

Desempeño de redes SD-WAN vs. MPLS: evaluación en diferentes escenarios

Performance of SD-WAN vs. MPLS networks: evaluation in different scenarios

Santiago Meneses Narvaez¹ , Emilio Zhuma Mera¹ , Ángel Torres Quijije¹ , Marco Chimbo Fogacho¹ 

¹ Universidad Técnica Estatal de Quevedo, Quevedo, Ecuador

Enviado: 03/04/2025, Recibido: 29/08/2025, Publicado: 30/01/2026

Autor de correspondencia:

Santiago Meneses Narvaez: smenesesn@uteq.edu.ec

DOI: [10.53358/ideas.v8i1.1238](https://doi.org/10.53358/ideas.v8i1.1238)



PALABRAS CLAVE

Redes,
análisis,
MPLS,
SD-WAN,
métrica,
GNS3,
calidad de servicio.

RESUMEN

Las redes definidas por software (SDN) han llegado para flexibilizar y agilizar los entornos de red, permitiendo la administración mediante la separación del plano de control y el plano de reenvío, teniendo la capacidad de adaptar una red rápidamente a entornos cambiantes. Por su parte, las redes de Comunicación Multiprotocolo por Etiqueta (MPLS) gestionan el tráfico mediante un mecanismo de conmutación de etiquetas, lo que les confiere ventajas clave en entornos empresariales. Esta investigación analiza comparativamente el rendimiento de las redes de área amplia definidas por software (SD-WAN) y las redes de Comunicación Multiprotocolo por Etiqueta (MPLS) centrándose en evaluar parámetros críticos como latencia, variación de retardo (jitter), ancho de banda y pérdida de paquetes. Para el análisis se ha planteado diferentes escenarios de simulación, a los cuales se les ha generado una carga controlada de tráfico, los resultados de cada carga generada se envian a una base de datos, a su vez esta base de datos conecta con grafana, que permite la fácil visualización de datos mediante gráficos.

KEYWORDS

Networks,
analysis,
quality of service,
MPLS,
SD-WAN,
metrics,
GNS3.

ABSTRACT

Software-Defined Networks (SDN) have emerged to enhance flexibility and agility in network environments by enabling management through the separation of the control plane and the forwarding plane. This separation allows networks to quickly adapt to changing environments. Meanwhile, Multiprotocol Label Switching (MPLS) networks manage traffic using a label-switching mechanism, providing key advantages in enterprise environments. This research comparatively analyzes the performance of Software-Defined Wide Area Networks (SD-WAN) and Multiprotocol Label Switching (MPLS) networks, focusing on critical parameters such as latency, jitter, bandwidth, and packet loss. For the analysis, different simulation scenarios have been designed, each subjected to a controlled traffic load. The results of each generated load are sent to a database, which is then connected to Grafana, allowing for easy data visualization through graphs.

1. Introducción

La innovación en servicios digitales, así como la facilidad de acceso a ellos, ha permitido una generación exponencial de datos e información provocando sobrecarga en las redes [1]. En la actualidad, existe una gran variedad de aplicaciones, las cuales requieren una gran cantidad de recursos en equipos de red al generar simultáneamente mucho tráfico, por lo que el ancho de banda debe utilizarse de forma eficiente [2].

En las últimas décadas se ha visto un incremento significativo en servicios de tiempo real, por lo que los proveedores de servicios de internet han tenido que tomar medidas para brindar un servicio confiable y de calidad. Se han realizado una serie de investigaciones para el desarrollo de diferentes mecanismos y técnicas que permitan el uso eficiente de los elementos de red, minimizando la pérdida y retraso de paquetes [3]. MPLS (Comutación de Etiquetas Multiprotocolo) ha sido la solución de muchas interrogantes, ya que es una tecnología que ofrece la posibilidad de transferencia de información a altas velocidades [2], por lo que la actual solución integrada de enrutamiento IP/comutación es una de las mejores aplicaciones de la tecnología MPLS [4].

En MPLS, las sucursales zonales se conectan a los centros de datos mediante enrutadores. El tráfico de origen a destino se envía y recibe mediante métodos de direccionamiento TCP/IP [5]. Esta arquitectura es ideal para redirigir el tráfico desde y hacia los centros de datos a las sucursales zonales. Con la evolución de la computación en la nube, el patrón de tráfico cambió por completo y el modelo MPLS no era compatible con el nuevo patrón de tráfico basado en aplicaciones en la nube [2].

Los métodos de enrutamiento convencionales presentan numerosos desafíos prácticos. La arquitectura SD-WAN disocia la gestión y el control de la red del hardware físico, lo que permite un plano de control centralizado basado en software [6], dando como resultado una arquitectura con tres componentes clave como el SD-WAN Edge, SD-WAN Controller, y SD-WAN Orchestrator como se ve en la Figura 1.

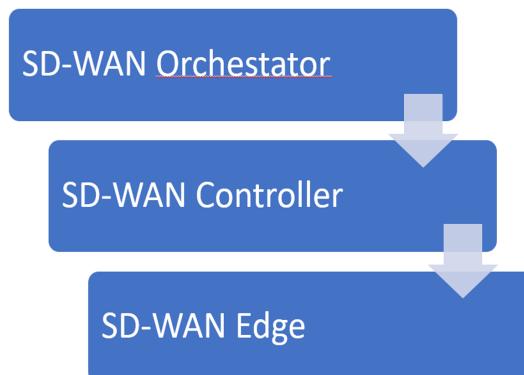


Figura 1: Arquitectura SD-WAN.

Esta red virtualizada opera como una capa adicional sobre hardware de bajo costo, mientras que controladores centralizados gestionan el plano de control. Su infraestructura IP virtual segura permite la segmentación del tráfico, el enrutamiento y la difusión de rutas. Además, incorpora autenticación y cifrado para garantizar la seguridad de los datos del usuario [7, 8]. La combinación del plano de control, el plano de datos y la estructura IP virtual proporciona una red robusta, segura y escalable [9, 10].

SD-WAN se aplica ampliamente para la interconexión de redes en varios escenarios, ISP, interconexión de centros de datos, soluciones en la nube e internet de las cosas [11]. En comparación con WAN tradicionales, SDWAN es superior y más económico, teniendo una implementación, administración y mantenimiento más ágil [11, 12].

Este trabajo está estructurado en cuatro secciones. En la segunda sección, se describen las herramientas y redes utilizadas en la simulación, junto con el análisis correspondiente a cada elección. En la Figura 2 se destaca la secuencia del flujo de información y el desarrollo de la implementación: una vez que las topologías estén operativas, se utilizará Python para generar una carga de tráfico en las redes. Los resultados obtenidos se almacenarán en una base de datos que es influx-db, la cual estará vinculada a una herramienta que facilitará su visualización de manera sencilla como lo es grafana.

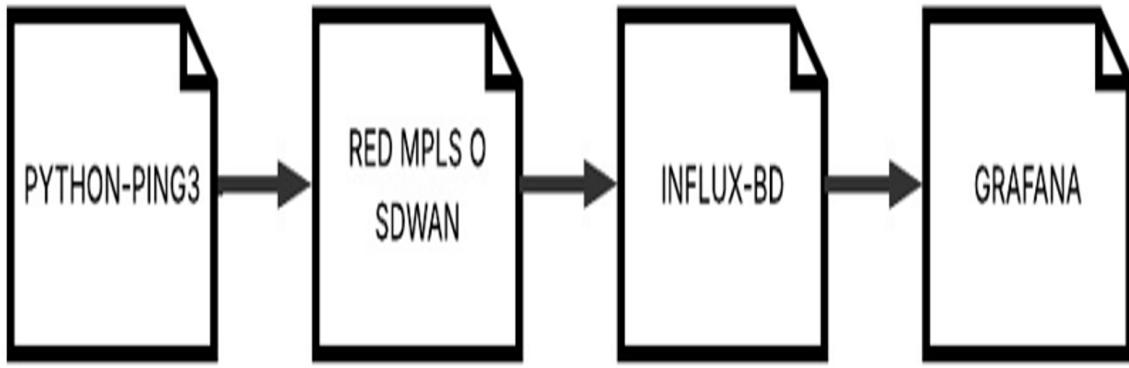


Figura 2: Flujo de datos en la simulación.

En la sección tres se analizan los resultados obtenidos de las pruebas realizadas para cada topología de red, con el respaldo de los gráficos generados en grafana, y por último en la etapa final que corresponde a la cuarta sección, se tiene la sección de bibliografía que respalda el desarrollo de este trabajo.

2. Metodología

2.1. Metodología de investigación

El presente trabajo utiliza el método inductivo que se basa en la observación y experimentación, construyendo una comprensión sólida de los conceptos involucrados, también se usa el método analítico para entender las diferentes métricas de rendimiento, y el método experimental, donde con el uso de diferentes herramientas se establecieron las configuraciones necesarias en los ambientes de simulación.

2.2. Herramientas

Emuladores. Se realizó un análisis detallado de diferentes software para simulación y emulación de redes de datos, teniendo como principales opciones cisco packet tracer, el cual es un simulador; tenemos gns3, eve-ng y mininet, que son emuladores, entendiéndose que un emulador lo que hace es ejecutar una copia exacta de un Sistema operativo real, permitiendo tener resultados cercanos a entornos de implementación, como se ve en la Tabla 1.

Tabla 1: Comparación de software para simulación y emulación.

Características	Cisco Packet Tracert	GNS3	EVE-NG	Mininet
Simulador	Si	Si	Si	Si
Emulador	No	Si	Si	No
Open Source	No	Si	Si	Si
Interfaz de usuario (GUI)	Si	Si	Si	No
Multiplataforma	No	Si	Si	Si
Programabilidad	No	Si	Si	Si
Controladores admitidos	No	Si	Si	Si
Herramientas de tráfico	No	Si	Si	Si
Escalabilidad	No	Si	Si	Si
Compatibilidad con Dynamips	No	Si	Si	No
Complejidad	Bajo	Media	Alto	Alto
Programabilidad	No	Si	Si	Si
Documentación	Si	Si	Si	Si
Requiere Máquina Virtual	No	Si	Si	No
Memoria RAM	4 Gb	4 Gb	4 Gb	2 Gb

GNS3 es un software (Graphical Network Simulator-3), un simulador que permite diseñar, construir y emular redes de datos, tiene su notabilidad ya que es una herramienta de software libre, que permite la integración de diferentes marcas de fabricantes, como Cisco, Mikrotik, Fortinet, entre otros [13].

Controladores SD-WAN. Los controladores SD-WAN son elementos que permiten el control, administración y gestión de políticas y servicios. En la Tabla 2 se muestra el resultado de la revisión y aspectos clave en cuanto a estos controladores [14].

Tabla 2: Controladores SD-WAN

Características	Cisco vManage	Cisco Meraki	VMware Velo-cloud SD-WAN	Juniper SD-WAN	Contrail	Fortimanager
Proveedor	Cisco	Cisco	VMware	Juniper Networks	Fortinet	
Interfaz de usuario	Web	Web	Web	Web, CLI	Web, CLI	
API	Restful	Restful	Restful	Restful	Restful	
Lenguaje	Js, Python	Js	Js, Python	Java, Python	C++, Python	
Multiplataforma	Si	Si	Si	Si	Si	
Seguridad	Si	Si	Si	Si	Si	
Ánalisis y reporte	Si	Si	Si	Si	Si	
Consumo de recursos	Alto	Bajo	Medio	Alto	Medio	
Componentes Adicional	Cisco Vbond, Vedge, Vsmart	No	Orchestrator, Gateway	Gateway, Orchestrator	Fortigate, Fortianalyzer	
Dificultad de configuración	Alto	Bajo	Medio	Medio	Medio	
Virtualización	Si	No	Si	Si	Si	

Fortimanager con el Sistema operativo FortiOs, es la herramienta que permite la administración de forma centralizada de productos Fortinet, siendo una elección idónea por su documentación detallada y la capacidad de operar sin la necesidad del controlador Fortimanager.

MPLS: Existen varios fabricantes de equipos que trabajan con el protocolo MPLS, además de ello permiten la simulación de sus sistemas operativos con software como GNS3. A continuación, se detalla en Tabla 3 algunos fabricantes que trabajan con MPLS.

Tabla 3: Fabricantes de equipos con MPLS.

Características	Cisco IOS XR	Juniper Junos	Mikrotik RouterOS	Huawei (VRP)
Proveedor	Cisco	Juniper	Mikrotik	Huawei
GUI	Si	Si	Si	Si
Multiplataforma	Si	Si	Si	Si
Escalabilidad	Alta	Alta	Alta	Alta
Soporte de scripts	Python	Python	Python, Winbox scripts	Python, Shell
Documentación	Extensa, profesional	Detallada y estructurada	Amplia y fácil de entender	Amplia y detallada
Facilidad de uso	Alta y requiere conocimiento especializado	Alta y familiaridad con Junos	Alta e intuitiva con Winbox	Alta y requiere conocimiento Huawei
Memoria	16GB	512MB	352MB	2Gb
Componentes adicionales	Si	No	No	No
Virtualización	Si	Si	Si	Si

Mikrotik RouterOS permite la fácil integración con simuladores como GNS3, además su fácil integración con la herramienta Winbox, que permite la configuración de manera gráfica, haciendo de esto una solución llamativa para el presente trabajo.

Bases de datos: La base de datos es un punto muy central en el presente trabajo, para lo cual se consideraron algunas soluciones para este tema, teniendo a influxDB como la solución más apropiada, por ser una base de serie temporal con capacidad de gestionar y almacenar datos de alta frecuencia de manera eficiente, como son las métricas de monitoreo obtenidas. Se presenta en la Tabla 4 una comparativa relacionada con bases de datos.

Tabla 4: Base de datos.

Características	Relacional	No relacional	Serie Temporal
Modelos de datos	Ordenados por filas, columnas y tablas	Clave-valor, documentos	Datos ordenados por tiempo
Esquema	Estructura rígida y predefinida	Flexible y dinámico	Optimizado para datos temporales
Lenguaje de consulta	SQL (Structured Query Language)	Diferentes lenguajes de consulta	SQL para serie temporal o lenguajes específicos de consulta (InfluxDB)
Escalabilidad	Escalabilidad vertical	Escalabilidad vertical	Escalabilidad horizontal y optimizada
Transacciones	Soporte completo (ACID)	Varia en el uso de transacciones	Optimizado para inserciones rápidas
Usos Típicos	Aplicaciones empresariales, sistemas de gestión de datos	Aplicaciones web, grandes volúmenes de datos no estructurados o semiestructurados	Ánálisis de series temporales, IOT, finanzas, ciencia de datos
Ejemplos de SGBD	MySQL, Oracle, PostgreSQL	MongoDB, Casandra, Redis	InfluxDB, Prometheus, TimeScaleDB

Visualización de datos: En el análisis de una herramienta completa para el monitoreo y visualización de la red SD-WAN y MPLS, se identificaron algunas opciones como las que se observan en la Tabla 5, donde se determina que Grafana proporciona una visualización de grandes cantidades de datos, esto sumado a la integración con diversas herramientas como InfluxDB y Python, lo hacen ideal para el propósito de este trabajo.

Tabla 5: Software de monitoreo

Características	Zabbix	Grafana	Nagios
Función principal	Monitoreo y recolección de datos de redes, servidores y aplicaciones	Visualización y análisis de datos de monitoreo	Monitoreo y recolección de datos de sistemas, redes y aplicaciones
Tipo de software	Server y agente	Servidor	Servidor
Interfaz de usuario	Web y CLI	Web	Web
Escalabilidad	Si	Si	Si
Protocolo de comunicación	SNMP, IPMI, JMX, SMTP	HTTP y API	SNMP, HTTP, SMTP, FTP
Integración con otras herramientas	Si	Si	Si
Multiplataforma	Si	Si	Si
Documentación	Si	Si	Si
Memoria RAM	8Gb	4 Gb	1 Gb

2.3. Diseño de escenarios

Se plantearon tres escenarios de prueba en el presente trabajo, en cada escenario se implementó MPLS con SD-WAN, las herramientas usadas para la simulación fueron descritas en las tablas anteriores, la implementación fue configurada bajo parámetros específicos y así poder evaluar el comportamiento de las redes bajo una carga de tráfico determinada.

Para los diferentes escenarios de la red MPLS los equipos etiquetados con “LSR” y “LER” son enrutadores y la etiqueta “TSBD” es la base de datos de serie temporal. Por otro lado, para los escenarios SD-WAN, la etiqueta “Zona” y “Central” hace referencia a enrutadores y finalmente en la red mixta SD-WAN y MPLS, el equipo “ESW” es un switch Ethernet de capa 3 para interconectar segmentos de red.

Topología Estrella MPLS: La presente topología consta de 5 enrutadores, 4 switch, 2 navegadores y 1vpc, adicional 1 máquina virtual con Linux para la carga de tráfico; cada enrutador tiene MPLS funcionando, también está presente la base de datos con grafana. En las Figuras 3 y 4 se muestra la topología y un fragmento de las configuraciones realizadas.

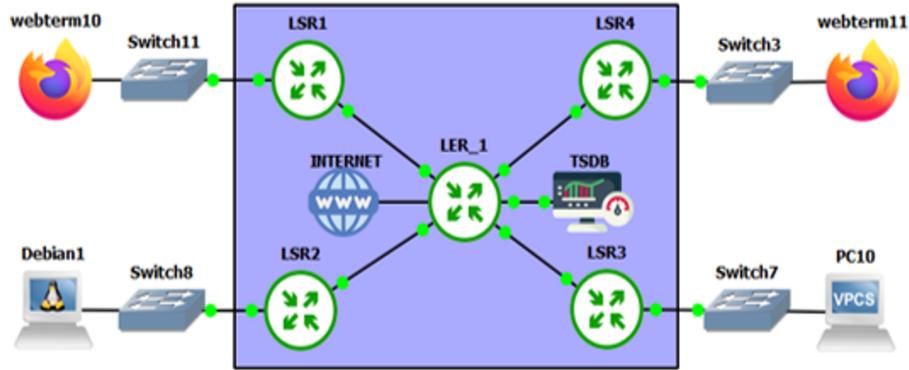


Figura 3: Topología MPLS en estrella

```

[admin@LSR2] > ip ad print
Flags: X - disabled, I - invalid, D - dynamic
#   ADDRESS           NETWORK      INTERFACE
0   192.168.10.12/24  192.168.10.0  Admin_ether2
1   20.3.3.3/32        20.3.3.3    Loopback2
2   178.168.20.2/24   178.168.20.0  LER_ether5
3   190.168.20.1/24   190.168.20.0  LAN_ether9

[admin@LSR2] >
[admin@LSR2] > mpls interface print
Flags: X - disabled, * - default
#   INTERFACE
0   * all
1   ether1
2   LER_ether5
3   Loopback2
MPLS-MTU
1508
1508
1508
1508

[admin@LSR2] > mpls ldp print
enabled: yes
local-id: 20.3.3.3
transport-address: 20.3.3.3
path-vector-limit: 255
hop-limit: 255
loop-detect: yes
use-explicit-null: no
distribute-for-default-route: no
[admin@LSR2] > mpls ldp neighbor print
Flags: X - disabled, D - dynamic, O - operational, T - sending-targeted-hello,
V - vpls
#   TRANSPORT      LOCAL-TRANSPORT PEER          SEN
V  DO  20.1.1.1  20.3.3.3          20.1.1.1:0  no

[admin@LSR2] >

```

Figura 4: Interfaces agregadas al protocolo MPLS y LDP del nodo LSR2

Topología malla MPLS. Este escenario cuenta con cinco enrutadores, 4 switch, un navegador tipo webterm, 3vpc, y una máquina virtual Linux, la variante de la topología anterior es el número de conexiones de cada router como se ve en las Figuras 5 y 6.

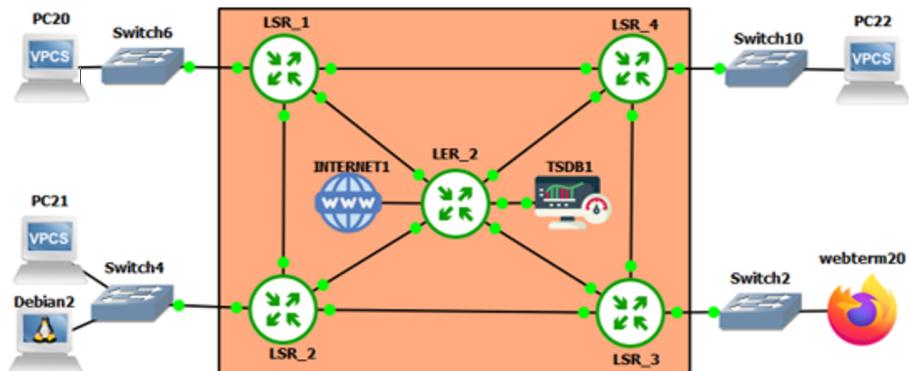


Figura 5: Topología MPLS en malla

MPLS											
LDP Interface		LDP Neighbor		Accept Filter		Advertise Filter		Forwarding Table			
MPLS Interface		Local Bindings		Remote Bindings							
<input type="button" value="Find"/>											
L	In Label / Out Labels	Interface	Nexthop	Destination	Bytes	packets	Hw. Bytes	Hw. Pack...			
expl-null					0 B	0	0 B	0			
L 16		LSR2_ether5	171.168.20.2	191.168.20.0/24	0 B	0	0 B	0			
L 17		LSR1_ether4	171.168.10.2	10.2.2.2	0 B	0	0 B	0			
L 18		LSR4_ether7	171.168.40.2	175.168.30.0/24	0 B	0	0 B	0			
L 19		LSR1_ether4	171.168.10.2	175.168.10.0/24	0 B	0	0 B	0			
L 20		LSR2_ether5	171.168.20.2	10.3.3.3	0 B	0	0 B	0			
L 21		LSR1_ether4	171.168.10.2	191.168.10.0/24	0 B	0	0 B	0			
L 22		LSR4_ether7	171.168.40.2	191.168.40.0/24	0 B	0	0 B	0			
L 23		LSR2_ether5	171.168.20.2	175.168.20.0/24	0 B	0	0 B	0			
L 24		LSR4_ether7	171.168.40.2	175.168.40.0/24	0 B	0	0 B	0			
L 25		LSR4_ether7	171.168.40.2	10.5.5.5	0 B	0	0 B	0			
L 26		LSR3_ether6	171.168.30.2	10.4.4.4	0 B	0	0 B	0			
L 27		LSR3_ether6	171.168.30.2	191.168.30.0/24	0 B	0	0 B	0			
13 items											

Figura 6: Estado de tabla de reenvío del protocolo MPLS en la topología malla

Topología Estrella Red SDWAN: Esta topología consta de 5 enrutadores, 4 switch, 1 navegador web, 2vpc, la base de datos y una máquina virtual Linux. La Figura 7 se muestra el diagrama de la topología, mientras que la Figura 8, las configuraciones realizadas.

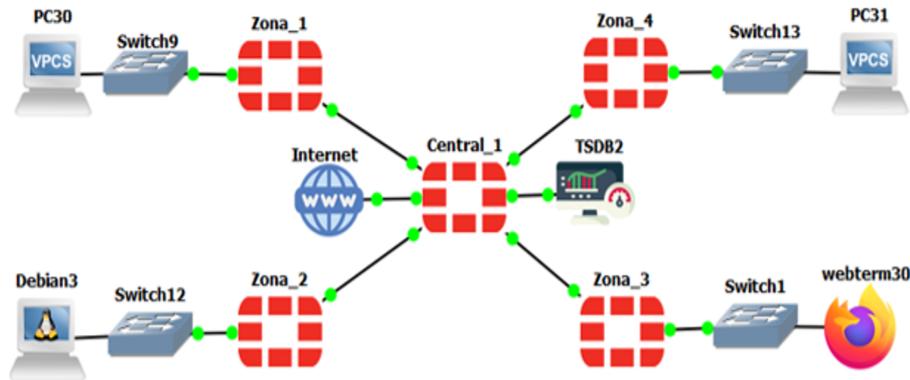


Figura 7: Topología SDWAN estrella

Name	Source	Destination	Schedule	Service	Action	NAT	Security Profiles	Log	Bytes
Zona1(port4)→SD_Fortinet 1									
Salida_Zona1	all	all	always	ALL	✓ ACCEPT	Enabled	no-inspection	UTM	81.62 kB
Zona2(port5)→SD_Fortinet 1									
Salida_Zona2	all	all	always	ALL	✓ ACCEPT	Enabled	no-inspection	UTM	362.43 kB
Zona3(port6)→SD_Fortinet 1									
Salida_Zona3	all	all	always	ALL	✓ ACCEPT	Enabled	no-inspection	UTM	55.67 kB
Zona4(port7)→SD_Fortinet 1									
Salida_zona4	all	all	always	ALL	✓ ACCEPT	Enabled	no-inspection	UTM	0 B

Figura 8: Configuración de firewall policy en SD-WAN estrella

Topología malla red SDWAN: Esta topología consta de 5 enrutadores, 4 switch, 3vpc, 1 navegador web, una máquina Linux; la diferencia con la topología anterior, son las conexiones por cada router. La Figura 9 muestra el diagrama de la topología, mientras que la Figura 10, las configuraciones realizadas.

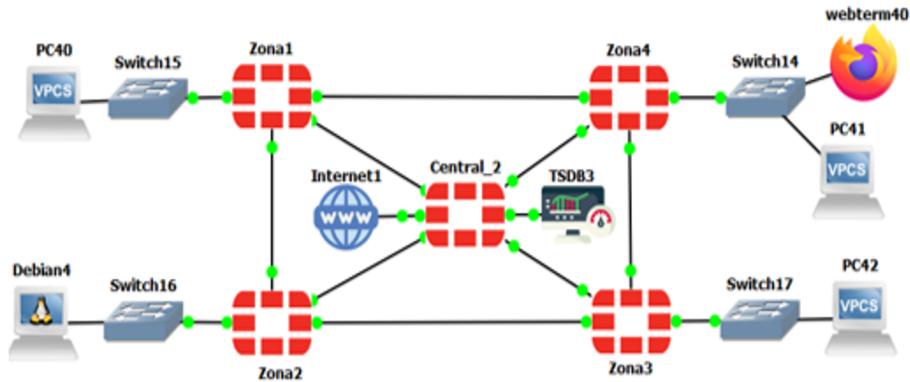


Figura 9: Topología SDWAN malla

	Name	Details	Interface
IP Range/Subnet 7			
FABRIC_DEVICE	0.0.0.0/0		
FIREWALL_AUTH_PORTAL_ADDRESS	0.0.0.0/0		
LAN	194.168.20.0/24	LAN (port9)	
RedundancyZona1	181.168.10.0/24	RedundancyZona1	
SSLVPN_TUNNEL_ADDR1	10.212.134.200 - 10.212.134.210	SSL-VPN tunnel interface (ssl.root)	
all	0.0.0.0/0		
none	0.0.0.0/32		
Interface Subnet 2			
RedundancyZona1 address	181.168.10.0/24	RedundancyZona1	
RedundancyZona3 address	181.168.20.0/24	RedundancyZona3	

Figura 10: Configuración de address en la zona 2 para la red SD-WAN malla

Escenario mixto con SDWAN Y MPLS: Se planteó el diseño de una topología mixta que combine SDWAN Y MPLS, teniendo un enfoque híbrido, donde el equipo “ESW” es un switch Ethernet de capa 3 para interconectar segmentos de red, además el enlace entre LER_3 y CENTRAL_3 es activo, tal como se muestra en la Figura 11.

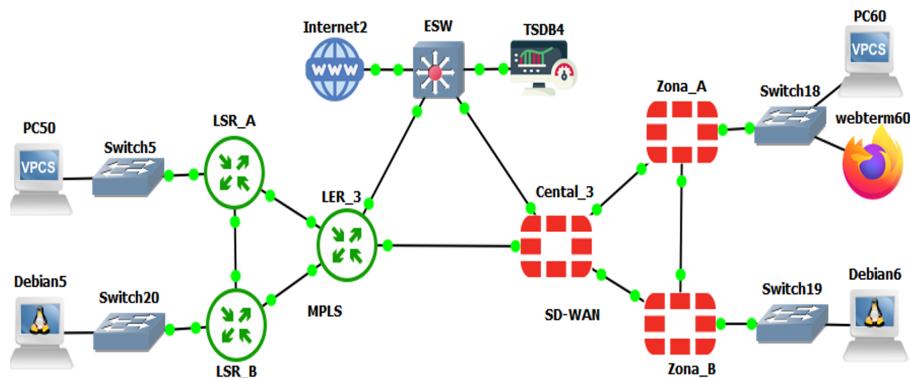


Figura 11: Topología MPLS y SDWAN

3. Resultados

Para el análisis se realizaron diferentes pruebas de tráfico con Python y la librería ping3, midiendo la latencia, jitter, paquetes perdidos, BER, consumo de ancho de banda, tasa de bits, paquetes enviados, tiempo de procesamiento, de

donde se obtuvieron los siguientes resultados tabulados desde la base de datos influx-db en conjunto con Grafana para las diferentes topologías [15], como se muestra en la Figura 12 y la Tabla 6.

```

user@debian:~$ sudo python3 datos.py
[sudo] password for user:
          Pings: 2200
          Topologia: SDWAN ESTRELLA
Latencia Total: 318321.0835
Latencia Max: 922.9598
Latencia Min: 0.0
Latencia Media: 147.3709
Desviacion Standar: 52.9403
Jitter: 43330.2469

          Pings: 2200
          Topologia: MPLS ESTRELLA
Latencia Total: 324482.4485
Latencia Max: 1355.4537
Latencia Min: 120.81
Latencia Media: 150.445
Desviacion Standar: 68.0327
Jitter: 44667.531

          Pings: 2200
          Topologia: MIXTA
Latencia Total: 329075.9296
Latencia Max: 356.3612
Latencia Min: 117.2881
Latencia Media: 152.0003
Desviacion Standar: 18.912
Jitter: 30092.6645

          Paquetes Enviados: 2200
          Paquetes Recibidos: 2160
          Paquetes Perdidos: 40
          Paquetes (Lost/Sent: 0.0182
          Tiempo(s): 400.151
          Bandwidth enviado (Mbps): 0.0447
          Bandwidth recibido (Mbps): 0.0435
          Tasa bits enviado (Mbps): 0.0559
          Tasa bits recibido (Mbps): 0.0553
          BER: 0.0022

          Paquetes Enviados: 2200
          Paquetes Recibidos: 2156
          Paquetes Perdidos: 44
          Paquetes (Lost/Sent: 0.02
          Tiempo(s): 413.824
          Bandwidth enviado (Mbps): 0.0441
          Bandwidth recibido (Mbps): 0.0433
          Tasa bits enviado (Mbps): 0.0455
          Tasa bits recibido (Mbps): 0.0449
          BER: 0.0024

          Paquetes Enviados: 2200
          Paquetes Recibidos: 2188
          Paquetes Perdidos: 12
          Paquetes (Lost/Sent: 0.0055
          Tiempo(s): 354.915
          Bandwidth enviado (Mbps): 0.0504
          Bandwidth recibido (Mbps): 0.0502
          Tasa bits enviado (Mbps): 0.0534
          Tasa bits recibido (Mbps): 0.052
          BER: 0.0007

          Sistema Operativo: Linux 4.19.0-6-az
          IP Local: 198.168.20.10
          IP Destino: 8.8.8.8
          Sistema Operativo: Linux 4.19.0-6-az
          IP Local: 198.168.20.254
          IP Destino: 8.8.8.8
          Sistema Operativo: Linux 4.19.0-6-az
          IP Local: 195.168.20.254
          IP Destino: 8.8.8.8

```

Figura 12: Envío de paquetes con python

Tabla 6: Resultados de la simulación

Network	Latencia (ms)	Desv. Stand.	Jitter (ms)	Paquete Perdido	Tiempo (s)	Bandwidth (Mbps)	Tasa Bits (Mbps)	Bit Error Rate (BER)
Mixta	329076	18.9	30093	12	355	0.101	0.0654	0.0007
SDWAN Malla	334238	57.7	48146	15	366	0.0975	0.103	0.0008
SDWAN Estrella	318321	52.9	43330	40	400	0.0882	0.111	0.0022
MPLS Malla	324983	34.8	46215	45	416	0.0851	0.0885	0.0025
MPLS Estrella	324402	60	44668	44	414	0.0874	0.0904	0.0024

3.1. Latencia y desviación estándar

De los resultados obtenidos, se tiene que la latencia promedio es similar entre todas las topologías, sin presentar mayores variaciones, siendo la topología con menor latencia SDWAN-MALLA con un valor de 147.37ms; en términos de estabilidad, la red mixta presentó una desviación estándar de 18.91, lo que significa que los valores de latencia en general fueron muy similares, como muestra la Figura 13.

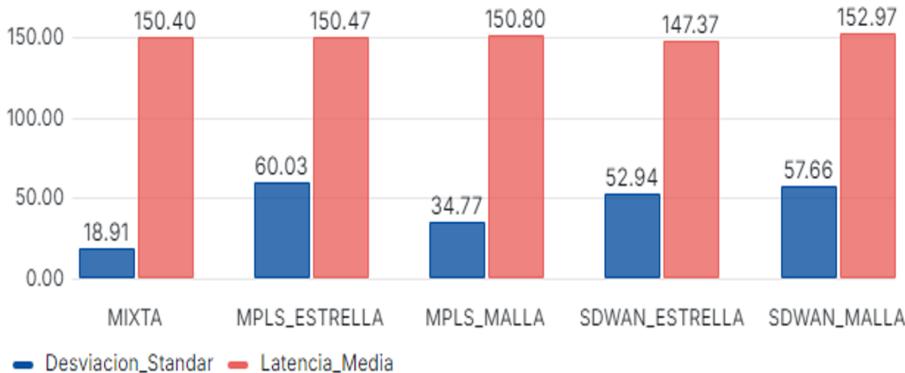


Figura 13: Latencia y desviación estándar

3.2. Ancho de banda y tasa de bits

En cuanto al ancho de banda total consumido y la tasa de bits, los resultados muestran que la topología estrella de la red SD-WAN es la más eficiente, registrando un mayor consumo de ancho de banda (0.111 Mbps) y una tasa de transferencia de 0.888 Mbps, como muestra la Figura 14.

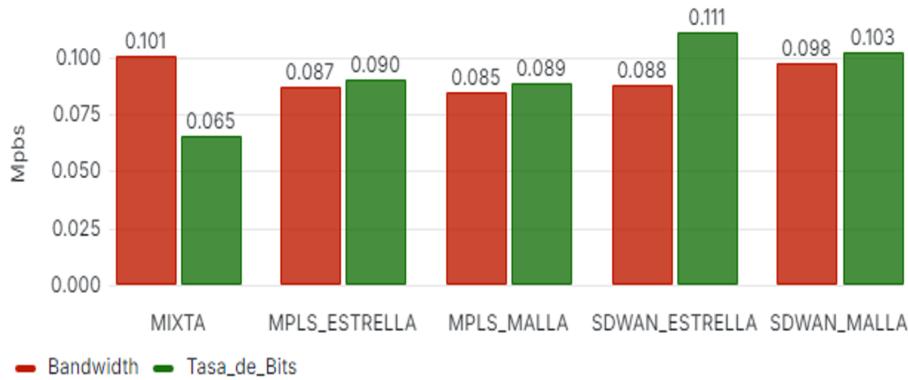


Figura 14: Gráfica de ancho de banda y tasa de bits

3.3. Jitter

El análisis del jitter presentado en la figura, muestra que la topología SD-WAN MALLA, tiene el mayor valor con un 22.66 %, indicando que existe una alta fluctuación en los tiempos de respuesta, sugiriendo que esta topología es susceptible a variaciones y la menos estable, como muestra la Figura 15.

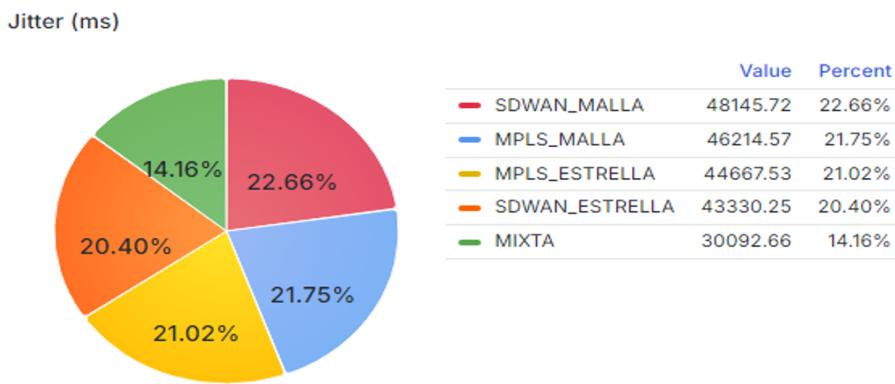


Figura 15: Gráfica de jitter

3.4. Paquetes enviados

En cuanto a la cantidad de paquetes enviados y perdidos, la topología MPLS MALLA presenta la mayor cantidad de paquetes perdidos con 44 paquetes de 2200, por el contrario, la topología mixta, presenta apenas 12 paquetes perdidos, siendo la mejor en este análisis como muestra la Figura 16.

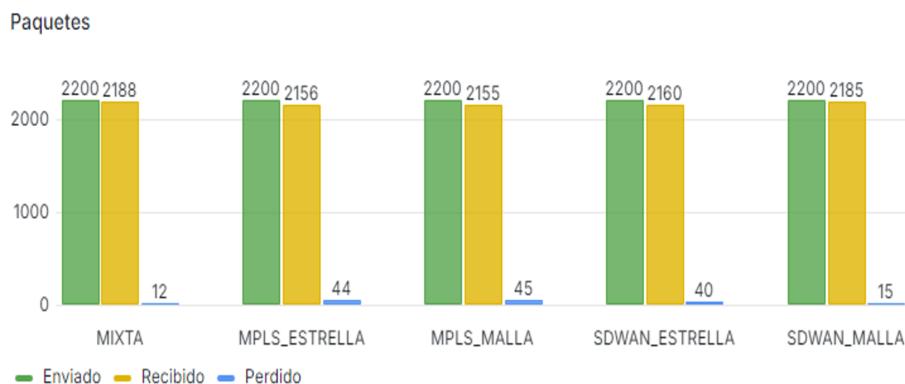


Figura 16: Gráfica de paquetes perdidos

3.5. Tiempo de procesamiento

En el envío de paquetes, la red mixta presenta mayor rapidez al procesar la información con tan solo 355 segundos, mientras que MPLS MALLA, fue la que más demoró con 416 segundos, siendo más eficientes las topologías SD-WAN, como se evidencia en la Figura 17.

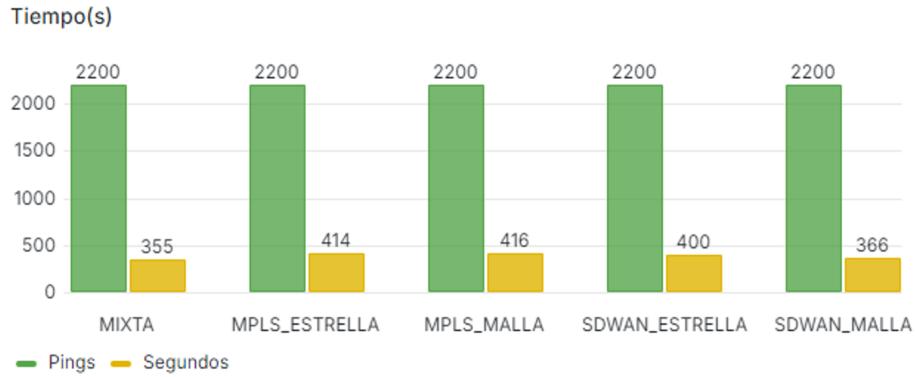


Figura 17: Grafica tiempo de procesamiento

3.6. Bit error rate

En este caso la topología MPLS presenta un valor elevado de 29 % y 27 % en MPLS malla y estrella respectivamente, en comparación con SD-WAN que solo fue de 25 % y 9 %, entendiendo que MPLS es más propenso a errores vs SD-WAN, como se ve en la Figura 18.

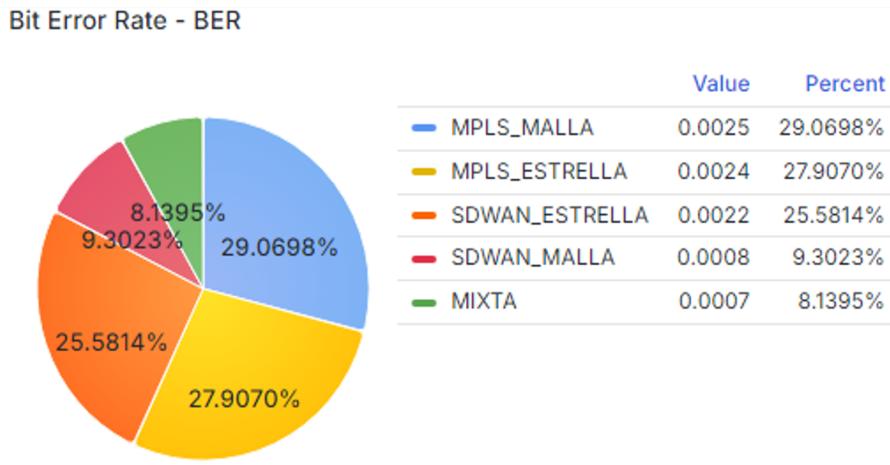


Figura 18: Gráfica BER

4. Conclusiones

Para las pruebas realizadas se utilizó la herramienta Ping, que con el uso de Python permite realizar la automatización y presentación de algunas métricas sin volver caótico este proceso, en este caso las métricas resultantes de esto son latencia total, máxima y media, desviación estándar, jitter, número de paquetes enviados, recibidos y perdidos, tiempo, tasa de bits y ancho de banda consumido.

De las pruebas realizadas en los diferentes escenarios como son la topología MPLS tipo estrella y malla, y la topología SD-WAN tipo estrella y malla, así como la topología mixta, podemos iniciar analizando el parámetro de paquetes perdidos, donde las topologías tipo SD-WAN muestran sin duda una evidente superioridad ante las topologías tipo MPLS, por ejemplo, la topología SD-WAN tipo malla presenta una pérdida de apenas 15 paquetes extraviados

vs ante la topología MPLS tipo malla que tiene un valor de 45 paquetes perdidos, es decir, casi el triple se encontró en las pruebas realizadas, teniendo así las topologías SD-WAN con un mejor rendimiento [16].

Otro parámetro a analizar sus resultados es el tiempo de procesamiento, el cual mide el tiempo en procesar todos los paquetes enviados en estas pruebas, para todas las topologías se enviaron flujos de datos de 2200 pings, donde las topologías tipo SD-WAN mostraron una mejor respuesta ante las topologías MPLS, por ejemplo, la topología SD-WAN tipo malla, presenta un tiempo de procesamiento de 366ms en esta prueba; por otro lado, la topología MPLS tipo malla presenta un tiempo de procesamiento de 416ms, consolidando a las redes con SD-WAN las topologías con mejor rendimiento al momento de procesar paquetes.

El BER que hace referencia a la tasa de error de bit es otro parámetro evaluado en las pruebas de las diferentes topologías. De las pruebas realizadas, las topologías tipo MPLS presentaron un BER alto en comparación con las topologías tipo SD-WAN, un ejemplo de ello es que la topología tipo malla MPLS presenta un BER de 0.0025, frente a la topología SD-WAN tipo malla, que solo tiene un 0.0008, esto nos muestra que hay una probabilidad del 0.25 % de que un bit llegue con error en una red, siendo un valor relativamente alto en telecomunicaciones.

En el resto de parámetros medidos en las diferentes topologías, en general las de tipo SD-WAN presentan ligeras mejoras en conjunto con la topología tipo mixta, frente a las de tipo MPLS, por ejemplo, en la latencia SDWAN tipo estrella presenta una ligera mejora, en el jitter la topología tipo SDWAN estrella presentó un menor jitter en comparación con MPLS, en la pérdida de paquetes la red SDWAN tipo malla, presentó una pérdida de la tercera parte en comparación con MPLS, demostrando una mayor robustez en SDWAN. Si se evalúa a nivel general los resultados obtenidos de las diferentes pruebas en los diferentes escenarios, se tiene que en general, SDWAN presenta mejores resultados ante MPLS.

Referencias

- [1] S. Meneses, Network Design Defined by Software on a Hyper-converged Infrastructure. Case Study: Northern Technical University FICA Data Center, 2020.
- [2] Ema-Maria Galeş, V Croitoru, "Traffic Engineering and QoS in a Proposed MPLS-VPN", IEEE, 2021.
- [3] Basu, K.: Performance comparison of a SDN network between cloud-based and locally hosted SDN controllers. IEEE (2018).
- [4] I. Ramadža, J. Ožegović, and V. Pekić, "Network performance monitoring within MPLS traffic engineering enabled networks," in 2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2015, pp. 315-319.
- [5] S. Mansfield, E. Gray, and K. Lam, "Network Management Framework for MPLS-based Transport Networks," Request for Comments: 5950, Internet Engineering Task Force, Sep. 2010.
- [6] J. Wang, M. Bewong and L. Zheng, "SD-WAN: Hybrid Edge Cloud Network between Multi-site SDDC," Computer Networks, vol. 250, p. 110509, 2024.
- [7] A. H. Abdi et al., "Security Control and Data Planes of SDN: A Comprehensive Review of Traditional, AI, and MTD Approaches to Security Solutions," in IEEE Access, vol. 12, pp. 69941-69980, 2024.
- [8] Z. A. Bhuiyan, S. Islam, M. M. Islam, A. B. M. A. Ullah, F. Naz and M. S. Rahman, "On the (in)Security of the Control Plane of SDN Architecture: A Survey," IEEE Access, vol. 11, pp. 91550-91582, 2023, doi: 10.1109/ACCESS.2023.3307467.
- [9] Blenk, A.: Pairing SDN with network virtualization: the network hypervisor placement problem. IEEE (2015).
- [10] Arpita Saxena: Improving Load Balancing through SD-WAN: Key Aspects to Not-to-beMissed, <https://searchnetworking.techtarget.com/definition/WAN-vsload-balancing-How-are-they-different>, 22 Sep 2022.
- [11] C. Fu, B. Wang, H. Liu and W. Wang, "Software-defined virtual private network for SD-WAN," Electronics, vol. 13, no. 13, p. 2674, 2024, doi: 10.3390/electronics13132674.
- [12] S. Oladosu, C. C. Ike, P. Adepoju, A. Afolabi, A. Ige and O. Amoo, "The future of SD-WAN: A conceptual evolution from traditional WAN to autonomous, self-healing network systems," Magna Scientia Advanced Research and Reviews, vol. 3, no. 2, pp. 95-107, 2021, doi: 10.30574/msarr.2021.3.2.0086.
- [13] Vora, J.: Performance evaluation of SDN based virtualization for data center networks. IEEE (2018).
- [14] Kshira Sagar Sahoo et. Al: A Comprehensive Tutorial on Software Defined Network: The Driving Force for the Future Internet Technology, ACM, ISBN 978-1-4503-4213-1/16/08, DOI: <http://dx.doi.org/10.1145/2979779.2983928>, 2016.

- [15] A. H. Alhilali and A. Montazerolghaem, "Artificial intelligence based load balancing in SDN: A comprehensive survey," *Internet of Things*, vol. 22, p. 100814, 2023.
- [16] Ventre, P.L.: Deploying SDN in G EANT production network. IEEE (2017).