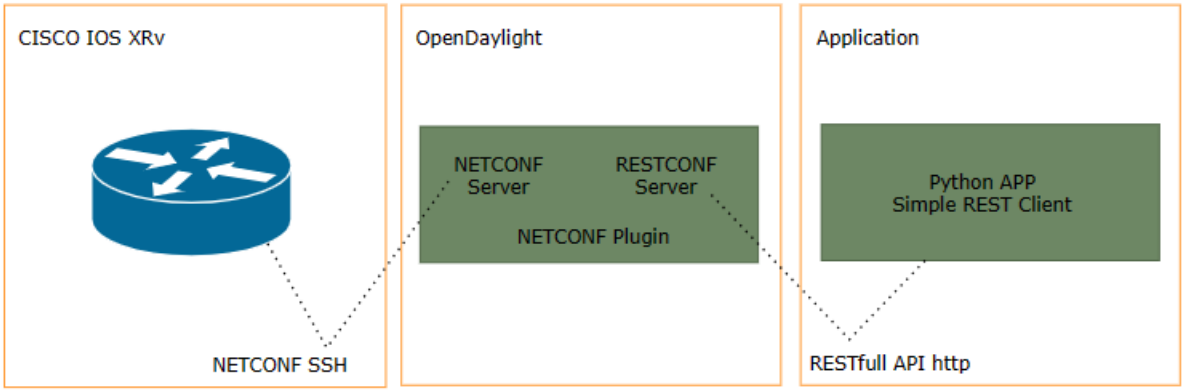


Fig. 2. Arquitectura funcional SND Netconf/Restconf [18].

BGP-LS (Protocolo de puerta de enlace fronteriza), a través de una extensión del BGP, se utiliza un mecanismo llamado Link State, el cual proporciona información sobre el estado de los enlaces. Esto permite el intercambio de información entre los sistemas autónomos. Para el controlador, posibilita la visualización detallada de la topología de la red.

PCEP (Path Computation Element Communication Protocol), Este protocolo facilita la comunicación entre un PCEP y los dispositivos de la red. Permite calcular rutas y determinar una ruta óptima, teniendo en cuenta la optimización del tráfico de datos, la latencia, entre otros aspectos [19].

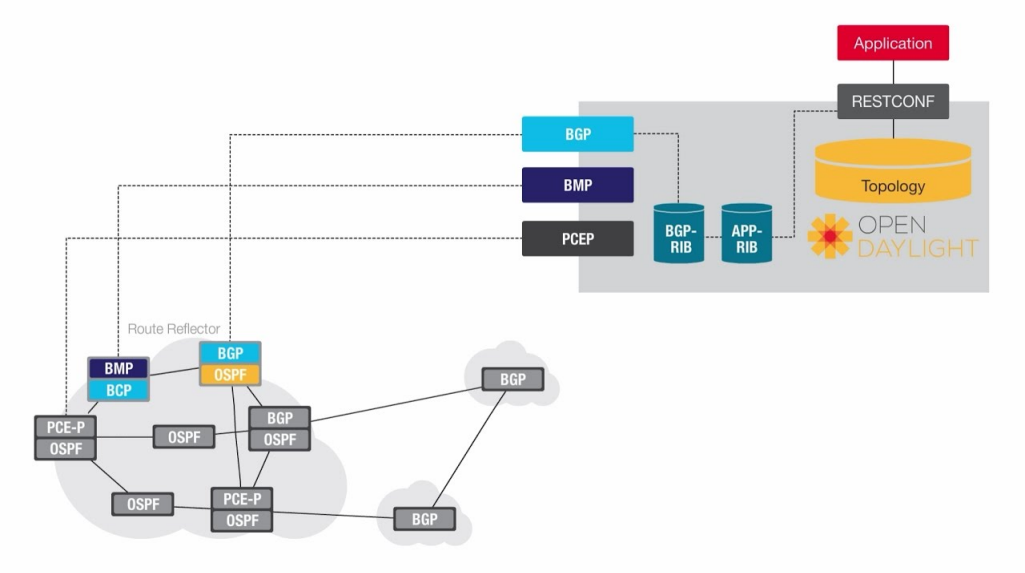


Fig. 3. Arquitectura funcional SND BGP-LS/PCEP [20]

OpenDaylight (ODL) es un proyecto de código abierto centralizado como un controlador de SDN. Su principal objetivo es proporcionar un controlador modular y escalable para diversas infraestructuras de red. ODL está escrito en el lenguaje de programación Java y busca acelerar el proceso de adopción de las SDN [14].

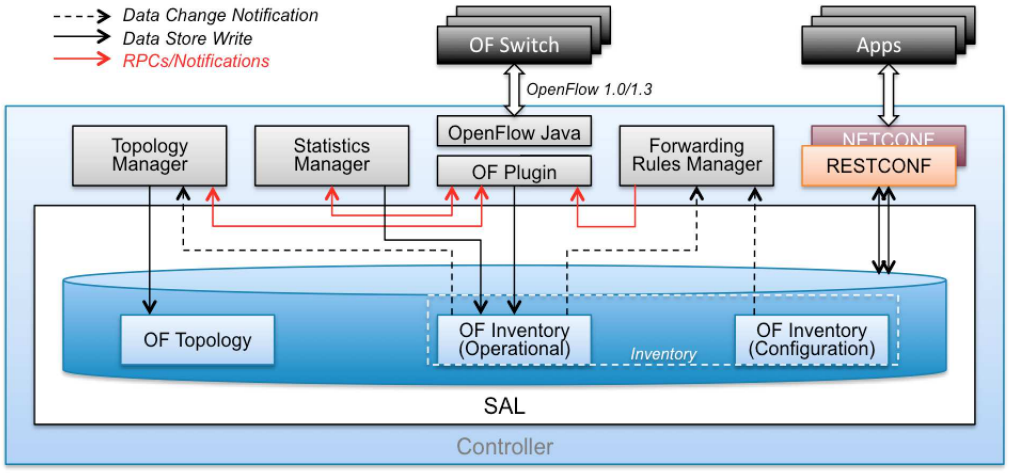


Fig. 4. Arquitectura funcional ODL [21].

ODL fue lanzado por la Linux Foundation y fue creado para desarrollar soluciones SDN. Su objetivo es proporcionar un controlador SDN que pueda modular y monitorizar arquitecturas de red. Actúa como el cerebro de la red, permitiendo el acceso a la capa de control de los

dispositivos subyacentes conectados. ODL trabaja y es compatible con diferentes protocolos, lo cual permite aprovechar la interoperabilidad.

B. Segment Routing (Enrutamiento por segmentos)

Dentro de un proveedor de servicios de internet, el Segment Routing (SR) o enrutamiento por segmentos brinda una mayor eficiencia para la gestión del tráfico de red, como se menciona en [22]. En este enfoque, la red se divide en segmentos, donde cada segmento tiene un identificador único. Estos identificadores se utilizan para definir las rutas de red. Anteriormente se mencionó que el envío de paquetes SR está configurado para identificar los segmentos de dicho envío, representando así un camino que debe seguir a través de la red [23].

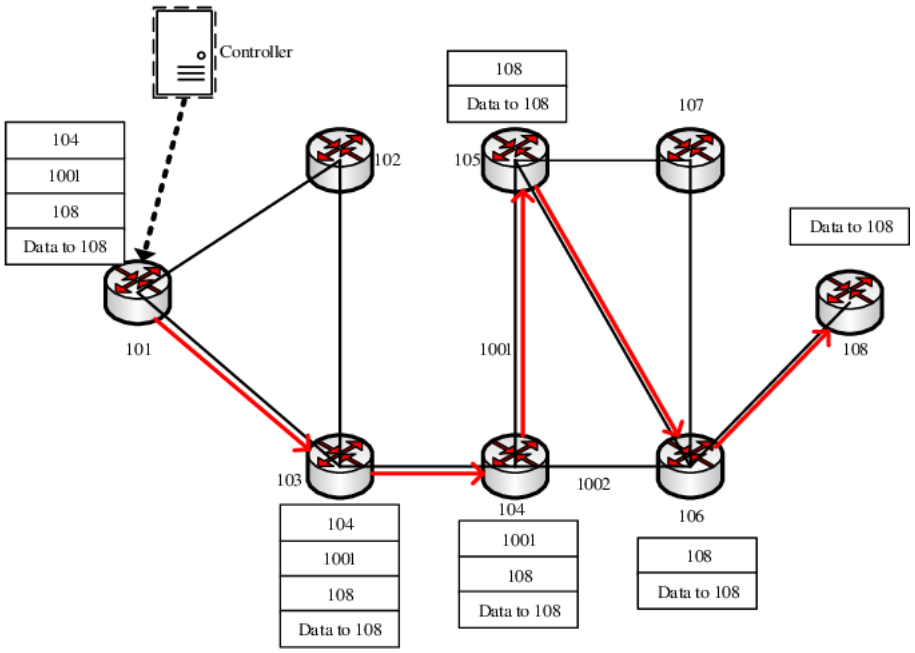


Fig. 5. Segment Routing funcionamiento [24].

El Segment Routing optimiza el tráfico de red al dirigir el envío de paquetes y garantizar la calidad de servicio. Además, permite la reducción de la congestión y mejora la utilización del ancho de banda. Dentro del Segment Routing (SR), existe una tecnología relacionada llamada Multiprotocol Label Switching (MPLS), la cual proporciona un rendimiento eficiente en la gestión del tráfico, como se menciona en [11].

El Segment Routing define rutas con identificadores llamados segmentos, mientras que MPLS utiliza etiquetas para conmutar paquetes a través de la red. Estas tecnologías destacan por

mejorar la eficiencia en la gestión de rutas y el rendimiento del tráfico en las redes [25].

Los IGP Segment Identifiers (IGP-SIDs), o Identificadores de Segmento de Protocolo de Enrutamiento Interior, son identificadores numéricos asignados a los enlaces o nodos de una red para facilitar el enrutamiento basado en segmentos. Esta técnica permite que los paquetes sigan una ruta predefinida a través de la red sin necesidad de consultar las tablas de enrutamiento en cada salto [22]. Los IGP-SIDs se pueden distribuir mediante protocolos de enrutamiento interior como OSPF o IS-IS.

OSPF (Open Shortest Path First) es un protocolo de estado de enlace ampliamente utilizado en redes IP. Admite varios tipos de redes, como punto a punto y multipunto, y utiliza el concepto de áreas para dividir una red en dominios más pequeños.

Por otro lado, IS-IS (Intermediate System to Intermediate System) también es un protocolo de estado de enlace, pero se ejecuta en la capa de enlace de datos. A diferencia de OSPF, IS-IS no utiliza direcciones IP para su funcionamiento, lo que lo hace más seguro.

Tanto OSPF como IS-IS pueden distribuir IGP-SIDs para Segment Routing, pero tienen diferencias en su funcionamiento y características. La elección entre ellos depende de la topología de la red y los requisitos específicos del diseño.

Los Prefix SIDs (Identificadores de Segmento de Prefijo) son extensiones del Segment Routing Global Block (SRGB) y representan segmentos de ruta dentro de un dominio de enrutamiento interior. En el contexto del Border Gateway Protocol (BGP), un BGP-SID es un identificador de segmento asignado a un prefijo BGP en una red de Segment Routing con BGP. Estos identificadores proporcionan instrucciones para enrutar paquetes hacia un prefijo relacionado a través de la mejor ruta calculada por BGP. Cuando los nodos BGP se comunican con sus vecinos, los mensajes BGP Update incluyen el Prefix-SID Label en el Labeled Unicast NLRI y un índice de Prefix-SID en un nuevo atributo llamado Prefix SID attribute [22].

Por otro lado, los Adjacency SIDs (Identificadores de Adyacencia) se utilizan para dirigir el tráfico sobre enlaces específicos. Cada nodo BGP asigna una etiqueta local a sus vecinos y anuncia esta etiqueta como un Adjacency SID a través de actualizaciones de estado de enlace BGP [22]. Esto permite que el camino de reenvío sea diferente del camino óptimo, lo que es útil para la ingeniería de tráfico. Por ejemplo, si se necesita forzar el tráfico sobre un enlace

específico, se utiliza un Adjacency-SID.

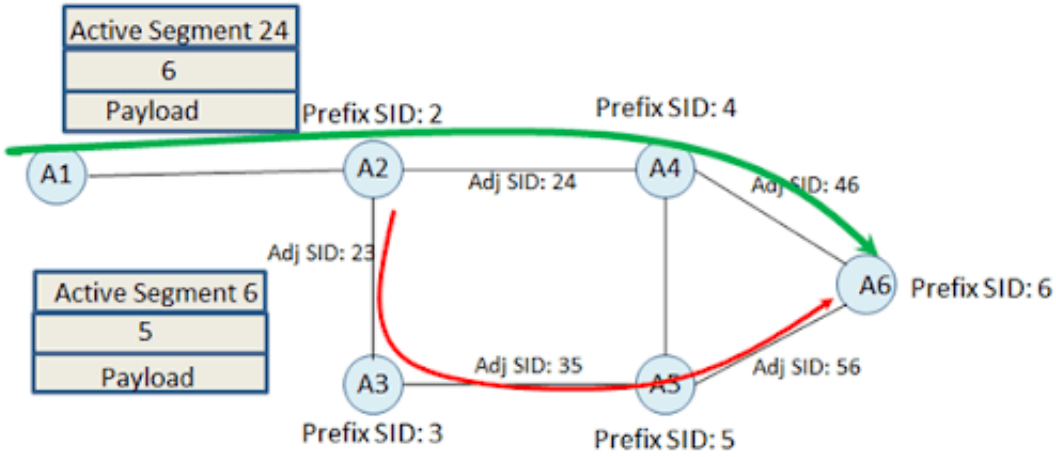


Fig. 6. Segment Routing-Prefix SIDs/Adjacency SIDs [24].

C. MPLS (Conmutación de etiquetas multiprotocolo)

Como su nombre lo indica, ”multiprotocol”significa que MPLS puede transportar varios protocolos además de IP, como IPv6, PPP, Frame-Relay, etc. MPLS es un protocolo de reenvío de paquetes etiquetados, donde estas etiquetas se utilizan como índice del siguiente salto y se asigna una nueva etiqueta para dirigir el paquete hacia su destino.

MPLS ha introducido servicios de Red Privada Virtual (VPN) y Calidad de Servicio (QoS) en toda la red. Cuando se diseña un túnel mediante el módulo de cálculo de rutas, este se establece utilizando protocolos de señalización como el Protocolo de Reserva de Recursos (RSVP) y/o el Protocolo de Distribución de Etiquetas (LDP) [26].

Dentro de un ISP, MPLS actúa en el enrutamiento de cada punto de la red. En primera instancia, el conmutador analiza el paquete que se envía y asigna una etiqueta que dirigirá el mismo. En cada conmutador que accede, el paquete llega con la etiqueta enviada y se asigna una nueva etiqueta. El uso de MPLS reduce la carga de procesamiento y acelera el proceso de enrutamiento en los dispositivos de red.

D. Segment Routing frente a MPLS

TABLA I
Comparación Segment Routing vs MPLS.

Características	Segment Rouitng	MPLS
Señalización de etiquetas	IGP	LDP+RSVP-TE
Sincronización IGP/LDP	No requerido	Requerido
Respaldo óptimo	Si	No
Soporte SDN	Si	No
Tipo de ruta	Basado en fuente	Basado en el destino
Cálculo de ruta	SPF o PCE restringidos	IGP+RSVP-TE
Escalabilidad	Alto	Bajo

E. Cisco IOS Xrv

El router Cisco IOS XRv es una plataforma basada en máquina virtual (VM) que ejecuta el software IOS XR de 32 bits con el microkernel QNX [27]. Proporciona soporte para el conjunto completo de funciones del software Cisco IOS XR, incluida la administración, las funciones del plano de control, el enrutamiento y el reenvío.

Este router admite funciones IP, como IPv4 e IPv6 de unidifusión y multidifusión. Además, dentro de los protocolos de enrutamiento, incluye el protocolo BGPv4, OSPFv2 y OSPFv3, IS-IS y MPLS.

Además, puede funcionar como un dispositivo virtualizado que puede ser monitoreado por un controlador SDN, el cual puede configurar el comportamiento de los dispositivos. También puede acceder a funcionalidades de servicios y aplicaciones dentro de SDN.

F. GNS3 (Simulador gráfico de red-3)

GNS3, es un software de simulación de codigo abierto y gratuito, permite crear topologías de manera virtual. GNS3 es muy popular en el uso de simulación y emulación de redes que

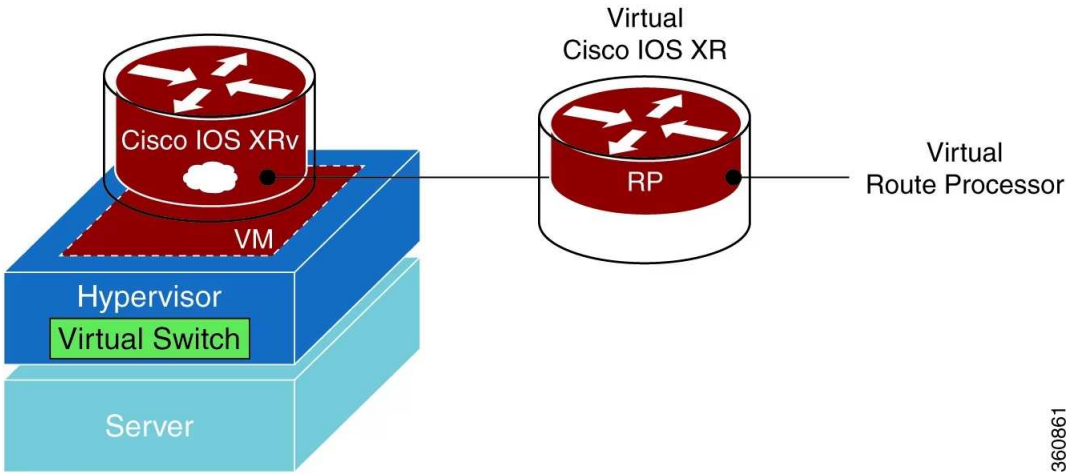


Fig. 7. Router Cisco IOS XRv [27].

permiten emular, configurar y solucionar problemas de redes tanto virtuales como reales.

Originalmente solo emulaba dispositivos Cisco utilizando un software llamado Dynamips, GNS3 ahora ha evolucionado y es compatible con muchos dispositivos de múltiples proveedores de red, incluidos conmutadores virtuales de Cisco, ASA de Cisco, vRouters Brocade, conmutadores Cumulus Linux, instancias de Docker, VSR de HPE, múltiples dispositivos Linux y muchos otros [28].

GNS3 consta de dos componentes dentro de su arquitectura uno es el software GUI, se maneja de forma local en la PC y la maquina virtual GNS3 esta puede ser tanto local como remota.

Fig. 8. GNS3 Logo [28].

G. Wireshark

Wireshark es un software de análisis de tráfico de red y protocolos que permite capturar y examinar los datos que se transmiten a través de una red. Con Wireshark, se puede analizar el tráfico de la red en tiempo real, capturar paquetes de red para su posterior análisis, identificar y solucionar problemas de red y seguridad, y examinar el tráfico de protocolos específicos [29].

Wireshark es capaz de interpretar y mostrar una gran cantidad de protocolos de red, desde los protocolos más comunes como TCP, UDP, HTTP, DNS, hasta protocolos menos conocidos o especializados. Además, el software puede filtrar y buscar paquetes de red específicos, y ofrece una variedad de herramientas de visualización y análisis para ayudar a los usuarios a entender y comprender los datos de red.

Fig. 9. Wireshark Logo [30].

H. Medologia PPDIOO

Para la implementación de la metodología se usa PPDIOO el cual se basa en la planificación, diseño, implementación, operación y optimización de redes de tecnología [31]. Fue desarrollado por Cisco Systems, primero asegura la gestión de redes y obtiene una eficiencia operativa a medida que evoluciona las necesidades tecnológicas [32].

Dicha metodología consta de seis fases:

- Preparación: Identifica los requisitos y analiza la infraestructura.
- Planificación: Planifica estrategias una vez analizados los riesgos.
- Diseño: Crea un diseño en base a los requerimientos, equipos y tecnologías a implementar.

Fig. 10. Router Cisco IOS XRv [33].

- Implementación: Construye la infraestructura de red, configura cada uno de los dispositivos y pruebas.
- Operación: Monitoreo o gestión de manejo.
- Optimización: Optimización y análisis de rendimiento

III. METODOLOGÍA/TÉCNICAS/DISEÑO

A. *Metodología de Desarrollo*

Para el desarrollo e implementación del presente trabajo se utiliza la metodología PP-DIOO, la cual se menciona en el capítulo II, sección del marco teórico. Esta metodología plantea un ciclo de vida continuo en las redes de telecomunicaciones, para lo cual se tiene definida cada fase del ciclo.

1. Preparación

En esta fase se realiza una investigación y análisis de cada uno de los requerimientos y características necesarias para la infraestructura de red que se pretende implementar. Para realizar la implementación, se definen los siguientes componentes:

Fig. 11. Componentes para la implementacion de la infraestructura

Cisco IOS XRv: Cisco IOS XR es una máquina virtual que ejecuta el software IOS XR, conteniendo un procesador de ruta con funcionalidad de plano de control. Esto permite la administración, características del plano de control y la funcionalidad del plano de datos.

Opendaylight: Controlador SDN que brinda una plataforma para la administración de redes, alojado en Linux. Dentro de la distribución se usa la versión Boron-SR2, encargada de brindar administración y monitoreo de la red.

Docker/Ubuntu: Contenedor de código abierto que implementa aplicaciones de forma más rápida en contenedores y las ejecuta en máquinas virtuales.

Firefox: Navegador desarrollado por Mozilla, que incluye características para mejorar la seguridad, velocidad, facilidad de uso y rendimiento, como la optimización de la memoria, el soporte para aceleraciones de hardware y la mejora de la velocidad de carga de las páginas.

Switch: Conmutador que permite la conexión de dispositivos, permitiendo que estos puedan comunicarse entre sí y con otras redes.

GNS3: Simulador de redes que permite emular, configurar y solucionar problemas de redes virtuales y reales.

2. Planificación

Para llevar a cabo la fase de planificación, es necesario establecer una serie de actividades para desarrollar la implementación adecuada y concluir con éxito. Dentro de la planificación, es crucial detallar las tecnologías y características fundamentales para la configuración. La Tabla III muestra cómo se desarrollan las fases del ciclo PPDIOO conforme a los entregables implementados.

- **Preparación:** Verificar los componentes y requisitos necesarios para la implementación.
- **Planificación:** Desarrollar una planificación de actividades que verifique el cumplimiento de la metodología PPDIOO.
- **Diseño:** Diseñar la infraestructura de red.
- **Implementación:** Configurar las interfaces y protocolos necesarios.
- **Operación:** Ejecutar y comprobar la implementación.
- **Optimización:** Realizar modificaciones para optimizar la implementación.

TABLA II

Planificación de actividades.

Fase	Entregable 1	Entregable 2	Entregable 3
Preparación	X		
Planificación	X		
Diseño		X	
Implementación		X	
Operación			X
Optimización			X

Entre los componentes más utilizados se encuentran los siguientes, los cuales se seleccionaron para la implementación de la infraestructura de red por las siguientes razones:

Cisco IOS XRv: Debido a su flexibilidad y compatibilidad para realizar configuraciones de implementación y gestión, se utilizó el router Cisco IOS XRv. Este permite realizar enrutamiento e implementación de protocolos como OSPF, MPLS, eBGP y Segment Routing. Los requisitos recomendados para su ejecución incluyen una CPU multicore de al menos 2 GHz para un rendimiento óptimo, así como 4 GB de RAM. También se necesita suficiente espacio de almacenamiento para la imagen del sistema operativo y los archivos de configuración, recomendándose al menos 8 GB de espacio libre en disco [27].

Opendaylight (ODL): Dentro de los controladores SDN, se decidió implementar Opendaylight debido a su capacidad para ofrecer administración y monitoreo de la red . En este caso, se accede a un router Cisco IOS XRv mediante la implementación del protocolo BGP para administrar todos los dispositivos enlazados al mismo y verificar el tráfico y envío de paquetes. Los requisitos para su ejecución incluyen Java JDK 8, y dentro de ODL, se necesitan funciones específicas que se instalan mediante el comando ”feature:install odl-bgpcep-bgp-all”[34].

Además, este controlador se alojó en un contenedor implementado en la máquina virtual de Ubuntu. Por lo tanto, también es necesario considerar un navegador web como Firefox para observar a través de la IP el acceso al controlador.

3. Diseño

En esta fase, se diseña la infraestructura lógica de red, para lo cual se establece que se emulará toda la infraestructura, lo cual no es necesario implementar, y se trabajará con recursos locales.

En la Fig. 12, se observa la siguiente terminología:

- CE: Enrutador perimetral del cliente.
- PE: Enrutador perimetral del proveedor.
- P: Enrutador de proveedor.
- SID de nodo: Identificador de segmento de un enrutador que participa en el enrutamiento de segmentos.
- AS: Sistema Autónomo.
- IS-IS: Protocolo de enrutamiendo interior IGP, permiter el intercambio de información a través del calculo de rutas eficientes.

- BGP: Protocolo de puerta de enlace fronteriza, protocolo de enrutamiento IP.
- SID de adyacencia: Identificador de segmento de una adyacencia en un dominio de enrutamiento de segmentos.

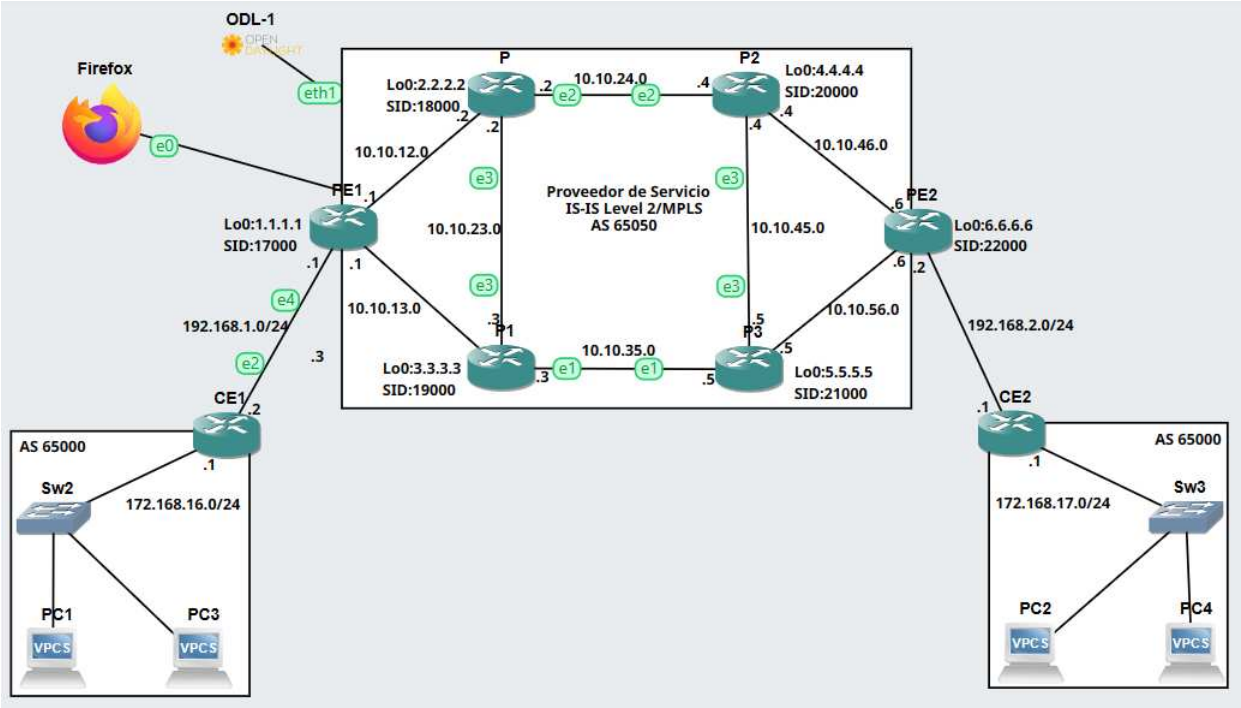


Fig. 12. Topologia de red

Esta topología de red está diseñada para probar la tecnología de Segment Routing con SDN. Desde el lado del controlador, ODL está conectado a través de un switch. A este dispositivo se le agrega un enlace al navegador y al router PE con el fin de obtener control sobre el mismo.

En la parte de la WAN, se implementa Segment Routing mediante ISIS y MPLS, mientras que en los routers de borde se utiliza BGP para tener conectividad de extremo a extremo.

En la Tabla III, se observa el direccionamiento IP utilizado para la configuración de las interfaces de cada uno de los dispositivos utilizados en la topología.

TABLA III

Tabla de enrutamiento.

Router	Interfaz	IP-Red/Máscara de red	Loopback
CE1	Gi0/0/0/0	172.168.16.1	10.10.10.10/32
		172.168.16.0/24	
	Gi0/0/0/1	192.168.1.2	
		192.168.1.0/24	
CE2	Gi0/0/0/0	172.168.17.1	20.20.20.20/32
		172.168.16.0/24	
	Gi0/0/0/1	192.168.2.1	
		192.168.2.0/24	
PE1	MgmthEth0/0/CPU0/0	192.168.100.1	1.1.1.1/32
		192.168.100.0/24	
	G0/0/0/0	10.10.12.1	
		10.10.12.0/24	
	G0/0/0/1	10.10.13.1	
		10.10.13.0/24	
	G0/0/0/2	192.168.2.2	
		192.168.2.0/24	
PE2	MgmthEth0/0/CPU0/0	192.168.100.6	6.6.6.6/32
		192.168.100.0/24	
	G0/0/0/0	10.10.46.6	
		10.10.46.0/30	
	G0/0/0/1	10.10.56.6	
		10.10.56.6/30	
	G0/0/0/3	192.168.2.2	
		192.168.2.0/24	
P1	G0/0/0/0	10.10.12.2	2.2.2.2/32
		10.10.12.0/24	

Continúa en la siguiente página...

Tabla III – continuación de la página anterior

Router	Interfaz	IP-Red/Máscara de red	Loopback
	G0/0/0/1	10.10.24.2	
		10.10.24.0/24	
	G0/0/0/2	10.10.23.2	
		10.10.23.0/24	
P2	G0/0/0/0	10.10.35.3	3.3.3.3/32
		10.10.35.3/24	
	G0/0/0/1	200.0.13.3	
		200.0.13.0/30	
	G0/0/0/2	10.10.23.3	
		10.10.23.0/24	
P3	G0/0/0/0	10.10.46.4	4.4.4.4/32
		10.10.46.0/24	
	G0/0/0/1	200.0.24.4	
		200.0.24.0/30	
	G0/0/0/2	10.10.45.4	
		10.10.45.0/24	
P3	G0/0/0/0	10.10.56.5	5.5.5.5/32
		10.10.56.0/24	
	G0/0/0/1	10.10.35.5	
		10.10.35.0/24	
	G0/0/0/2	10.10.45.5	
		10.10.45.0/24	
PC1	Eth0/0	172.168.16.2	
		172.168.16.3	
		172.168.16/24	
PC1	Eth0/0	172.168.17.2	
		172.168.17.3	
		172.168.17/24	

4. Implementación

INSTALACIÓN OPENDAYLIGHT

Para realizar la instalación del controlador, se utiliza el contenedor en el cual deben estar habilitadas las interfaces para obtener acceso a Internet y descargar la distribución Boron-SR2. Una vez realizada la descarga, se inicializa el contenedor y se procede a instalar Java JDK Environment 8. Luego, se agrega la variable de entorno **JAVA HOME** y se exporta el **PATH**.

Para ejecutar la distribución de Boron-SR2, se agrega **(./bin/karaf)** y se inicializa el controlador Opendaylight. [35].

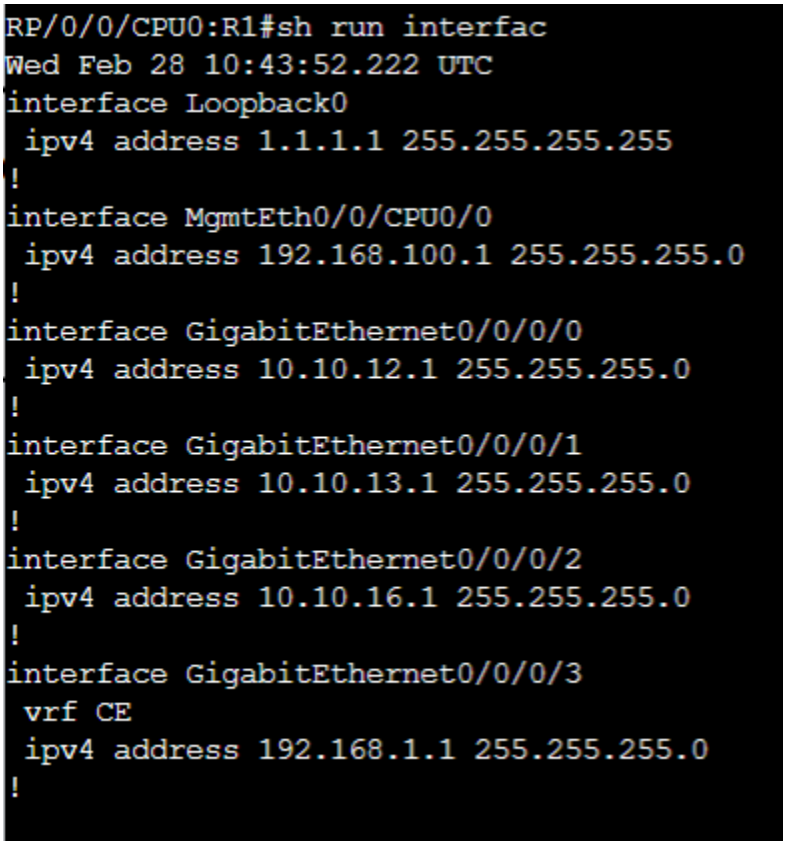
- Configurar el archivo /etc/network/interfaces con la IP del controlador
- Instalar las funciones en el controlador Opendaylight feature:install odl-bgpcep-bgp-all, feature:install odl-netconf-all, feature:install odl-dlux-core, feature:install odl-restconf odl-mdsal-apidocs

CONFIGURACIÓN DE LA RED

Inicialmente, se deben configurar las interfaces con cada una de las direcciones IP mencionadas en la Tabla III. En la Fig. 13, se observa que la interfaz de administración se enlaza directamente hacia el controlador ODL. Mientras que las interfaces Gi0/0/0/0 - Gi0/0/0/1 se enlazan a la WAN, que es el ISP, y la interfaz Gi0/0/0/3 se conecta a la LAN, que en este caso es el cliente.

En la configuración global, se configura una política de enrutamiento en la que se agrega la función **pass**, la cual permite que los paquetes se pasen sin ninguna modificación.

A continuación, se realiza la configuración del protocolo IS-IS, Segment Routing-MPLS [36]. Para ello, se define el nivel en el que va a trabajar el protocolo IS-IS y se agrega la dirección NET para definir el proceso. Se especifica la familia de direcciones, en este caso, es IPv4 unidifusión, y se define una métrica con valores grandes. Además, se habilita ISPF para realizar el cálculo de la ruta más corta por SPF y se habilita la ingeniería de tráfico MPLS para IS-IS. También se establece un ID de enrutador en la interfaz Loopback0.



```
RP/0/0/CPU0:R1#sh run interfac
Wed Feb 28 10:43:52.222 UTC
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 ipv4 address 192.168.100.1 255.255.255.0
!
interface GigabitEthernet0/0/0/0
 ipv4 address 10.10.12.1 255.255.255.0
!
interface GigabitEthernet0/0/0/1
 ipv4 address 10.10.13.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
 ipv4 address 10.10.16.1 255.255.255.0
!
interface GigabitEthernet0/0/0/3
 vrf CE
 ipv4 address 192.168.1.1 255.255.255.0
!
```

Fig. 13. Configuración interfaces en PE1

TABLA IV

Tabla interfaces Loopback, SIDs y Administración.

Router ID	Loopback	Managment Interface	Node SID
PE1	1.1.1.1	192.168.100.1	17000
P1	2.2.2.2	192.168.100.2	18000
P2	3.3.3.3	192.168.100.3	19000
P3	4.4.4.4	192.168.100.4	20000
P4	5.5.5.5	192.168.100.5	21000
PE2	6.6.6.6	192.168.100.6	22000

Se debe configurar la redistribución para las rutas conectadas por IS-IS y habilitar Segment Routing MPLS para establecer la distribución de etiquetas. En la interfaz Loopback 0, se configura el modo pasivo para que no participe en el enrutamiento de IS-IS. Dentro de la familia de direcciones IPv4, se asigna un identificador de segmento (SID), considerar la Tabla IV para las

interfaces Loopback y Node SID.

Finalmente, en las interfaces Gi0/0/0/0 y Gi0/0/0/1, se configura una conexión directa entre los dispositivos agregando la interfaz **point-to-point**, como se muestra en la Fig. 14.

```
router isis CORE-SR
 is-type level-2-only
 net 49.0000.0000.0000.0001.00
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
  ispf
  mpls traffic-eng level-2-only
  mpls traffic-eng router-id Loopback0
  redistribute connected
  segment-routing mpls sr-prefer
 !
 interface Loopback0
  passive
  address-family ipv4 unicast
  !
 !
 interface GigabitEthernet0/0/0/0
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface GigabitEthernet0/0/0/1
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface GigabitEthernet0/0/0/2
  point-to-point
  address-family ipv4 unicast
  !
```

Fig. 14. Configuración interfaces en PE1

Para un enrutamiento efiente de paquetes y que sea optimizado es necesario Para lograr un enrutamiento eficiente y optimizado de paquetes, es necesario configurar MPLS, como se muestra en la Fig. 15. Esta configuración habilita la función de Operación, Administración y Mantenimiento de MPLS. Además, se habilita el tráfico de ingeniería para las interfaces Gi0/0/0/0-Gi0/0/0/1, lo que optimiza el uso de enlaces y proporciona calidad de servicio (QoS).

Para optimizar el cálculo de ruta, se utiliza la tecnología PCE. Se especifica la dirección

```
RP/0/0/CPU0:R1#sh run mpls
Wed Feb 28 10:51:08.722 UTC
mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 pce
  peer source ipv4 192.168.100.1
  peer ipv4 192.168.100.100
  !
  segment-routing
  stateful-client
   instantiation
  !
  speaker-entity-id 1.1.1.1
 !
 auto-tunnel pcc
  tunnel-id min 1 max 99
 !
 reoptimize timers delay installation 0
!
mpls oam
!
```

Fig. 15. Configuración MPLS en PE1

IP de origen y la dirección de comunicación con el controlador PCE. Se agrega Segment Routing para dirigir el tráfico de la red y se configura un cliente SR, lo que crea un túnel SR comunicado por PCE. Además, se agrega una instancia para que PCE establezca la comunicación entre el túnel SR.

Además, se crea un identificador único para ese nodo, el cual se utiliza en Segment Routing para distinguir entre los nodos de la red y dirigirse específicamente a ese nodo. Se crea automáticamente un túnel PCC, que es un cliente de cálculo de rutas para el tráfico. Se especifica el rango de identificadores para el túnel PCC y se ajusta el tiempo de reoptimización del túnel.

Con la configuración del protocolo BGP, es posible redistribuir toda la información de IGP a OpenDaylight. La configuración de BGP se presenta en la Fig. 16. Al igual que en OSPF, en primer lugar se debe abrir la instancia BGP. Dentro de la configuración, se agrega la IP del controlador, en este caso, 192.168.100.100, con el AS 1, a través de la interfaz de administración.

Además, se utiliza la etiqueta de área 100, que es la misma que la etiqueta de Segment Routing.

Es importante anunciar la red hacia los vecinos BGP. Para ello, se habilita la familia de direcciones de estado de enlace BGP (**link-state**), que permite intercambiar información entre los enrutadores. Dentro de esta configuración, se deben registrar los detalles de los vecinos.

Se configura un vecino BGP con la dirección del controlador ODL, se especifica el número de AS del vecino y se establece la interfaz por la cual se envían las actualizaciones de BGP, como se observa en la Fig. 16. Además, se aceptan conexiones entrantes en modo pasivo desde el controlador y se habilita el estado de enlace BGP. Finalmente, se asigna la configuración de política de ruta entrante y salida del vecino.

```
RP/0/0/CPU0:R1#sh run router bgp
Wed Feb 28 13:27:23.240 UTC
router bgp 65050
 bgp router-id 1.1.1.1
 bgp log neighbor changes detail
 address-family ipv4 unicast
  network 10.10.12.0/24
  network 10.10.13.0/24
 !
 address-family link-state link-state
 !
 neighbor 6.6.6.6
  remote-as 65050
  address-family ipv4 unicast
  !
 !
 neighbor 192.168.1.2
  remote-as 65000
  ebgp-multihop 255
  ignore-connected-check
  address-family ipv4 unicast
   route-policy PASS in
   route-policy PASS out
  !
 !
 neighbor 192.168.100.100
  remote-as 65050
  update-source MgmtEth0/0/CPU0/0
  session-open-mode active-only
  address-family link-state link-state
   route-policy PASS in
   route-policy PASS out
```

Fig. 16. Configuración BGP en PE1

A continuación, especificamos el vecino BGP como un controlador SDN, como se indica en la Fig. 17. El controlador SDN pertenece al AS remoto 1, y la interfaz de administración se utiliza para actualizar al controlador SDN sobre la información de estado de los vínculos. Además, habilitamos el intercambio de políticas. Este archivo de configuración se realiza dentro del directorio de distribución **/etc/opendaylight/karaf** en un archivo llamado **bgp-example.xml**.

```
 GNU nano 4.8                        41-bgp-example.xml

       <data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
           <modules xmlns="urn:opendaylight:params:xml:ns:yang:controller:conf
>              <module>
                   <type xmlns:prefix="urn:opendaylight:params:xml:ns:yang:con
>                  <name>example-bgp-peer</name>
                   <host>192.168.100.1</host>
                   <peer-role>ibgp</peer-role>
                   <remote-as>65050</remote-as>
```

Fig. 17. Configuración ODL archivo XML

Dentro de todas las configuraciones necesarias para lograr una conexión exitosa, se configura un VRF entre los PE, como se muestra en la Fig. 18, el cual sirve como medio de comunicación con los clientes. La implementación de VRF dentro de la topología permite segmentar el tráfico IP de los clientes acompañado de encapsulación MPLS.

```
RP/0/0/CPU0:PE-1#sh run
Fri Feb 16 06:31:01.251 UTC
Building configuration...
!! IOS XR Configuration 6.1.3
!! Last configuration change at Fri Feb 16 05:29:42 2024 by cisco
!
hostname PE-1
vrf Ce
 address-family ipv4 unicast
  import route-target
   65000:2
  !
  export route-target
   65000:1
  !
 !
!
```

Fig. 18. Configuración de la VRF para CE

En OpenDaylight, para establecer conexión con el plano de datos, se utilizan varios protocolos, uno de ellos es NETCONF, el cual está disponible para los dispositivos Cisco. Esta conexión se establece a través de SSH, donde se declara un servidor NETCONF y se especifica el puerto correspondiente. Como se observa en la Fig. 19

```
hostname R1
username eabv
 password 7 09494F0B0F
!
netconf agent tty
!
netconf-yang agent
 ssh
!
ssh server v2
ssh server netconf port 1830
```

Fig. 19. Credenciales de sesión para el Router PE

En la consola del controlador OpenDaylight, una vez instaladas las librerías necesarias, se agrega cada dispositivo para establecer una sesión entre el controlador y el dispositivo, en este caso un router Cisco IOS XRv. Para ello, es necesario agregar la dirección IP, el puerto de NETCONF, el nombre de usuario y la contraseña de cada uno de los dispositivos.

En el dispositivo, se configura la sesión de acceso habilitando en la configuración global el protocolo netconf-yang, el cual se conecta mediante SSH. Además, se crea una sesión por SSH donde se especifica el puerto por el cual el dispositivo se conecta al controlador OpenDaylight mediante NETCONF.

Con esta API, se configura BGP en el controlador, indicando el sistema autónomo al que va a pertenecer, su identificador dentro del sistema y que tendrá habilitado BGP-LS para conocer la topología en tiempo real. Luego, se puede configurar los vecinos. En este caso, el vecino será uno de los routers de la red desplegada, específicamente PE1, y será este el encargado de pasarle toda la información de la red al controlador, como se muestra en la Fig. 20. La configuración para esto se muestra a continuación.

Fig. 20. BGP - PCEP

Una vez que el controlador está listo, se puede utilizar Postman, ya que el protocolo RESTCONF permite que el controlador interactúe con aplicaciones de terceros. Para conectar el router Cisco, se realiza una obtención de datos a través de la operación GET de la siguiente forma:

http://192.168.100.100:8181/restconf/config/network-topology:network-topology/topology/topology-netconf

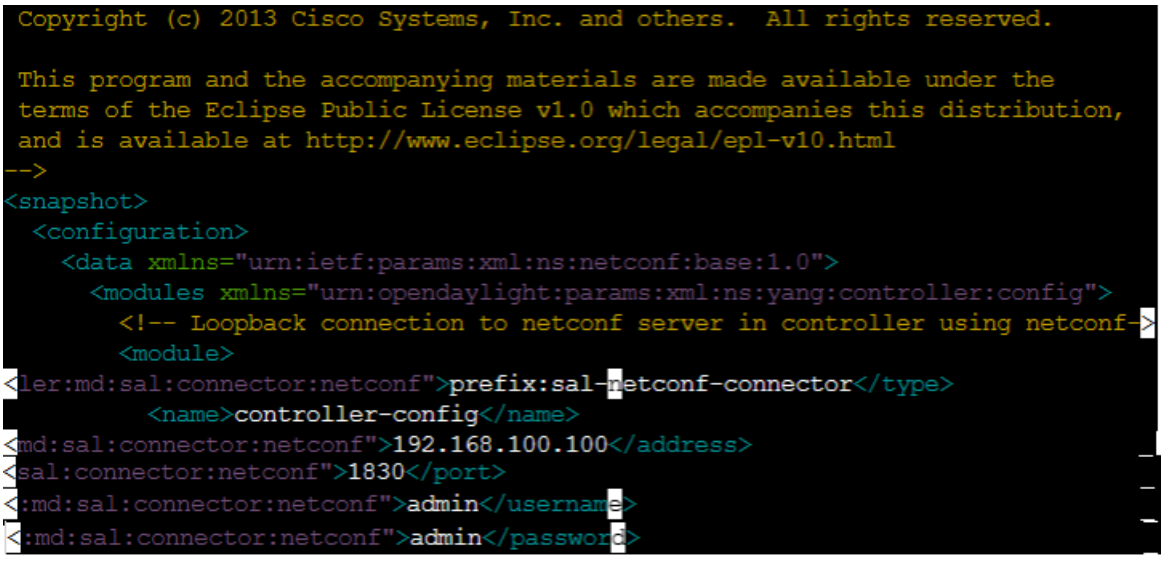


Fig. 21. API de acceso a la topología

5. Operación

En la fase de operación se observan las pruebas de conectividad y las respuestas mediante la configuración de los protocolos anteriormente mencionados. En la Fig. 22, se distinguen las rutas de envío.

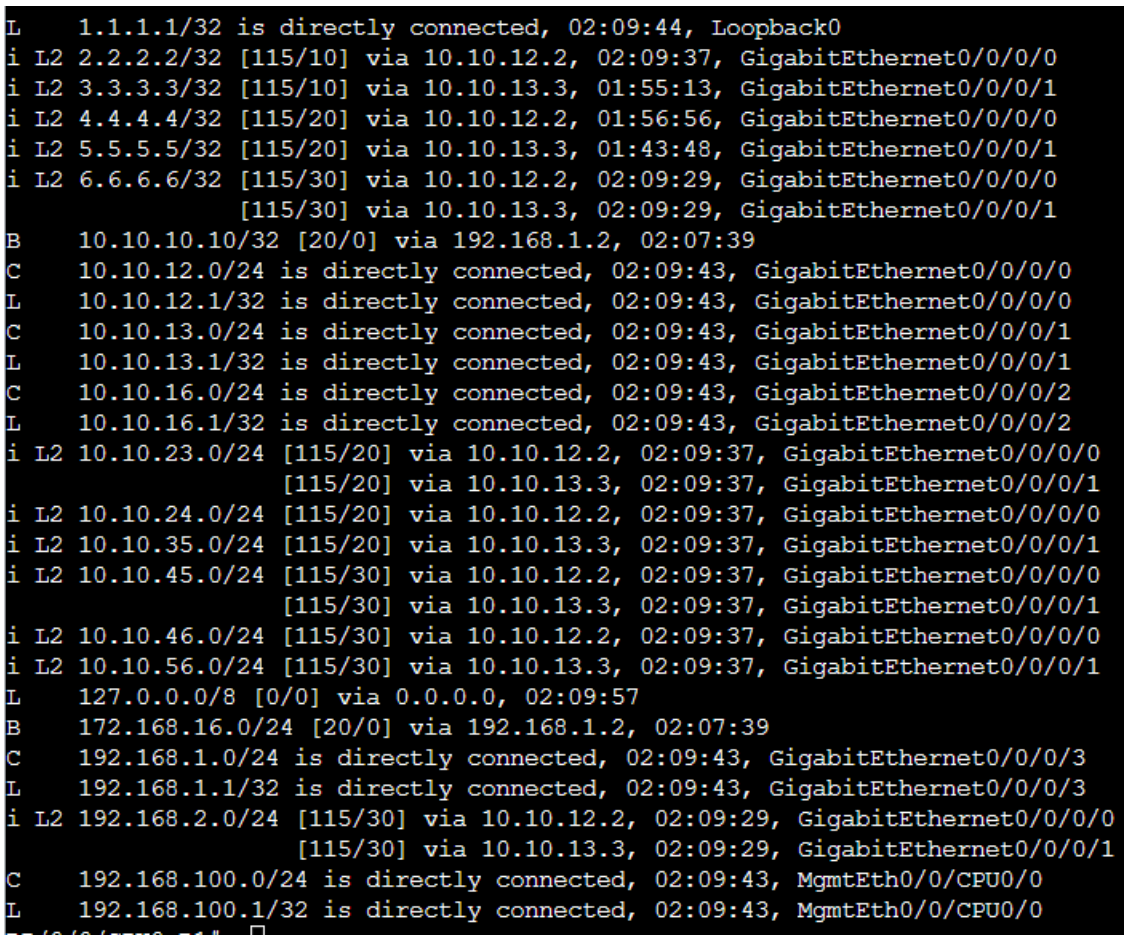


Fig. 22. Rutas OSPF

Se observa la tabla de enrutamiento actual donde se obeserva el alcance de las redes específicas del dispositivos Cisco IOS XRv implementando los protocolos ISIS, BGP, MPLS.

En la Fig. 23, se observan las etiquetas de Segment Routing asignadas para cada nodo de adyacencia. En este caso, se indican los prefijos agregados en cada router, y cómo se realiza la adyacencia mediante las etiquetas asignadas por MPLS, asociando las rutas e interfaces de salida.

```
RP/0/0/CPU0:R1#sh mpls forwarding
Tue Feb 27 19:37:34.365 UTC
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
19000  Pop         SR Pfx (idx 3000) Gi0/0/0/1    10.10.13.3      0
20000  20000       SR Pfx (idx 4000) Gi0/0/0/0    10.10.12.2      0
21000  21000       SR Pfx (idx 5000) Gi0/0/0/1    10.10.13.3      0
22000  22000       SR Pfx (idx 6000) Gi0/0/0/0    10.10.12.2      0
       22000       SR Pfx (idx 6000) Gi0/0/0/1    10.10.13.3      0
24000  Pop         SR Adj (idx 1)    Gi0/0/0/0    10.10.12.2      0
24001  Pop         SR Adj (idx 3)    Gi0/0/0/0    10.10.12.2      0
24002  Pop         SR Adj (idx 1)    Gi0/0/0/1    10.10.13.3      0
24003  Pop         SR Adj (idx 3)    Gi0/0/0/1    10.10.13.3      0
RP/0/0/CPU0:R1#
```

Fig. 23. MPLS Forwarding.

Para comprobar conectividad entre los dispositivos se realiza una prueba de ping, en la Fig. 24, muestra que se realizó con éxito. Esta prueba ayuda a rastrear la comunicación entre dispositivos conectados a la red. El comando ping también ayuda a probar la conectividad y medir el tiempo de respuesta.

```
RP/0/0/CPU0:PE-1#ping 200.0.11.2
Fri Feb 16 06:40:23.232 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.0.11.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
RP/0/0/CPU0:PE-1#
```

Fig. 24. Prueba de conectividad a PE-2.

En la Fig. 25, se realiza un traceroute para detectar la ruta que toman los paquetes enviados. Al ejecutar un traceroute, se envía una serie de paquetes ICMP, en su mayoría con incrementos progresivos en el campo de tiempo de vida (TTL) de la cabecera IP. Cada salto en la red decrementa el valor del TTL. Cuando el TTL llega a cero, el router descarta el paquete y envía un mensaje de ”tiempo excedido”(Time Exceeded) al remitente.

Además, se muestra el tiempo que ha tardado en alcanzar cada uno de estos routers, lo que puede ser útil para identificar posibles cuellos de botella o problemas de latencia en la red.

```
RP/0/0/CPU0:R1#traceroute 6.6.6.6 source 1.1.1.1
Fri Mar  1 17:50:12.839 UTC

Type escape sequence to abort.
Tracing the route to 6.6.6.6

 1  10.10.13.3 [MPLS: Label 22000 Exp 0] 39 msec  9 msec  9 msec
 2  10.10.35.5 [MPLS: Label 22000 Exp 0] 19 msec  9 msec  9 msec
 3  10.10.56.6 19 msec  *  9 msec
RP/0/0/CPU0:R1#
```

Fig. 25. Traceroute para PE-1.

6. Optimización

La implementación de listas de control de acceso (ACL) en los nodos de enrutamiento es esencial para optimizar el segmento de enrutamiento. Esta etapa implica configurar los dispositivos de red para implementar políticas ACL específicas que afectarán el enrutamiento de paquetes según las necesidades de la red.

Inicialmente, se realizó un análisis donde se incluyó la identificación de aplicaciones sensibles al rendimiento y otras cargas de trabajo que requieren ancho de banda garantizado y baja latencia. Basándose en este análisis, crean reglas ACL para priorizar y dirigir estos flujos de tráfico importante a través de las rutas de red óptimas. Por ejemplo, se establecieron reglas ACL que asignan mayor ancho de banda y menor latencia a los paquetes de VoIP, permitiendo una comunicación fluida y sin interrupciones.

Otra parte importante, la calidad de servicio (QoS) en redes de comunicaciones es un conjunto de tecnologías y técnicas diseñadas para asegurar un rendimiento satisfactorio de los servicios ofrecidos. Su objetivo es gestionar eficientemente los recursos de red para cumplir con los requisitos de rendimiento de aplicaciones y usuarios. La QoS se centra en aspectos como el ancho de banda, la latencia, el jitter y la pérdida de paquetes. Para implementarla, se utilizan mecanismos como la clasificación y priorización de tráfico, la limitación y control de congestión.

IV. RESULTADOS

En esta sección, se presentan los resultados obtenidos a través de la implementación de la red simulada para un proveedor de servicios en el entorno virtual GNS3, utilizando SDN el controlador Opendaylight, protocolo de comunicación Netconf y Segment Routing.

A. Controlador Opendaylight

Una vez realizadas las configuraciones se establece una conexión con el controlador, se enlaza una conexión física desde el router Cisco IOS XRv a través de la interfaz de administración MgEth0/0/CPU0/0 hacia la interfaz del controlador. Para verificar el funcionamiento del controlador OpenDaylight, se inició sesión en un navegador web utilizando la dirección IP del controlador y puerto establecido 192.168.100.100:8181, además agrega las credenciales Como se muestra en la Fig. 26.

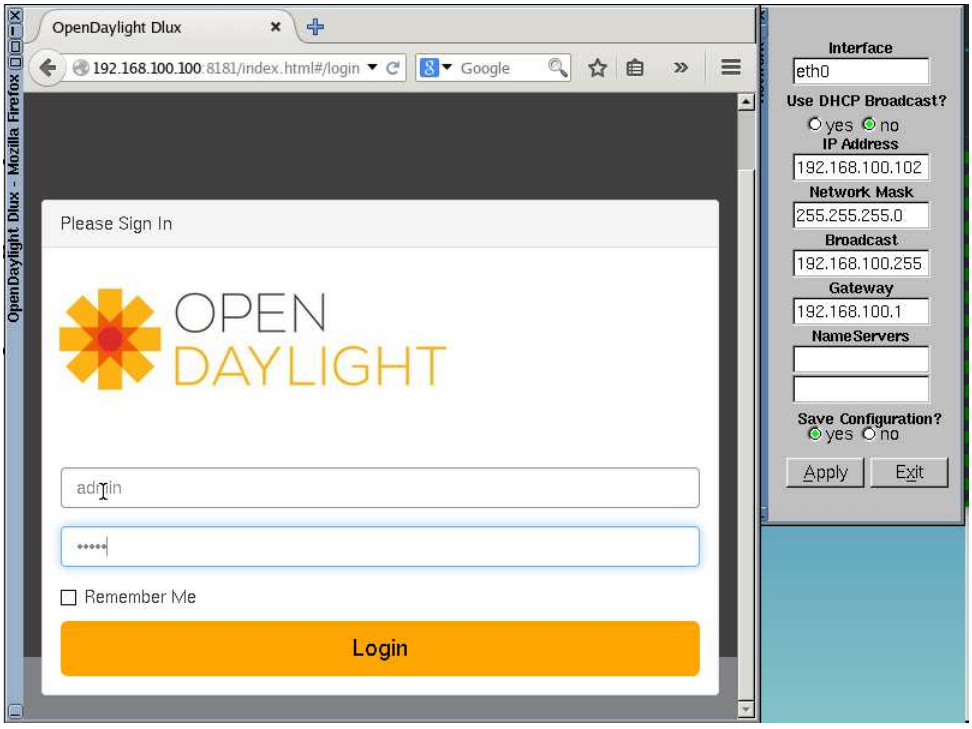


Fig. 26. Inicio de sesión al controlador Opendaylight.

Dentro de la interfaz gráfica del controlador Opendaylight, permite administrar los dispositivos enlazados. El protocolo de comunicación entre Opendaylight y el router Cisco IOS XRv

es Netconf trabaja a través de API y archivos XML, lo que facilita la gestión de dispositivos de red.

Una vez activo el controlador Opendaylight es necesario crear la sesión de comunicación con los dispositivos. Se especifican un nombre de usuario y una contraseña para el router, y luego el controlador OpenDaylight agrega el dispositivo Cisco utilizando estas credenciales junto con el puerto por defecto de Netconf.

El router Cisco IOs XRV establece una conexión con el servidor Netconf y abre la sesión Netconf. A través del protocolo Netconf permite la configuración, monitoreo y operación de dispositivos de red de manera remota y programática. En la Fig. 27, se observa un identificador de la sesión actual que se establece, protocolo de conexión, usuario por el cual se conecta, fecha donde se crea la sesión y el estado si se encuentra bloqueado.

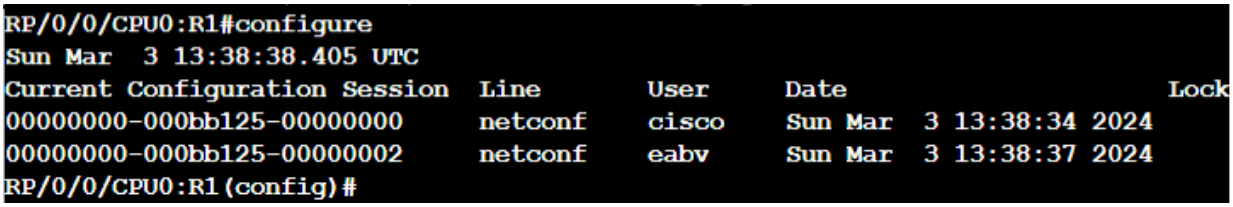


Fig. 27. Sesión establecida entre el router PE y controlador Opendaylight.

Dentro del CLI del controlador Opendaylight, mediante la herramienta Netconf, agrega los dispositivos con los que estableces sesión, en la Fig. 28, indica el identificador, dirección IP, puerto establecido para Netconf y verifica el estado si ha logrado enlazarse.

Es importante declarar que en el router Cisco IOS XRv, debe establecer y habilitar la sesión y el usuario para conectarse, puede realizar una prueba de conectividad a través de la interfaz de administración más no establecer comunicación por el protocolo Netconf. Se verifica en el controlador la lista de los dispositivos enlazados, a través de Netconf.

```
opendaylight-user@root>netconf:list-devices
NETCONF ID                       | NETCONF IP    | NETCONF Port | Status
-------------------------------------------------------------------------
1805e770c7be4f369ebc6989814071c1 | 192.168.100.6 | 1830         | connecting
0326fdba083a47feb750dbddf2fdd998 | 192.168.100.5 | 1830         | connecting
controller-config                | 127.0.0.1     | 1830         | connected
ed2878a37ee741e79ce6e87c690f90b5 | 192.168.100.1 | 1830         | connected
8412d9a60247461d98a5982dcc8ca24c | 192.168.100.1 | 1830         | connected
6bb6956df0cf4f7da38d119896a7116c | 192.168.100.1 | 1830         | connected
d23f7043a3f04885a6b91937a0577a88 | 192.168.100.6 | 1830         | connecting
```

Fig. 28. Sesión establecida entre el router PE y controlador.

En la Fig. 29, se muestra en el router Cisco XRv, que se levanta el protocolo PCE con el controlador. En la documentación tenemos que PCE permite el intercambio de información de estado y de capacidad de la red, creando rutas de para evaluar la demanda de tráfico de datos.

Además, se observa entre el router PE y el controlador Opendaylight se establece conexión y comunicación.

```
RP/0/0/CPU0:R1#sh mpls traffic-eng pce peer
Sun Mar  3 14:32:20.424 UTC
        Address    Precedence        State          Learned From
--------------- ------------ ------------ --------------------
192.168.100.100           255           Up         Static config
RP/0/0/CPU0:R1#
```

Fig. 29. Segment Routing captura de tráfico.

En las Fig. 27, 28 y 29, se observa la conexión y comunicación establecida entre la topología de red y el controlador Opendaylight, parcialmente aplicando la funcionalidad de SDN dentro de la red emulada.

Como se menciona anteriormente, el protocolo Netconf trabaja con archivos XML y API, una vez conectado, se puede realizar una solicitud GET para obtener la información del router. En el acceso a la interfaz del controlador, tenemos la posibilidad de acceder mediante la API para realizar operaciones, como se observa en la Fig. 31.

http://192.168.100.100:8181/restconf/config/network-topology:network-topology/topology/topology-Netconf

Para trabajar con solicitudes, es necesario revisar en la documentación oficial y adecuar

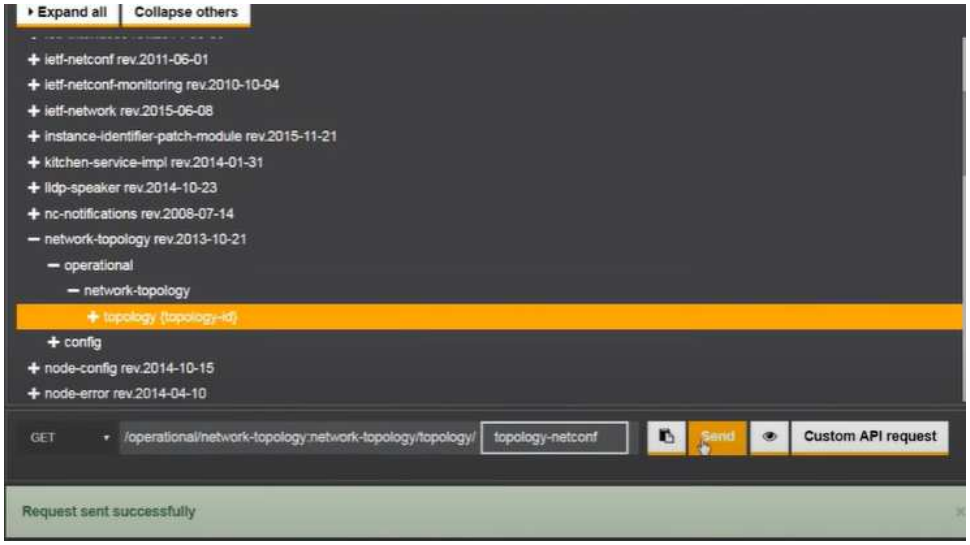


Fig. 30. API de acceso a la topología.

a los requerimientos necesarios. En este caso sé establece conexión, comunicación y solicitudes de respuesta.

B. Segment Routing

En la Fig. 31, se muestran dos entradas de métrica con un valor de 10 cada una. La primera métrica, asociada con la versión del protocolo IS-IS R2.00, se aplica a la conexión entre el router R1 y otro router identificado como R2. Esta conexión tiene direcciones IP específicas para la interfaz del router R1 (10.0.12.1) y para el vecino R2 (10.0.12.2). Se han asignado dos identificadores de Segmento de Adyacencia (ADJ-SID) con valores de 24000 y 24001 para esta conexión.

La segunda métrica, asociada con la versión del protocolo IS-IS R2.00, se aplica a la conexión entre el router R1 y otro router identificado como R2. Esta conexión tiene direcciones IP específicas para la interfaz del router R1 (10.0.13.1) y para el vecino R2 (10.0.13.3). Se han asignado dos identificadores de Segmento de Adyacencia (ADJ-SID) con valores de 24002 y 24003 para esta conexión.

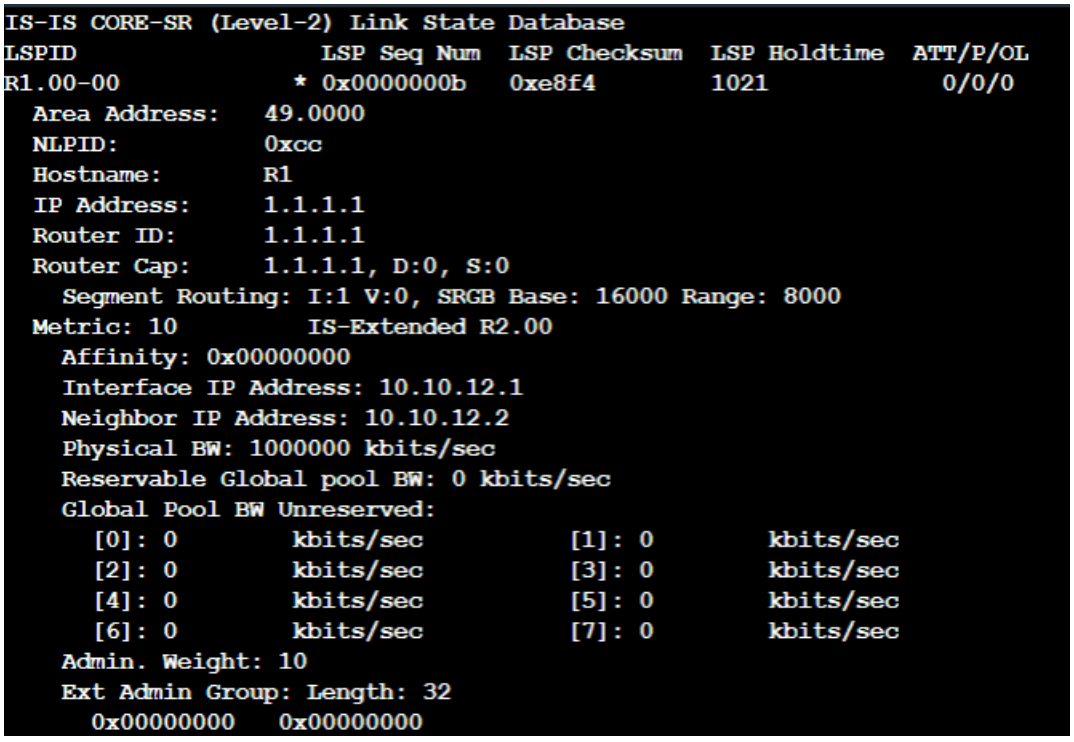


Fig. 31. Información sobre el estado de enlace IS-IS.

Para evaluar topología de red emulada, se usa la herramienta de captura de tráfico Wireshark, esta se encarga de analizar los paquetes de la red que se envían a través de la red. Wireshark permite ver los protocolos de comunicación que se implementaron en la topología de este trabajo.

Mediante la captura de tráfico realizado con esta herramienta, se podrá evidenciar estadísticas generales y específicas que se realizan, y así evaluar el rendimiento de toda la red.

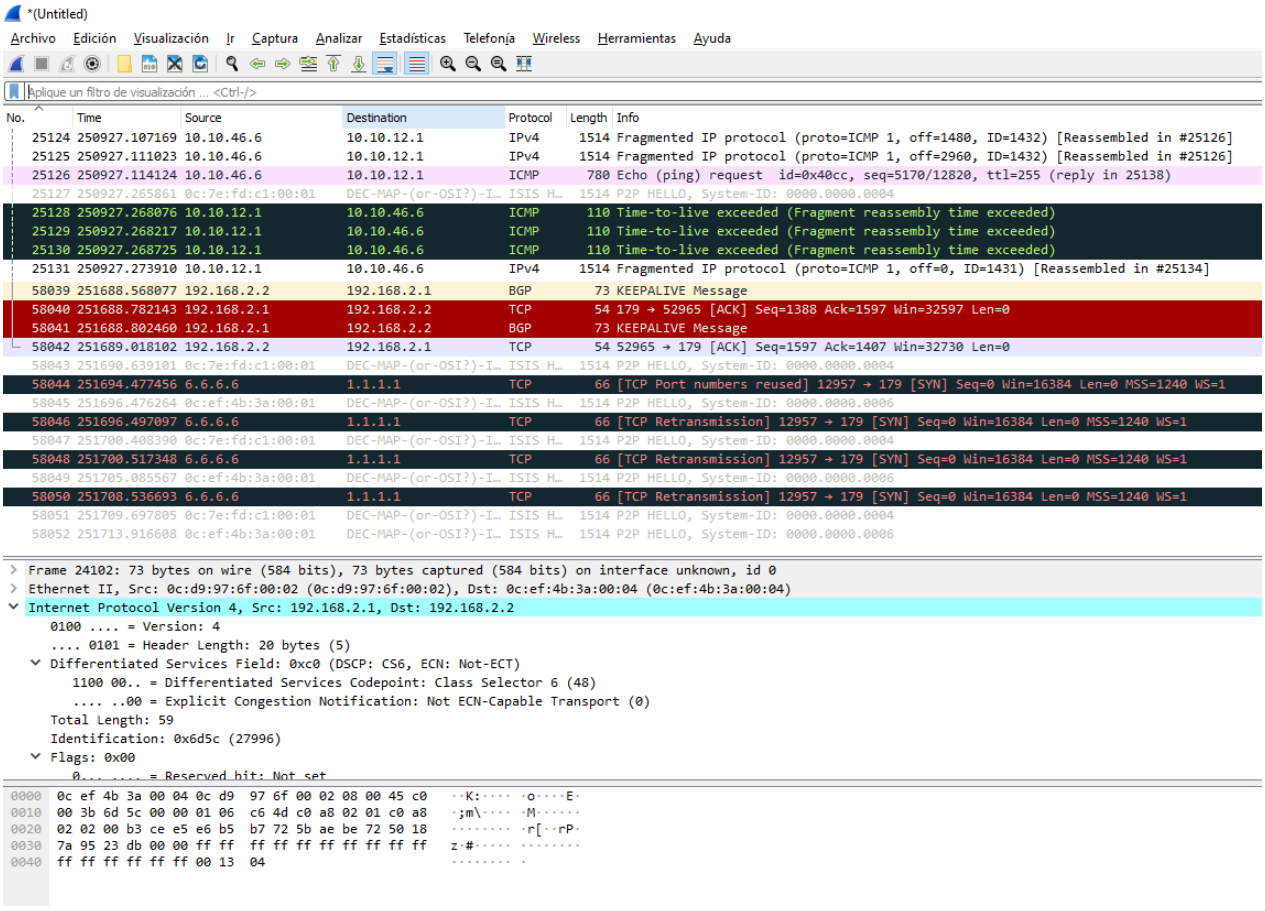


Fig. 32. Captura de tráfico usando Wireshark.

Dentro de la primera evaluación, en la Fig. 33 del rendimiento de la red implementando segment routing Se enviaron 89945 paquetes ICMP Echo de tamaños crecientes (de 36 a 18024 bytes) a la dirección IP 192.168.1.2. El timeout para cada paquete fue de 2 milisegundos. La tasa de éxito fue del 96% (7670 paquetes recibidos correctamente de los 7921 enviados). Los tiempos de ida y vuelta (round-trip) mínimo, promedio y máximo fueron de 1 ms, 9 ms y 1929 ms respectivamente.

Estos resultados indican que la mayoría de los paquetes fueron entregados correctamente, con un tiempo promedio de ida y vuelta de 9 ms. Sin embargo, algunos paquetes experimentaron tiempos de entrega más largos, con un tiempo máximo de 1929 ms, lo que sugiere posibles problemas de latencia en la red o en el dispositivo de destino.

```
RP/0/0/CPU0:R6#ping 192.168.1.2 validate size 36 sweep
Mon Mar  4 16:43:45.271 UTC
Type escape sequence to abort.
Sending 89945, [36..18024]-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
CCCCCCCCCC!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*
* The idle timeout is soon to expire on this line
*

Success rate is 96 percent (7670/7921), round-trip min/avg/max = 1/9/1929 ms
```

Fig. 33. Evaluación de latencia.

Ahora se establece tres pruebas en el mismo contexto con diferentes tamaño de paquetes de envío, esto se realiza con el fin de evaluar el rendimiento y poder concluir sobre la demanda de tráfico, las cuales se evidencian en la Fig. 34.

- Ping con tamaño de paquete 18024 bytes: la conectividad entre tu dispositivo y la dirección IP 192.168.1.2 es buena, con una tasa de éxito del 96%. Los tiempos de respuesta son generalmente bajos, con un promedio de 8 ms, lo que sugiere una red con baja latencia y buen rendimiento. Sin embargo, algunos paquetes experimentaron tiempos de respuesta más altos, con un máximo de 29 ms, lo que podría indicar congestión temporal en la red.
- Ping con tamaño de paquete 9012 bytes: De los 7873 paquetes enviados, 7634 recibieron una respuesta exitosa, lo que representa una tasa de éxito del 96%. Los tiempos de ida y vuelta (round-trip) registraron un mínimo de 1 ms, un promedio de 9 ms y un máximo de 1839 ms. Esto indica una buena conectividad en general, aunque se observa una variabilidad en los tiempos de respuesta, posiblemente debido a las variaciones en la latencia.
- Ping con tamaño de paquete 36 bytes: a mayoría de los paquetes de datos enviados recibieron una respuesta exitosa, lo que indica una conectividad sólida con la dirección IP de destino. Sin embargo, algunos paquetes experimentaron tiempos de respuesta más altos, con un máximo de 1779 ms.

En los tres pruebas de ping que se realizó, se obtiene un 96% de la tasa de éxito al enviar una prueba de conectividad con diferente tamaño de paquetes, lo que nos porporciona saber que el tiempo de ida y vuelta varia de 1-1839ms, con promedio de 8-9ms para las tres pruebas que se enviarón con tamaños diferentes de paquetes. Con ello, se establece un tiempo promedio y se podría decir que independientemente del tamaño del paquete entregara o realiza un envío de paquetes en tiempo mínimo.

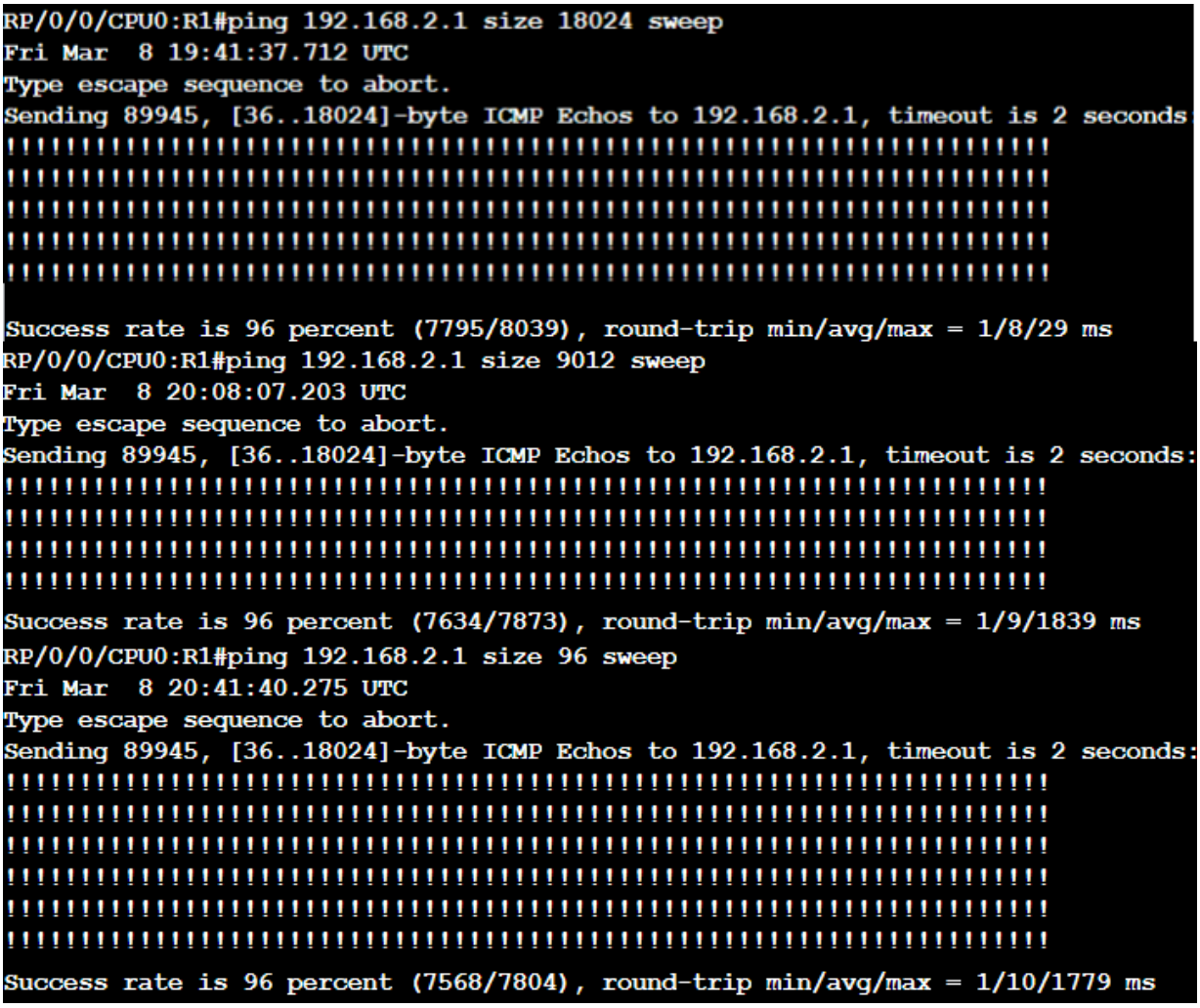


Fig. 34. Evaluación de rendimiento throughput.

En la Fig. 35, se observa que el túnel se ha creado mediante el PCE con la IP del controlador. En este caso, se indica que será gestionado desde el controlador y no desde el dispositivo Cisco IOS XRv, por lo tanto, no se puede modificar el túnel desde este último.

En la implementación de la topología y las pruebas de conectividad, se han logrado

tiempos de respuesta mínimos a pesar de tratarse de un entorno virtualizado. En este contexto, se incluye una captura de tráfico para Segment Routing. En la última parte, se hace referencia a los segmentos que definen la ruta. Con esto, es posible crear diferentes rutas para el túnel.

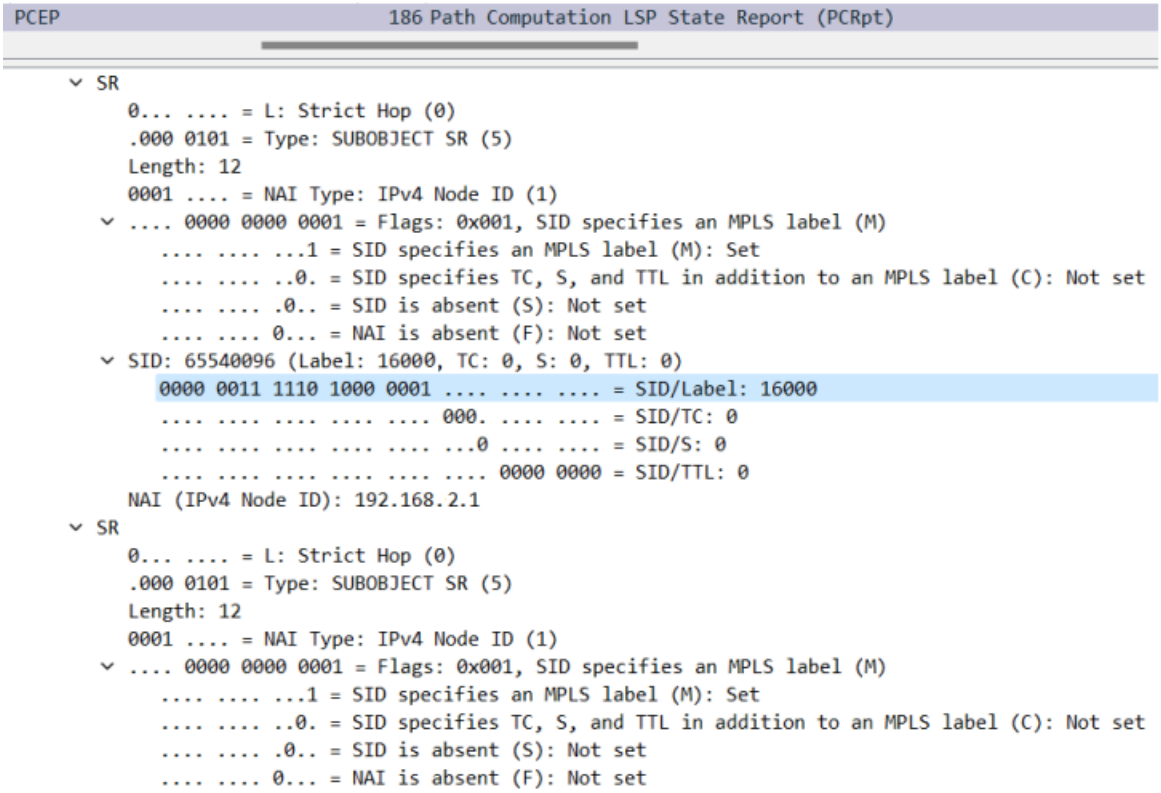
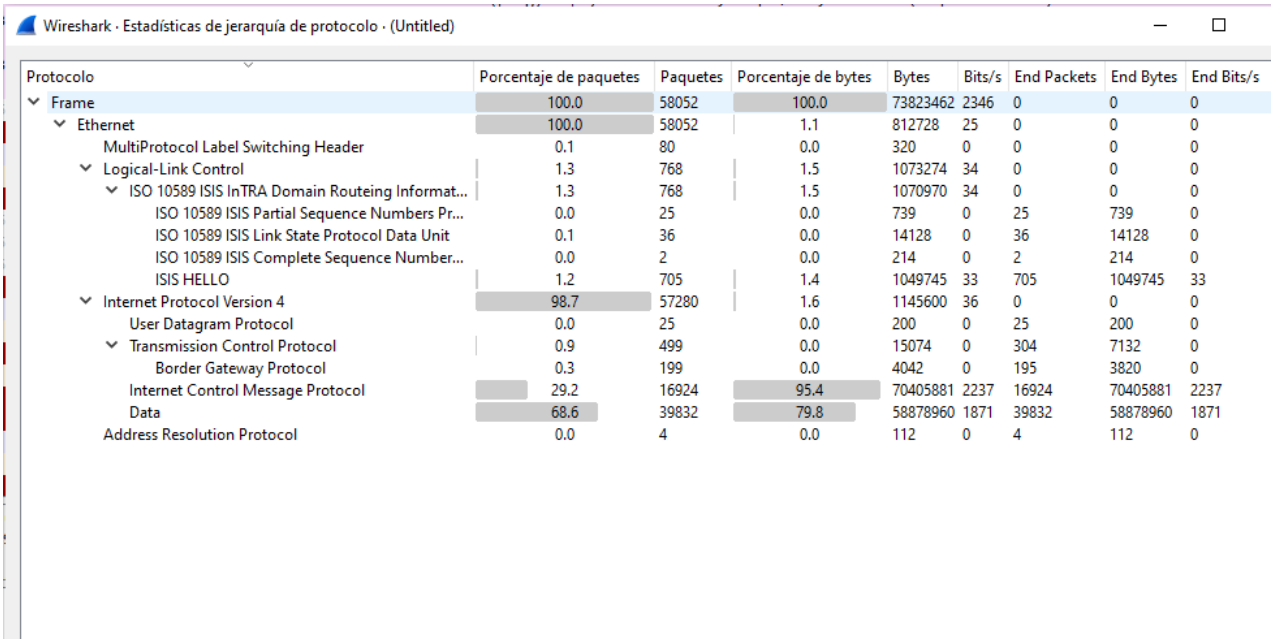


Fig. 35. Comunicación al controlador mediante Segment Routing - Wireshark.

En la Fig. 36, se muestra los resultados muestran una diversidad de protocolos presentes en el tráfico de la red. La mayoría de los paquetes pertenecen al protocolo ICMP, representando casi el 99,36% del total, seguido por el protocolo de datos, que constituye alrededor del 69,88%. Otros protocolos, como Ethernet, ISIS HELLO, UDP, BGP, MPLS y TCP, también están presentes en menor medida. Destaca la presencia de paquetes IPv4, aunque en una proporción relativamente baja.

La velocidad de transmisión varía entre los diferentes protocolos, siendo ICMP el más notable con una velocidad de aproximadamente 469,403 bits/s.

En la Fig. 37, muestra las estadísticas detalladas sobre las longitudes de paquetes observadas. Se divide en diferentes rangos de longitud de paquete, desde 0-19 hasta 5120 y superior.



Protocolo	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	58052	100.0	73823462	2346	0	0	0
Ethernet	100.0	58052	1.1	812728	25	0	0	0
MultiProtocol Label Switching Header	0.1	80	0.0	320	0	0	0	0
Logical-Link Control	1.3	768	1.5	1073274	34	0	0	0
ISO 10589 ISIS InTRA Domain Routeing Informat...	1.3	768	1.5	1070970	34	0	0	0
ISO 10589 ISIS Partial Sequence Numbers Pr...	0.0	25	0.0	739	0	25	739	0
ISO 10589 ISIS Link State Protocol Data Unit	0.1	36	0.0	14128	0	36	14128	0
ISO 10589 ISIS Complete Sequence Number...	0.0	2	0.0	214	0	2	214	0
ISIS HELLO	1.2	705	1.4	1049745	33	705	1049745	33
Internet Protocol Version 4	98.7	57280	1.6	1145600	36	0	0	0
User Datagram Protocol	0.0	25	0.0	200	0	25	200	0
Transmission Control Protocol	0.9	499	0.0	15074	0	304	7132	0
Border Gateway Protocol	0.3	199	0.0	4042	0	195	3820	0
Internet Control Message Protocol	29.2	16924	95.4	70405881	2237	16924	70405881	2237
Data	68.6	39832	79.8	58878960	1871	39832	58878960	1871
Address Resolution Protocol	0.0	4	0.0	112	0	4	112	0

Fig. 36. Jerarquía de protocolos.

Para cada rango, se proporciona el número de paquetes observados, la longitud promedio, la longitud mínima y máxima, así como la tasa de llegada de paquetes y la tasa de ráfaga.

La información proporcionada permite comprender la distribución de las longitudes de los paquetes de la red, así como identificar posibles fallos en el tráfico de red. Por ejemplo, se observa que la mayoría de los paquetes tienen longitudes en el rango de 1280-2559, lo que representa aproximadamente el 74,42% del total de paquetes observados.

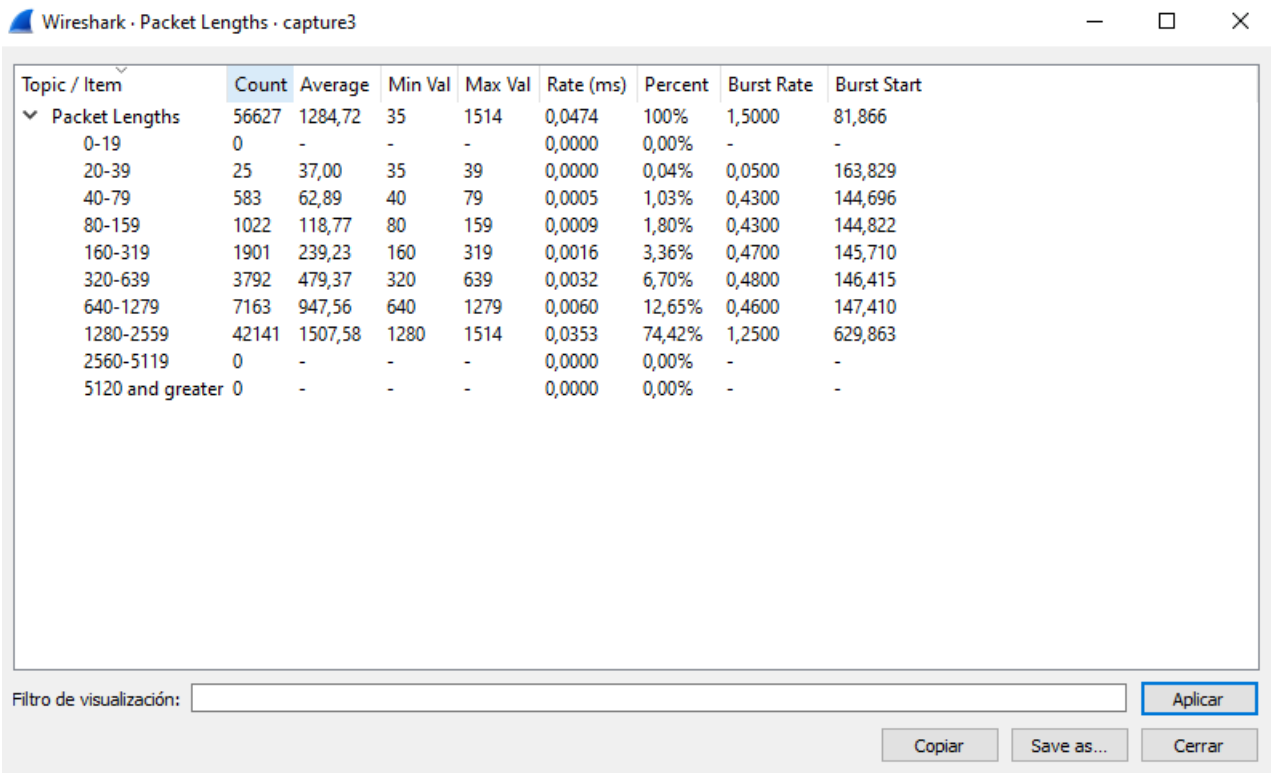


Fig. 37. Longitud de paquetes de envío.

En la Fig. 38, se observa la gráfica E/S donde muestra el rendimiento de la red a través del filtro de protocolo IS-IS, se está registrando el valor en cada intervalo de 1seg. Presenta una representación visual del comportamiento de la protocolo IS-IS que se define como variable, a lo largo del tiempo.

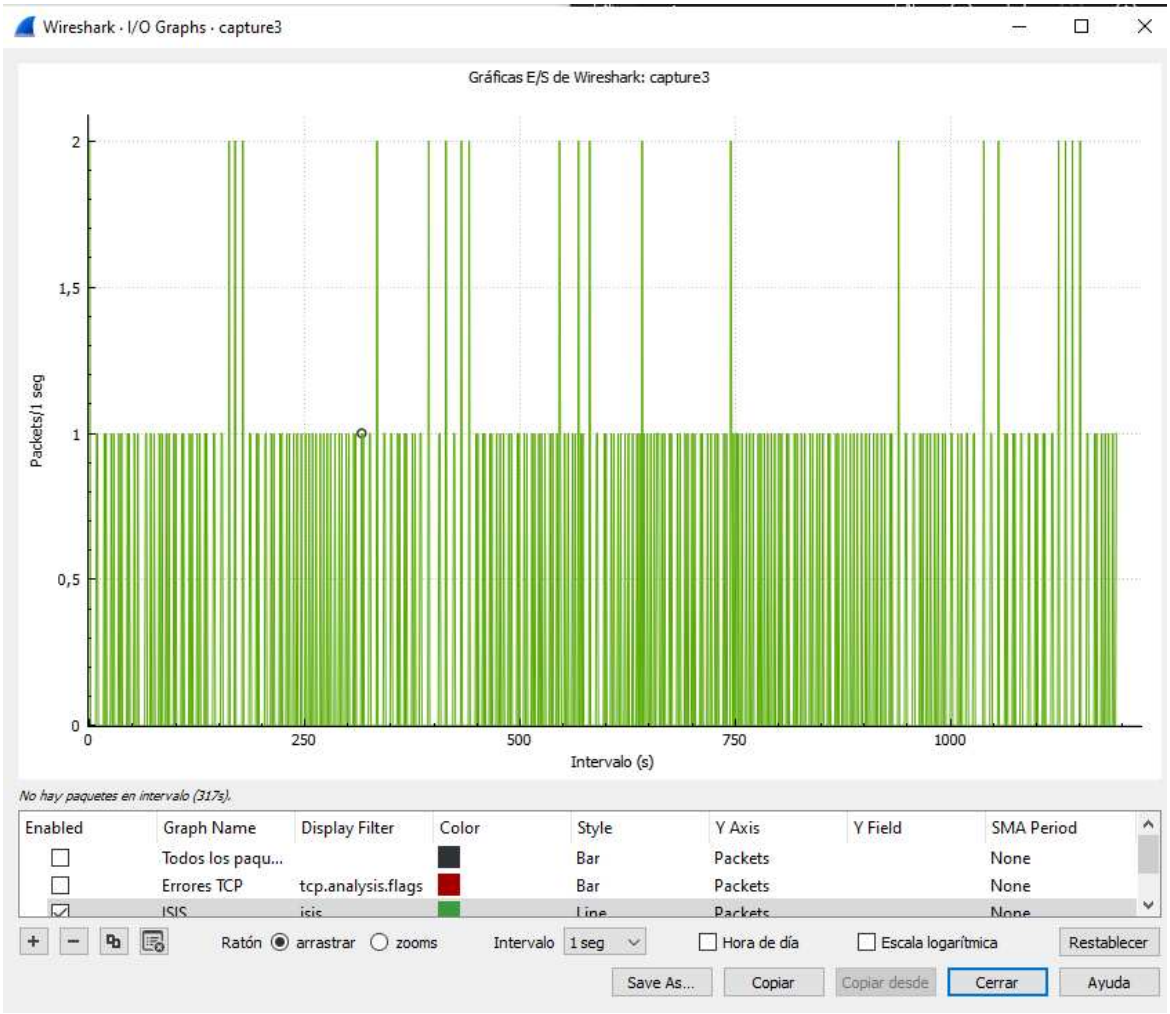


Fig. 38. Gráfica E/S rendimiento.

En la Fig. 39, se observa el estado del entorno en el que se trabajó. Se puede notar que al iniciar la simulación, se obtiene un alto consumo de recursos.

Recursos de Arranque para OpenDaylight en el Contenedor de Ubuntu:

Requisitos Mínimos:

- CPU: 1 núcleo
- RAM: 2 GB
- Almacenamiento: 5 GB de espacio disponible en disco

Requisitos Recomendados:

- CPU: 2 núcleos o más
- RAM: 4 GB o más

- Almacenamiento: 10 GB de espacio disponible en disco

Recursos de Arranque para Routers Cisco IOS XRv:

Requisitos Mínimos:

- CPU: 1 núcleo
- RAM: 2 GB
- Almacenamiento: 10 GB de espacio disponible en disco

Requisitos Recomendados:

- CPU: 2 núcleos o más
- RAM: 4 GB o más
- Almacenamiento: 20 GB de espacio disponible en disco

Fig. 39. Estado del entorno de trabajo

V. Conclusiones y recomendaciones

A. *Conclusiones*

La combinación de Redes Definidas por Software y Segment Routing ha demostrado ser ventaja para la gestión y el control de redes avanzadas. A pesar de estos beneficios, reconocemos ciertos obstáculos, como la necesidad de infraestructura compatible y una planificación cuidadosa de políticas de enrutamiento. Sin embargo, la integración de SDN y Segment Routing se propone como una solución sólida y adaptable para mejorar la eficiencia, seguridad y agilidad de las redes de comunicaciones en entornos empresariales.

Segment Routing representa una nueva forma de utilizar MPLS con menos protocolos involucrados. Ahora trabaja con el Protocolo de Enrutamiento Interior (IGP), que se encarga de la distribución de etiquetas. Con cada router conociendo todas las etiquetas, se crea una lista de segmentos que permite un mayor control sobre la ruta del tráfico en toda la red.

Se concluye que se han cumplido parcialmente los objetivos planteados. La implementación de la tecnología Segment Routing ha proporcionado resultados satisfactorios en cuanto a la evaluación del rendimiento. Sin embargo, la separación del plano de control con el plano de datos no se logra completamente, dado que al utilizar el demo del router Cisco IOS XRv, se limita la capacidad de configuración para permitir que el controlador OpenDaylight ejecute su funcionalidad total.

En este contexto, la comunicación entre el controlador y los dispositivos se realiza a través del protocolo NETCONF, el cual permite una administración eficiente de las configuraciones y el estado de los dispositivos de red, así como un transporte seguro de mensajes entre un cliente y un servidor.

En cuanto al rendimiento, se concluye que la red simulada está diseñada para una escala media a grande. Las pruebas realizadas definieron que el throughput varía en función del envío de paquetes y el tiempo, y este puede verse afectado por múltiples factores, incluyendo la congestión de la red y la carga del dispositivo. Además, se observa que permite que los flujos de tráfico utilicen los enlaces de red a su capacidad máxima, lo que contribuye a mejorar el rendimiento de la red y puede conducir a una simplificación en la distribución de los elementos hardware y una reducción en el tiempo de procesamiento.

Durante la implementación, se enfrentaron diversos desafíos, desde la captura de paquetes hasta las pruebas de conectividad en la infraestructura de red y el controlador OpenDaylight. Se confirmó que el diseño de red propuesto representa una solución efectiva para mejorar el rendimiento en entornos empresariales, donde la escalabilidad, la eficiencia y la optimización son elementos clave.

Además, se subraya la importancia crítica de seleccionar el controlador adecuado para la implementación. Se optó por utilizar el controlador OpenDaylight debido a su capacidad para acceder a través del protocolo NETCONF. Sin embargo, durante el desarrollo, se observaron limitaciones en el control que este ofrece sobre la administración de la red. Esto resalta la necesidad de una cuidadosa consideración al elegir el controlador, ya que su elección influirá significativamente en la interacción con los dispositivos de red y la efectividad general de la gestión y el control del sistema.

B. Recomendaciones

La integración de SDN y Segment Routing ha demostrado ser una ventaja significativa para la gestión y el control de redes avanzadas en entornos empresariales. Aunque se reconocen obstáculos como la necesidad de infraestructura compatible y una planificación cuidadosa de políticas de enrutamiento, esta combinación se presenta como una solución sólida y adaptable para mejorar la eficiencia, seguridad y agilidad de las redes de comunicaciones. La capacidad de Segment Routing para utilizar MPLS con menos protocolos involucrados, combinada con las capacidades de gestión y control proporcionadas por SDN, ofrece un enfoque prometedor para abordar los desafíos de las redes empresariales modernas.

La implementación exitosa de la tecnología SDN y Segment Routing requiere una planificación detallada y una investigación exhaustiva de los recursos necesarios, los dispositivos y los protocolos. La identificación de los recursos mínimos necesarios, incluso en entornos virtuales, es crucial para garantizar un rendimiento óptimo de la red. Además, se destaca la importancia de investigar y comprender completamente la documentación oficial de cada protocolo a implementar, así como mantenerse actualizado con las nuevas actualizaciones y estándares, lo que garantiza una implementación sin contratiempos y compatible con los dispositivos y herramientas utilizadas.

Una vez implementada la tecnología SDN y Segment Routing, es fundamental establecer un proceso de monitoreo continuo y optimización para evaluar el rendimiento de la red y realizar ajustes según sea necesario. Esto incluye la optimización de políticas de enrutamiento, la identificación y resolución de cuellos de botella de rendimiento y la adaptación a cambios en las necesidades de la red empresarial. Un enfoque proactivo para el monitoreo y la optimización garantizará que la red pueda mantenerse al día con las demandas cambiantes y siga funcionando de manera eficiente y segura en todo momento.

VI. Referencias

[1] L. Luo, H. Yu, S. Luo, M. Zhang y S. Yu, «Achieving Fast and Lightweight SDN Updates with Segment Routing,» en *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, págs. 1-6. DOI: 10.1109/GLOCOM.2016.7841562.

[2] Dirección: https://dspace.ups.edu.ec/bitstream/123456789/25927/4/UPS-GT004557.pdf.

[3] A. Kos, G. Maier y Alberto, *SEGMENT ROUTING PRINCIPLES AND APPLICATIONS FOR SDN*. 8165. dirección: https://www.politesi.polimi.it/retrieve/a81cb05b-c2e5-616b-e053-1605fe0a889a/AnaKosThesisFinal.pdf.

[4] Z. Li, L. Huang, H. Xu y G. Zhao, «Segment routing in hybrid software-defined networking,» en *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, 2017, págs. 160-165. DOI: 10.1109/ICCSN.2017.8230098.

[5] G. Maila, I. Marius y C. Victor, «Segment Routing,» en *2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, 2017, págs. 34-38. DOI: 10.1109/ATEE.2017.7905138.

[6] D. Nuñez-Agurto, W. Fuertes, L. Marrone y M. Macas, *Machine Learning-Based Traffic Classification in Software-Defined Networking: A Systematic Literature Review, Challenges, and Future Research Directions*. dirección: https://www.iaeng.org/IJCS/issues_v49/issue_4/IJCS_49_4_04.pdf.

[7] D. Nunez-Agurto, W. Fuertes, L. Marrone y M. Macas, «Machine Learning-Based Traffic Classification in Software-Defined Networking: A Systematic Literature Review, Challenges, and Future Research Directions.,» *IAENG International Journal of Computer Science*, vol. 49, nº 4, 2022.

[8] A. Pradhan y R. Mathew, «Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN),» *Procedia Computer Science*, vol. 171, págs. 2581-2589, 2020, Third International Conference on Computing and Network Communications (CoCoNet'19), ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.04.280. dirección: https://www.sciencedirect.com/science/article/pii/S1877050920312734.

[9] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano y S. Salsano, «Traffic Engineering with Segment Routing: SDN-Based Architectural Design and Open Source Implementation,»

2015 Fourth European Workshop on Software Defined Networks, págs. 111-112, 2015. dirección: https://api.semanticscholar.org/CorpusID:2953634.

[10] D. Todorov, H. Valchanov y V. Aleksieva, «Load Balancing model based on Machine Learning and Segment Routing in SDN,» en *2020 International Conference Automatics and Informatics (ICAI)*, oct. de 2020, págs. 1-4. DOI: 10.1109/ICAI50593.2020.9311385.

[11] Z. N. Abdullah, I. Ahmad e I. Hussain, «Segment Routing in Software Defined Networks: A Survey,» *IEEE Communications Surveys Tutorials*, vol. 21, nº 1, págs. 464-486, 2019. DOI: 10.1109/COMST.2018.2869754.

[12] O. M. Mon y M. T. Mon, «Quality of Service Sensitive Routing for Software Defined Network Using Segment Routing,» en *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, 2018, págs. 180-185. DOI: 10.1109/ISCIT.2018.8587944.

[13] N. Figuerola, «SDN–Redes definidas por Software,» *Recuperado de https://d1wqtxts1xzle7. cloudfront. net/35217763/SDN. pdf*, vol. 1413880566, 2013.

[14] U. Distrital, F. Jose, J. Guevara, C. Alexis y C. Fajardo, *Arquitectura y funcionamiento de redes definidas por software (SDN)*. dirección: https://repository.udistrital.edu.co/bitstream/handle/11349/29727/CasilimasFajardoCarlosAlexis2022.pdf?sequence=2&isAllowed=y.

[15] P. L. Ventre, M. M. Tajiki, S. Salsano y C. Filsfils, «SDN Architecture and Southbound APIs for IPv6 Segment Routing Enabled Wide Area Networks,» *IEEE Transactions on Network and Service Management*, vol. 15, nº 4, págs. 1378-1392, 2018. DOI: 10.1109/TNSM.2018.2876251.

[16] F. Hu, Q. Hao y K. Bao, «A survey on software-defined network and openflow: From concept to implementation,» *IEEE Communications Surveys & Tutorials*, vol. 16, nº 4, págs. 2181-2206, 2014.

[17] A. Centeno, C. M. Rodriguez Vergel, C. Anías Calderón, F. Camilo, C. Bondarenko y Uci, «Controladores SDN, elementos para su selección y evaluación,» *Telemática*, vol. 13, págs. 10-20, nov. de 2014.

[18] 2017. dirección: https://blog.51cto.com/u_8493144/1970670.

[19] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini y P. Castoldi, «SDN and PCE implementations for segment routing,» en *2015 20th European Conference on Networks and Optical Communications - (NOC)*, 2015, págs. 1-4. DOI: 10.1109/NOC.2015.7238607.

[20] I. Technologies, *Traffic Engineering using Segment Routing (BGP-LS/PCEP)*, ene. de 2017. dirección: https://www.youtube.com/watch?app=desktop&v=2GZ2C0nZbFs.

[21] 2023. dirección: https://docs.opendaylight.org/projects/openflowplugin/en/latest/users/architecture.html.

[22] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski y R. Shakir, «Segment routing architecture,» inf. téc., 2018.

[23] J. Pang, G. Xu y X. Fu, «SDN-Based Data Center Networking With Collaboration of Multipath TCP and Segment Routing,» *IEEE Access*, vol. 5, págs. 9764-9773, 2017. DOI: 10.1109/ACCESS.2017.2700867.

[24] 2019. dirección: https://www.researchgate.net/figure/Segment-routing-example_fig1_335134481.

[25] R. A. Ríos, «Conceptualización de SDN y NFV,» *Maskay*, vol. 6, nº 1, págs. 29-34, 2016.

[26] M. K. Porwal, A. Yadav y S. Charhate, «Traffic Analysis of MPLS and Non MPLS Network including MPLS Signaling Protocols and Traffic Distribution in OSPF and MPLS,» en *2008 First International Conference on Emerging Trends in Engineering and Technology*, jul. de 2008, págs. 187-192. DOI: 10.1109/ICETET.2008.58.

[27] Dirección: https://www.cisco.com/en/US/docs/ios_xr_sw/ios_xrv/install_config/b_xrvr_432_chapter_01.html.

[28] 2016. dirección: https://docs.gns3.com/docs/.

[29] 2016. dirección: https://www.wireshark.org/docs/.

[30] 2024. dirección: https://blog.invgate.com/hubfs/que-es-wireshark%20%281%29.jpg.

[31] A. Peña, D. Fabricio, I. De et al., *FACULTAD DE INGENIERÍA CIVIL CARRERA DE INGENIERÍA DE SISTEMAS MACHALA 2021 METODOLOGÍA PPDIOO APLICADA A PYMES*. dirección: http://repositorio.utmachala.edu.ec/bitstream/48000/16854/1/TTFIC-2021-IS-DE-00001.pdf.

[32] 2018. dirección: https://www.studocu.com/ec/document/universidad-de-guayaquil/diseno-de-redes-i/ppdioo-explicacion-de-las-fases/3449505.

[33] A. Darwin, A. Sánchez, A. Santiago, G. Gallo, J. Domínguez y A. Quito, *UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO CARRERA DE TELECOMUNICACIONES DISEÑO DE FRONTERA DE RED EN SOFTWARE LIBRE PARA LA EMPRESA ABSORPELSA BAJO LA METODOLOGÍA Y ARQUITECTURA DE SEGURIDAD CISCO SAFE.*

Trabajo de titulación previo a la obtención del Título de Ingeniero en Telecomunicaciones. 2022. dirección: https://dspace.ups.edu.ec/bitstream/123456789/23288/1/UPS%20-%20TTS931.pdf.

[34] 2016. dirección: https://docs.opendaylight.org/projects/bgpcep/en/latest/bgp/bgp-user-guide-running-bgp.html.

[35] 2016. dirección: https://docs.opendaylight.org/en/latest/getting-started-guide/installing_opendaylight.html.

[36] Mar. de 2019. dirección: https://letsnetworking.wordpress.com/2019/03/12/vrf-based-path-selection-on-cisco-ios-xr/.