



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA EN REDES Y
COMUNICACIÓN DE DATOS**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN ELECTRÓNICA EN REDES Y COMUNICACIÓN DE
DATOS**

**TEMA: EVALUACIÓN DE RENDIMIENTO ENTRE LAS TECNOLOGÍAS
DE EVPN Y VPLS SOBRE UNA RED MPLS EN UN AMBIENTE JUNIPER
SIMULADO EN GNS3**

AUTOR: CALAHORRANO VEGA, CÉSAR AUGUSTO

DIRECTOR: ING. ROMERO GALLARDO, CARLOS GABRIEL

SANGOLQUÍ

2019



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA EN
REDES Y COMUNICACIÓN DE DATOS

CERTIFICACIÓN

Certificó que el trabajo de titulación, “EVALUACIÓN DE RENDIMIENTO ENTRE LAS TECNOLOGÍAS DE EVPN Y VPLS SOBRE UNA RED MPLS EN UN AMBIENTE JUNIPER SIMULADO EN GNS3”, realizado por el Sr. CÉSAR AUGUSTO CALAHORRANO VEGA, ha sido revisado en su totalidad y analizado en el Software anti-plagio, el mismo cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas - ESPE, por lo tanto me permito acreditarlo y autorizar al Sr. CÉSAR AUGUSTO CALAHORRANO VEGA para que lo sustente públicamente.

Sangolquí, enero del 2019

Ing. Carlos Gabriel Romero Gallardo

C.C. 1712198066



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA EN
REDES Y COMUNICACIÓN DE DATOS

AUTORÍA DE RESPONSABILIDAD

Yo, CÉSAR AUGUSTO CALAHORRANO VEGA, con cédula de identidad # 1717556714 declaro que este trabajo de titulación “EVALUACIÓN DE RENDIMIENTO ENTRE LAS TECNOLOGÍAS DE EVPN Y VPLS SOBRE UNA RED MPLS EN UN AMBIENTE JUNIPER SIMULADO EN GNS3”, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado derechos intelectuales de terceros, considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, enero del 2019

César Augusto Calahorrano Vega
C.C. 1717556714



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA EN
REDES Y COMUNICACIÓN DE DATOS

AUTORIZACIÓN

Yo, CÉSAR AUGUSTO CALAHORRANO VEGA, autorizo a la Universidad de las Fuerzas Armadas - ESPE publicar en la biblioteca virtual de la Institución el presente trabajo de titulación “EVALUACIÓN DE RENDIMIENTO ENTRE LAS TECNOLOGÍAS DE EVPN Y VPLS SOBRE UNA RED MPLS EN UN AMBIENTE JUNIPER SIMULADO EN GNS3”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, enero del 2018

César Augusto Calahorrano Vega
C.C. 1717556714

DEDICATORIA

Dedico este trabajo con mucho amor y gratitud a mi Padre Oswaldo Calahorrano Conrado quien con su ejemplo de honestidad y trabajo duro ha marcado mi vida para conseguir grandes logros, a mi Madre Marlene Vega Jaramillo por enseñarme los principios más importantes que debe existir en una persona, la humildad y el respeto.

A mis hermanos Carlos, Paola y Oswaldo, por compartir conmigo cada momento brindándome su amor incondicional, apoyo y confianza.

CÉSAR CALAHORRANO VEGA

AGRADECIMIENTO

Agradezco a Dios por la bendición de tener a mis Padres con vida y permitirme disfrutar todos mis logros con ellos.

A los docentes que compartieron sus conocimientos para mi formación académica y que han sido fundamentales para llevar a cabo la culminación de mi carrera exitosamente.

CÉSAR CALAHORRANO VEGA

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS	xii
RESUMEN	xvii
ABSTRACT	xviii
CAPÍTULO 1	1
INTRODUCCIÓN	1
1.1. ANTECEDENTES	1
1.2. JUSTIFICACIÓN E IMPORTANCIA	2
1.3. ALCANCE	3
1.4. OBJETIVOS	4
1.4.1. General.	4
1.4.2. Específicos.	4
CAPÍTULO 2	5

ESTADO DEL ARTE	5
1.5. MPLS Multiprotocolo Label Switching	5
2.1.1. Características básicas y funcionamiento	5
2.1.2. Arquitectura	7
2.1.2.1. Características y Funcionamiento	7
2.1.2.2. Elementos	7
2.1.2.3. Formato de trama	8
2.2. VPN	9
2.2.1. ¿Cómo funciona una VPN?	10
2.2.2. Usos de una VPN	12
2.2.3. VPN de capa 2	12
2.2.3.1. VPN basadas en BGP	13
2.3. VPLS Virtual Private LAN Service	15
2.3.1. Terminología	16
2.3.2. Interacción	17
2.3.3. Plano de Control	17
2.3.4. Auto-descubrimiento	18
2.3.4.1. Funciones	18
2.3.4.2. Especificación del Protocolo	19
2.3.4.3. Señalización	19
2.3.4.4. Conceptos	20

2.3.4.5.	Capacidades de PE de Señalización	21
2.3.5.	Plano de Datos	22
2.3.5.1.	Encapsulación	22
2.3.5.2.	Reenvío	23
2.3.5.3.	Aprendizaje de direcciones MAC	23
2.3.5.4.	Tiempo de vida.....	24
2.3.5.5.	Inundación	24
2.3.5.6.	Transmisión y multidifusión.....	25
2.3.5.7.	Desvío “Split Horizon”	25
2.3.5.8.	Aprendizaje calificado y no calificado	26
2.3.5.9.	Opciones de implementación	26
2.4.	EVPN Ethernet Virtual Private Network	27
2.4.1.	Terminología.....	27
2.4.2.	EVPN basado en BGP MPLS.....	29
2.4.3.	Segmento Ethernet ES	30
2.4.4.	ID de etiqueta de Ethernet (Ethernet Tag ID).....	32
2.4.4.1.	Interfaz de servicio basada en VLAN.....	33
2.4.4.2.	Interfaz de servicio de paquete VLAN	33
2.4.4.3.	Rutas BGP EVPN.....	34
2.4.4.4.	Ruta de descubrimiento automático	35
2.4.4.5.	Ruta de anuncio MAC/IP	35

2.4.4.6.	Ruta de etiqueta de Ethernet de multidifusión inclusiva	36
2.4.4.7.	Ruta del segmento de Ethernet	36
2.4.5.	Funciones de Multihoming	37
2.4.5.1.	Autodescubrimiento del segmento MultiHomed Ethernet	37
2.4.5.2.	Convergencia rápida	37
2.4.5.3.	Construcción de AD de Ethernet por ruta de segmento de Ethernet	38
2.4.5.3.1.	Destinos de ruta AD de Ethernet	39
2.4.6.	Determinación de la accesibilidad a direcciones MAC de unidifusión	39
2.4.6.1.	Aprendizaje local	39
2.4.6.2.	Aprendizaje remoto	40
2.4.7.	Movilidad MAC	40
2.5.	Fundamentación	42
2.5.1.	Aspectos Generales	42
2.5.1.1.	Hardware	43
2.5.1.2.	Software	45
2.5.1.2.1.	JUNOS (Juniper Network Operating System)	45
2.5.1.2.2.	IOS (Cisco IOS)	47
2.5.1.2.3.	GNS3	47
2.5.1.2.4.	Motores de Simulación	49
2.5.1.2.4.1.	QEMU	49
2.5.1.2.4.2.	VMware player	50

2.5.1.2.5. Herramientas de Análisis.....	51
2.5.1.2.5.1. Ostinato	51
2.5.1.2.5.2. Wireshark.....	51
2.5.1.2.5.3. Tiny Core Linux	53
CAPÍTULO 3.....	54
PROTOTIPO DE IMPLEMENTACIÓN DE VPLS Y EVPN.....	54
3.1. Especificación de Requerimientos.....	54
3.2. Especificación de escenarios	56
3.3. Preparación del entorno de simulación	57
3.3.1. GNS3.....	57
3.3.1.1. Instalación	57
3.3.1.2. Configuración Básica	61
3.3.1.2.1. Creación de interfaz loopback.....	61
3.3.1.2.2. Configuración de servidores.....	62
3.3.2. Motores de virtualización	63
3.3.2.1. VMware.....	63
3.3.2.2. QEMU.....	67
3.3.3. Herramientas de análisis.....	70
CAPÍTULO 4.....	72
IMPLEMENTACIÓN Y ANÁLISIS DE RENDIMIENTO	72
4.1. Topología de Red	72

4.1.1.	Direccionamiento Físico	73
4.1.2.	Direccionamiento Lógico	75
4.1.3.	Direccionamiento lógico VPLS	75
4.1.3.1.	Direccionamiento lógico EVPN	76
4.2.	Lineamientos generales	77
4.3.	Información de alcanzabilidad	79
4.4.	Información L2VPN	80
4.4.1.	VPLS	80
4.4.2.	EVPN	81
4.5.	Análisis de resultados	83
4.5.1.	Convergencia de la red total	83
4.5.2.	Movilidad MAC	92
4.5.2.1.	EVPN	93
4.5.2.2.	VPLS	100
4.5.3.	Supresión de Tráfico BUM	102
4.5.3.1.	EVPN	104
4.5.3.2.	VPLS	107
CAPÍTULO 5		109
CONCLUSIONES Y RECOMENDACIONES		109
5.1	Conclusiones	109
5.2	Recomendaciones	111

BIBLIOGRAFÍA	112
---------------------------	-----

ÍNDICE DE TABLAS

Tabla 1. <i>Papel de Routers.</i>	73
Tabla 2. <i>Asignación de interfaces.</i>	74
Tabla 3. <i>Interfaces loopback.</i>	78
Tabla 4. <i>Lineamientos generales</i>	78
Tabla 5. <i>Configuración de red de los Clientes Finales.</i>	83
Tabla 6. <i>Convergencia EVPN.</i>	89
Tabla 7. <i>Convergencia VPLS.</i>	91
Tabla 8. <i>Convergencia VPLS vs EVPN.</i>	91

ÍNDICE DE FIGURAS

Figura 1. <i>Esquema de una Red MPLS.</i>	7
Figura 2. <i>Formato de trama MPLS.</i>	8
Figura 3. <i>Esquema de una VPN.</i>	11
Figura 4. <i>Tipos de redes privadas virtuales.</i>	13
Figura 5. <i>Tipos de redes privadas virtuales.</i>	15
Figura 6. <i>Ejemplo de una VPLS.</i>	16
Figura 7. <i>BGP NLRI para información de VPLS.</i>	21
Figura 8. <i>Comunidad ampliada de capa 2 INFO.</i>	21

Figura 9. Comunidad ampliada de capa 2 INFO.....	22
Figura 10. Trama ESI.....	31
Figura 11. Trama NLRI.....	34
Figura 12. Ruta de auto-descubrimiento.	35
Figura 13. Ruta de anuncio.	35
Figura 14. Ruta de etiqueta de Ethernet de multidifusión inclusiva.	36
Figura 15. Ruta del segmento Ethernet.	36
Figura 16. Router MX 80.	43
Figura 17. Cisco C2691.....	44
Figura 18. Ordenador.	44
Figura 19. Logo JunOS.	45
Figura 20. Logo Cisco IOS.	47
Figura 21. Logo GNS3.....	48
Figura 22. Logo QEMU.	49
Figura 23. Logo VMware Player.	50
Figura 24. Logo Ostinato.	51
Figura 25. Logo Wireshark.	52
Figura 26. Logo Tiny Core Linux.....	53
Figura 27. Arquitectura de simulación.....	56
Figura 28. Topología EVPN [23].....	56
Figura 29. Instalación GNS3 [Paso 1].....	58
Figura 30. Instalación GNS3 [Paso 2].....	58
Figura 31. Instalación GNS3 [Paso 3].....	59

Figura 32. Instalación GNS3 [Paso 4].....	60
Figura 33. Instalación GNS3 [Paso 5].....	60
Figura 34. Interfaz Loopback [Paso 1].....	61
Figura 35. Interfaz Loopback [Pasos 2-4].....	62
Figura 36. Preferencias de Servidores GNS3.....	63
Figura 37. Link de descarga GNS3 VM.....	64
Figura 38. Importación de GNS3 VM en VMware.....	64
Figura 39. Configuración de red del adaptador VMnet1.	65
Figura 40. Pasos para la configuración de red de GNS3 VM [Pasos 1-4].	66
Figura 41. Configuración de red de la arquitectura de simulación.	67
Figura 42. GNS3 VM en línea.	68
Figura 43. QEMU VM template.	69
Figura 44. Creación del nuevo template Qemu VM [Pasos 2-4].	69
Figura 45. Appliance JunOS vMX 14.1R1.10.	70
Figura 46. Appliance Ostinato 0.9.	71
Figura 47. Topología de Red.....	72
Figura 48. Direccionamiento Físico.....	74
Figura 49. Direccionamiento Lógico VPLS.....	75
Figura 50. Direccionamiento Lógico EVPN.....	76
Figura 51. Tabla inet.0 PE1-R1 (VPLS vs EVPN).....	80
Figura 52. Tabla vpls.l2vpn.0 PE1-R1.....	81
Figura 53. Tabla vpls.l2vpn.0 PE2-R2.....	81
Figura 54. Tabla vpls.l2vpn.0 PE3-R3.....	81

Figura 55. Tabla vrf.inet.0 PE1-R1	82
Figura 56. Tabla vrf.inet.0 PE2-R2.....	82
Figura 57. Tabla vrf.inet.0 PE3-R3.....	82
Figura 58. Conectividad Ping PC-1 a PC-2 y PC-3.....	84
Figura 59. Conectividad Ping PC-2 a PC-1 y PC-3.....	84
Figura 60. Conectividad Ping PC-3 a PC-1 y PC-2.....	85
Figura 61. Trazado PC-1 a PC-2 y PC-3.....	86
Figura 62. Trazado PC-2 a PC-1 y PC-3.....	86
Figura 63. Trazado PC-3 a PC-1 y PC-2.....	86
Figura 64. Escenario de pruebas -Convergencia-.....	87
Figura 65. Captura de paquetes EVPN.....	88
Figura 66. Captura de paquetes VPLS.....	90
Figura 67. Escenario de pruebas Movilidad MAC.....	92
Figura 68. Movilidad MAC.....	93
Figura 69. Tabla-MAC EVPN.....	94
Figura 70. Captura de paquetes enlace PE1-P1.....	95
Figura 71. Mensaje Update 162.....	96
Figura 72. Mensaje Update 251.....	97
Figura 73. Mensaje Update 177.....	98
Figura 74. Mensaje Update 177.....	99
Figura 75. Mensaje Update 177.....	99
Figura 76. Tabla-MAC VPLS.....	101
Figura 77. Captura de paquetes enlace PE1-P1.....	101

Figura 78. Tabla-MAC VPLS.....	102
Figura 79. Escenario de pruebas tráfico BUM.....	103
Figura 80. Configuración tipo de Protocolo.....	104
Figura 81. Configuración direcciones MAC e IP.....	104
Figura 82. Inundación ARP -antes-.....	105
Figura 83. Inundación ARP-después-.....	105
Figura 84. Tabla EVPN MAC.....	106
Figura 85. Inundación ARP -antes-.....	107
Figura 86. Inundación ARP-después-.....	107
Figura 87. Tabla VPLS MAC.....	108

RESUMEN

En el presente proyecto se estudia la comparativa entre dos tecnologías de Red Virtual Privada, VPLS y EVPN, las mismas que están implementadas sobre una red IP/MPLS, con el fin de evaluar el rendimiento tanto cualitativo como a nivel de entramado. Lo cual permita a futuras implementaciones basadas en VPN escoger la opción más adecuada para permitir el despacho de información entre varios puntos ubicados distantes geográficamente. La comparación se realiza en un ambiente simulado sobre la plataforma del proveedor Juniper (JunOS), con la ayuda del software de simulación GNS3, donde se implementa un escenario de pruebas y se verifica por medio de herramientas de análisis el rendimiento y características que prestan cada una de las tecnologías de VPN.

PALABRAS CLAVE:

- **JUNOS SISTEMA OPERATIVO**
- **GNS3 SIMULADOR**
- **VPN RED VIRTUAL**
- **VPLS SERVICIO**
- **EVPN SERVICIO**

ABSTRACT

In this project, the comparison between two Private Virtual Network technologies VPLS and EVPN is studied. These technologies are implemented on IP/MPLS network in order to evaluate both qualitative and framing performance which will allow future implementations (based on VPN) to choose the best option and at the same time to permit the dispatch of information among several points placed geographically distant. The comparison is made in a simulated environment (on the platform of the Juniper provider 'JunOS') with the assistance of the GNS3 simulation software, where a test scenario is implemented and the performance and features that each of the VPN technologies provide are verified with tools of analysis.

KEY WORDS:

- **JUNOS OPERATING SYSTEM**
- **GNS3 SIMULATOR**
- **VPN VIRTUAL NETWORK**
- **VPLS SERVICE**
- **EVPN SERVICE**

CAPÍTULO 1

INTRODUCCIÓN

1.1. ANTECEDENTES

La tecnología VPN creada originalmente para las grandes empresas, nunca fue elaborada para propósitos que hoy en día son tan cotidianos para medianas y pequeñas organizaciones. La necesidad en ese momento era indispensable: compañías, universidades, gobiernos y muchos otros con información altamente importante estaban en riesgo de ser alterada, hurtada u otras pérdidas de datos en el ambiente de internet. Se necesitaban tener conexiones que fuesen más seguras para los usuarios remotos, las oficinas sucursales y operadores de campo puedan acceder y utilizar archivos de la empresa sin permitir que sus secretos se escapen. La solución a la que arribaron fue la VPN. (Townnsley, Valencia, Rubens, & Pall, 1999)

Los proveedores de servicios SP (Service Provider) efectúan conexiones a nivel de usuarios finales por medio de VPN (Redes Virtuales Privadas – Virtual Private Networks) éste tipo de técnicas ha ido evolucionando en los últimos diez años, de la misma forma los ambientes de enrutamiento IP, para esto, los modelos de VPN's punto a punto han sido replicados en múltiples compañías, las cuales utilizan el protocolo MPLS (Multi-protocol Label Swithing), obteniendo una mejor funcionalidad en cuanto a planificación y expansión de aplicaciones extendidas en una red; esto gracias al incremento de usuarios y sus necesidades, entre ellas sistemas de monitoreo, sistemas gobierno, ampliación de segmentos de redes privadas con altos estándares de seguridad. Ante todas éstas exigencias de los usuarios, surge la necesidad de caracterizar las redes

MPLS/VPN/BGP, con mecanismos de mejora del rendimiento en este caso las rutas reflejadas. (Rosen & Rakhter, 2006)

1.2. JUSTIFICACIÓN E IMPORTANCIA

La popularidad de MPLS (Multi-Protocol Label Switching) no es indiferente para las empresas; la capacidad que presentan en integrar voz, vídeo y datos en un entorno común con garantías de calidad de servicio (QoS), se deben sumar las mejoras en disponibilidad y rendimiento que se obtienen con ésta solución, así como el soporte de una amplia y escalable gama de servicios. Su topología de muchos-a-muchos (any-to-any) ofrece a los administradores la flexibilidad para manejar el tráfico sobre la marcha en caso del fallo de enlaces y congestión de red.

Los servicios VPN se tratan de una tecnología extensamente probada y exitosamente funcional, continuamente evolucionando y se ha convertido en una forma simple y económica de brindar a los usuarios de Internet conexiones seguras y privadas.

VPN Ethernet (EVPN) es la solución de próxima generación que proporciona servicios Ethernet multipunto a través de redes MPLS. EVPN es diferente en comparación con las ofertas existentes de Virtual Private LAN Service (VPLS) debido a su uso del control basado en MAC basado en el aprendizaje del núcleo.

Los servicios de VPN Ethernet pueden implementarse sobre cualquier tipo de acceso o circuito terminal, incluso sobre redes no Ethernet basadas en tecnologías convencionales como Frame Relay o ATM, así como a través de conexiones DSL. Con ello, las organizaciones pueden disfrutar de todas las ventajas que aporta una red basada íntegramente en Ethernet, minimizando el costo del cambio. Una característica exclusiva de este servicio es que permite al cliente gestionar

su propio enrutamiento a través de su propia red VPN, lo que le proporciona un extraordinario grado de control y seguridad. (Sajassi, y otros, 2015)

1.3. ALCANCE

El tema “Evaluación de rendimiento entre las tecnologías de EVPN y VPLS sobre una red MPLS en un ambiente JUNIPER simulado en GNS3”, se contemplará de la siguiente manera:

El entorno de pruebas será realizado en el software de simulación gráfico GNS3 v2.1.4 con sus respectivos complementos para el análisis de resultados (Wireshark, Ostinato), además utilizará los motores de virtualización como QEMU y VMWARE para optimizar el ambiente de simulación debido a las características de hardware (PC, Router, Switch).

En el entorno de pruebas se utilizarán imágenes del sistema operativo JUNOS (.img) versión 14.1R2.10 de la serie VMX del proveedor JUNIPER, e imágenes del sistema operativo CISCO IOS (c2691-js-mz.122-15.T13).

Mediante la simulación, se pretende recrear un ambiente con los elementos de red necesarios, la misma que permitirá entre otras cosas, determinar los parámetros de análisis y las pruebas necesarias a realizarse. Así también, se analizarán las características y funcionamiento de los equipos a utilizarse.

Dentro de la etapa de pruebas, se realizará el diseño, implementación de escenarios que permitan obtener resultados con métricas, así como el comportamiento de los protocolos para VPN's.

Para el estudio comparativo se utilizará el analizador de protocolos WIRESHARK, con el que se evidenciará el comportamiento a nivel de tramas de las tecnologías VPN.

Para la simulación en el estudio comparativo se utilizará el software OSTINATO, con el propósito de crear un ambiente aproximado al de una red real de un proveedor de servicios y verificar el rendimiento de las tecnologías.

Posteriormente, los resultados obtenidos serán analizados y validados, generando una comparativa, la cual permita en futuras implementaciones basadas en MPLS, utilizar protocolos de despacho más robustos.

1.4. OBJETIVOS

1.4.1. General.

- Evaluar el desempeño de rendimiento de tecnologías basadas en Red Virtual Privada, a fin de determinar la que preste el mejor desempeño en redes de proveedores.

1.4.2. Específicos.

- Investigar las herramientas-software con las características para la medición del desempeño de la red con cada tecnología de VPN.
- Encontrar el escenario adecuado para evaluar el comportamiento de EVPN y VPLS sobre una red MPLS.
- Simular el escenario idóneo en GNS3 con imágenes de sistemas operativos de dispositivos de redes de datos que soporten las tecnologías de VPN.
- Revisar las configuraciones necesarias para la simulación de EVPN y VPLS en Juniper.
- Analizar los resultados de EVPN y VPLS sobre el escenario simulado

CAPÍTULO 2

ESTADO DEL ARTE

1.5. MPLS Multiprotocolo Label Switching

MPLS se inventó a finales de la década de 1990, en un momento en que el modo de transferencia asincrónica (ATM) era una tecnología WAN generalizada. ATM tenía algunas virtudes: multiservicio, transporte asíncrono, clase de servicio, estado de envío reducido, previsibilidad, etc. Pero tenía al menos tantos defectos: ninguna tolerancia a la pérdida o reordenamiento de los datos, una sobrecarga de reenvío que lo hacía inadecuado para las altas velocidades, ningún multipunto decente, la falta de una integración nativa con IP, etc. MPLS aprendió de la instructiva experiencia ATM, aprovechando sus virtudes mientras solucionaba sus defectos. MPLS es una tecnología de reenvío asíncrono basado en paquetes. En ese sentido, es similar a IP, pero MPLS tiene un plano de reenvío mucho más ligero y reduce en gran medida la cantidad de estado que necesita ser señalado y programado en los dispositivos. (Sánchez & Grzegorz, 2015)

2.1.1. Características básicas y funcionamiento

Cuando un enrutador de Internet recibe un paquete de IP, ese paquete no contiene información más allá de una dirección IP de destino. No hay instrucciones sobre cómo ese paquete debe llegar a su destino o cómo debe tratarse en el camino. Cada enrutador debe tomar una decisión de reenvío independiente para cada paquete basándose únicamente en el encabezado de la capa de red del paquete. Por lo tanto, cada vez que un paquete llega a un enrutador, el enrutador debe

"pensar" dónde enviar el paquete a continuación. El enrutador hace esto al referirse a tablas de enrutamiento complejas. El proceso se repite en cada salto a lo largo de la ruta hasta que el paquete finalmente llegue a su destino. Todos esos saltos y todas esas decisiones de enrutamiento individuales dan como resultado un rendimiento deficiente para aplicaciones sensibles al tiempo como videoconferencia o voz sobre IP (VoIP). Con MPLS, la primera vez que un paquete ingresa a la red, se asigna a una clase de equivalencia de reenvío específica (FEC), que se indica al agregar una secuencia de bit corto (la etiqueta) al paquete. Cada enrutador de la red tiene una tabla que indica cómo manejar los paquetes de un tipo de FEC específico, por lo que una vez que el paquete ha ingresado a la red, los enrutadores no necesitan realizar un análisis de encabezado. En cambio, los enrutadores posteriores usan la etiqueta como un índice en una tabla que les proporciona un nuevo FEC para ese paquete.

Esto da a la red MPLS la capacidad de manejar paquetes con características particulares (tales como provenientes de puertos particulares o que transportan tráfico de tipos de aplicaciones particulares) de manera consistente. Los paquetes que transportan tráfico en tiempo real, como voz o video, se pueden asignar fácilmente a rutas de baja latencia en toda la red, algo que es difícil con el enrutamiento convencional. El punto clave de la arquitectura con todo esto es que las etiquetas proporcionan una forma de adjuntar información adicional a cada paquete, información que va más allá de lo que los enrutadores tenían previamente. (Anónimo, 2014)

2.1.2. Arquitectura

2.1.2.1. Características y Funcionamiento

MPLS hace uso de los protocolos de ruteo IP heredados, mediante ellos, MPLS dispone de un conocimiento preciso del estado de la Red. Son necesarios mecanismos de señalización, su empleo siempre precederá al establecimiento de una comunicación extremo a extremo. LDP (Protocolo de Distribución de Etiquetas) y RSVP (Protocolo de Reserva de Recursos), son los protocolos de señalización elegidos.

Cada conexión transita por un trayecto virtual de extremo a extremo, éste trayecto es pactado y establecido según el estado de la Red y las necesidades de la conexión. El proceso de Forward no actúa sobre el contenido de nivel 3 de cada paquete, se añade una etiqueta a cada paquete, y en función de ésta se realiza el Forward. La interpretación y sustitución de cada etiqueta se circunscribe a un ámbito local, es decir, en cada conmutador MPLS. (Garcia, 2002)

2.1.2.2. Elementos

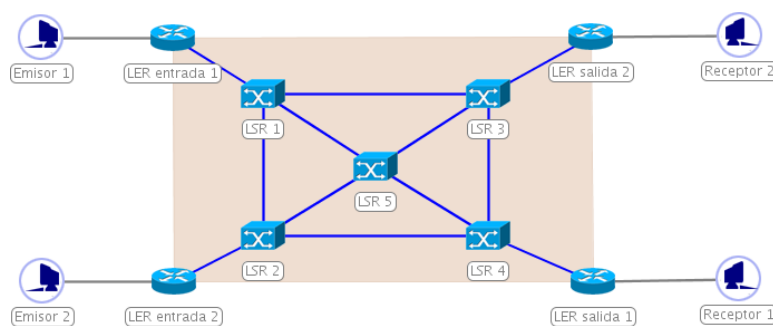


Figura 1. Esquema de una Red MPLS

- LSR (Label Switching Router – Enrutadores conmutadores de etiquetas): Elemento que conmuta etiquetas.

- LSP (Label Switched Path - Caminos conmutados mediante etiquetas): Nombre genérico de un camino MPLS (para cierto tráfico o FEC) es decir del túnel MPLS establecido entre los extremos.
- FEC (Forwarding Equivalence Class - Clase Equivalente de Envío): Nombre que se le da al tráfico que se encamina bajo una etiqueta. Subconjunto de paquetes tratados del mismo modo por el conmutador.

2.1.2.3. Formato de trama

En la figura 2 se representa el esquema de los campos de la cabecera genérica MPLS y su relación con las cabeceras de los otros niveles. Según se muestra en la figura, los 32 bits de la cabecera MPLS se reparten en: 20 bits para la etiqueta MPLS, 3 bits para identificar la clase de servicio en el campo EXP (experimental, anteriormente llamado CoS), 1 bit de stack para poder apilar etiquetas de forma jerárquica (S) y 8 bits para indicar el TTL (time-to-live) que sustenta la funcionalidad estándar TTL de las redes IP. De este modo, las cabeceras MPLS permiten cualquier tecnología o combinación de tecnologías de transporte, con la flexibilidad que esto supone para un proveedor IP a la hora de extender su red. (Tanenbaun, 2011)

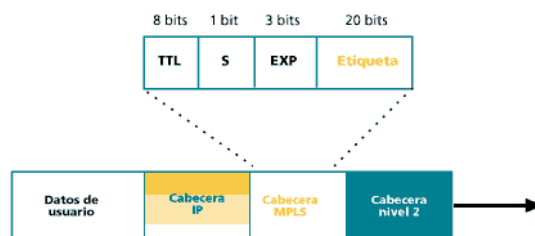


Figura 2. Formato de trama MPLS

2.2. VPN

Las grandes empresas a menudo tienen sitios que se extienden en lugares distantes, estos sitios dispersos deben interconectarse con el mismo nivel de seguridad y privacidad que en una red de área local. Las redes privadas virtuales (VPN) se usan para satisfacer esta necesidad. Una topología VPN común está compuesta por extensiones multiprotocolo para el protocolo Border Gateway (MP-BGP) y Multiprotocol Label Switching (MPLS). Esta tecnología preserva la privacidad entre múltiples VPN a través del aislamiento del tráfico de túneles MPLS, mientras que BGP distribuye información de rutas VPN. A pesar del amplio despliegue de BGP MPLS VPN, las desventajas de esta tecnología VPN se exponen gradualmente con la expansión de la escala VPN. Un análisis simple muestra que la convergencia de rutas lentas y la transferencia lenta de la tabla de rutas es la razón principal. La convergencia de rutas lentas aumenta el tiempo para actualizar las rutas de actualización de intercambio entre los enrutadores, mientras tanto, la transferencia de la tabla de ruta lenta conduce a una menor utilización de la red troncal del proveedor. (Scalability testing of legacy MPLS-based Virtual Private Networks)

Básicamente, una VPN es una red privada que utiliza una red pública (usualmente la Internet) para conectarse a sitios remotos o a varios usuarios. En vez de utilizar líneas especiales, una VPN utiliza conexiones "virtuales" que son dirigidas a través de la Internet desde la red privada al sitio remoto de los usuarios en el lugar que se encuentren.

Existen dos tipos de VPN: de acceso remoto, llamada también red privada virtual de llamada (VPDN), que es una conexión de área local utilizada por una Matriz que requiere que sus usuarios se conecten a una red privada encriptada desde varios puntos remotos y la red de acceso

de sitio a sitio en la cual una Matriz se puede conectar a múltiples sitios fijos de una red pública como la Internet.

Las VPN permiten que cada miembro remoto de una red se comunique mediante un sistema seguro y confiable, utilizando la Internet como el medio de conexión a una red privada virtual. Una VPN puede crecer lo suficiente como para acomodar muchos usuarios en distintas ubicaciones, siendo el costo mínimo comparado con el sistema de líneas dedicadas. (Andrew, 2007)

2.2.1. ¿Cómo funciona una VPN?

Una VPN usando una analogía puede ser representada como una autopista con un inicio y un final, y con diferentes puntos de control por donde viajan los paquetes de información. Ahora bien, al usar una VPN se le aplica una capa de cifrado y autenticación a ésta autopista, para proteger el tráfico de red por donde viajan los datos, esta técnica se llama VPN Tunneling, que precisamente crea éste túnel o canal de comunicación dentro de una red de computadoras. De esta manera, lo que realmente está ocurriendo en la comunicación, es que los datos que se envían están cubiertos. Todos los nodos intermedios que participen en la comunicación van a interactuar con el paquete, pero solamente al final de la comunicación la información podrá ser descubierta y descifrada para su uso, por su parte la capa de autenticación verificará la autenticidad de los usuarios y restringirá el acceso a quienes no estén autorizados.



Figura 3. Esquema de una VPN

De esta manera las fases de una conexión VPN son las siguientes:

1) Autenticación

En esta fase, los datos de los paquetes son primeramente encapsulados, básicamente envueltos dentro de otros paquetes en procesos que van añadiendo nueva información y aumentando el tamaño del paquete enviado. Estos datos son información sobre la máquina emisora, bits de CRC (para garantizar la integridad del paquete), la propia información enviada y otros datos.

El dispositivo inicia la negociación con el servidor VPN -saludo- y este responde con una respuesta acorde (ACK o Acknowledgement). Entonces, el servidor pide las credenciales del usuario para comprobar su identidad real.

2) Tunnelización

Después de la fase de autenticación, se crea un túnel que proporciona un camino seguro para la información entre dos puntos. Cualquier dato que se envíe mediante dicho túnel estará protegido del resto de paquetes de la red mediante cifrado.

3) Cifrado

Después de haber creado el túnel con éxito, se puede enviar cualquier información a través de él, pero no se puede dar la información por segura si se tiene está conectado a un servicio de VPN gratuito, que estará utilizada por más usuarios.

Por eso, se debe cifrar los paquetes transmitidos por una máquina ANTES de que alcancen el túnel, impidiendo así que el resto de usuarios del túnel pueda acceder a la información (pues siguen el mismo camino de red que todos paquetes).

2.2.2. Usos de una VPN

- ✓ Acceso Remoto: Una conexión VPN se utiliza para proveer acceso directo a una red corporativa para un usuario que no está dentro del ámbito físico que cubre dicha red corporativa. Evidentemente, el usuario remoto consigue los mismos permisos y funcionamiento que cualquier usuario local de la red.
- ✓ Sitio a sitio: Una red privada virtual también se utiliza para proporcionar un entorno de red homogéneo para una empresa que tiene oficinas en diferentes ubicaciones geográficas. Esto permite gestionar de manera lógica las diferentes subredes como una sola, para compartir los recursos sin importar la ubicación. (Cosoi, s.f.)

2.2.3. VPN de capa 2

Las VPN's de capa 2 (L2VPN) se clasifican como "Provider-Provisioned". Esto significa que la responsabilidad de crear y administrar los túneles para el tráfico privado entre los sitios recae

en el proveedor. El proveedor utiliza MPLS como medio de transporte para crear túneles entre los sitios privados. (Auben Networks, s.f.)

2.2.3.1. VPN basadas en BGP

En esta arquitectura el protocolo de Gateway de borde (BGP) es usado para el intercambio de la información de enrutamiento.

Las VPN's se separan en dos grandes grupos: VPN Overlay y VPN igual-a-igual. La VPN Overlay es aquella donde el proveedor de servicios emula una línea dedicada mediante un circuito virtual (VC) entre los sitios remotos. Los circuitos virtuales (VC) son propios de las tecnologías ATM y Frame-relay. En el modelo VPN igual-a-igual se intercambia información de enrutamiento entre la VPN y la WAN. En la figura 4 se muestran las tecnologías usadas para los dos modelos de VPN.

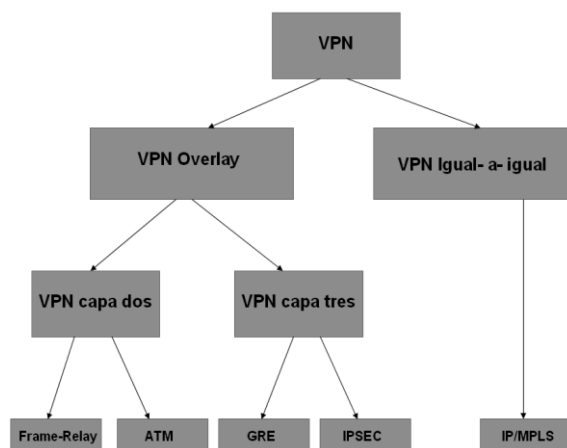


Figura 4. Tipos de redes privadas virtuales

Las ventajas que ofrece una VPN MPLS por sobre las VPN Overlay son escalamiento al agregar un nuevo sitio en la VPN ya que no se tiene que reconfigurar todos los demás sitios al

agregarse uno nuevo, no se necesitan IP públicas para la comunicación entre sitios remotos (solo para salir a Internet). Ofrece una topología de malla completa, ofrece una plataforma de rápido despliegue de nuevos servicios como telefonía IP, multimedia, Intranet, extranet, capacidad para implementar QoS, capacidad de implementa ingeniería de tráfico (TE).

La RFC 2547 define un mecanismo el cual permite que los proveedores de servicios utilizar su backbone IP/MPLS para proporcionar servicios de VPN a sus clientes. La RFC 2547 también se conoce como VPN's MPLS MP-BGP porque MP-BGP se utiliza para el intercambio de la información de enrutamiento entre sitios remotos sobre la WAN y MPLS se utiliza para enviar tráfico de VPN.

Un sitio cliente está conectado a la red del proveedor de servicios (SP) por una interfaz. El SP asocia la interfaz a una tabla de enrutamiento y envío VPN (VRF) en un PE. En la arquitectura VPN MPLS MP-BGP se definen 3 tipos de dispositivos el CE, el PE y el P.

El dispositivo de borde cliente (CE) puede ser un conmutador de capa dos pero típicamente el CE es un enrutador que establece adyacencia con el PE directamente conectado. Después de establecer adyacencia el enrutador CE anuncia las rutas locales del sitio VPN y aprende rutas remotas desde el PE.

El enrutador de borde del proveedor (PE) intercambia información de enrutamiento con el enrutador CE. Para intercambiar la información de enrutamiento se puede usar enrutamiento estático a algunos protocolos de enrutamiento como RIP, OSPF, EIGRP o EBGp. Cada enrutador PE mantiene una VRF para cada uno de los sitios directamente conectado. El enrutador interno proveedor (P) es cualquier enrutador en la red del proveedor que no une a CE's. Los enrutadores P

funcionan como LSR de MPLS enviando y conmutando etiquetas. En la figura 5 se muestra la arquitectura de una VPN MPLS MP-BGP.

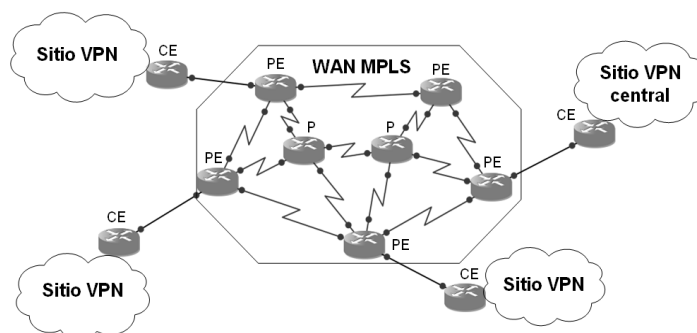


Figura 5. Tipos de redes privadas virtuales

Dos flujos de control son necesarios para el establecimiento de la VPN. El primer flujo de control es el intercambio de la información de enrutamiento entre sitios remotos esto es realizado a través del protocolo MP-BGP. El segundo flujo de control es el establecimiento de la ruta conmutada por etiquetas (LSP) vía el protocolo LDP, luego ocurre el tráfico de datos entre sitios remotos. (Icaran)

2.3. VPLS Virtual Private LAN Service

En una VPLS, los clientes no están todos conectados a una sola LAN; los clientes pueden extenderse a través de un área metropolitana o amplia. En esencia, una VPLS conecta varias LAN individuales a través de una red conmutada por paquetes para aparecer y funcionar como una sola LAN.

El mecanismo de señalización utiliza BGP como protocolo de plano de control, es decir, para el autodescubrimiento de miembros VPLS, para la configuración y desmontaje de las pseudo-rutas que constituyen una instancia de VPLS determinada. (Lesserre & Kompella, 2007)

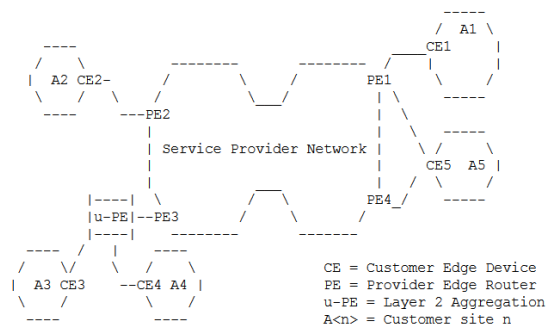


Figura 6. Ejemplo de una VPLS

2.3.1. Terminología

Los dispositivos PE y u-PE son "VPLS-aware", lo que significa que saben que se está realizando un servicio de VPLS. Llamaremos a estos dispositivos de borde VPLS, que podrían ser un PE o un u-PE.

Por el contrario, el dispositivo CE (puede ser operado por el SP o por cualquier cliente) no sabe de VPLS; en cuanto al CE en cuestión, está conectado a los demás CE's en la VPLS mediante una red conmutada de capa 2. Esto significa que no debe haber cambios en un CE dispositivo, ya sea al hardware o al software, para ofrecer VPLS.

El término "demultiplexor" se refiere a un identificador en un paquete de datos que identifica tanto el VPLS al que pertenece el paquete como el ingreso PE. En este documento, el demultiplexor es una etiqueta MPLS. El término "VPLS" se referirá al servicio, así como a una particular creación de instancias del servicio (es decir, una LAN emulada); debería ser claro del contexto el uso que se pretende. (Lesserre & Kompella, 2007)

2.3.2. Interacción

VPLS es un "Servicio LAN" en el que los dispositivos CE que pertenecen a VPLS pueden interactuar a través de la red SP como si estuvieran conectados por una LAN.

Los dispositivos PE interactúan para "descubrir" a todos los demás PE que participan en el mismo VPLS, y para intercambiar demultiplexores. Estas interacciones son controladas por el plano de control.

Los u-PE interactúan con PE para establecer conexiones con PE remotos o u-PE en el mismo VPLS. Esta interacción es controlada por el control. Los dispositivos PE pueden participar simultáneamente tanto en VPLS como en VPN IP. Estos son servicios independientes, y la información intercambiada para cada tipo de servicio se mantiene separado como la capa de red.

La información de accesibilidad (NLRI) utilizada para este intercambio tiene diferente Address Family Identifiers (AFI) e Identificadores de Familia de Direcciones Subsecuentes (SAFI). En consecuencia, una implementación DEBE mantener un almacenamiento de enrutamiento separado para cada servicio. Sin embargo, múltiples servicios pueden usar los mismos túneles subyacentes; la etiqueta VPLS o VPN se usa para demultiplexar los paquetes que pertenecen a diferentes servicios. (Lesserre & Kompella, 2007)

2.3.3. Plano de Control

Hay dos funciones principales del plano de control VPLS que lo constituyen:

- Autodescubrimiento, y
- Configuración y desmontaje de las pseudo-rutas

2.3.4. Auto-descubrimiento

El descubrimiento se refiere al proceso de encontrar todos los PE que participan en un VPLS dado. Un PE se puede configurar con las identidades de todos los otros PE en un VPLS dado, o el PE puede usar algún protocolo para descubrir los otros PE. El último se llama autodescubrimiento.

En el enfoque de autodescubrimiento, cada PE "descubre" qué otros PE son parte de un VPLS dado por medio de algún protocolo, en este caso BGP.

Esto permite que la configuración de cada PE consista solo en la identidad de la instancia de VPLS establecida en este PE, no la identidad de cualquier otro PE en esa instancia de VPLS; eso se descubre automáticamente.

Además, cuando cambia la topología de un VPLS, solo los PE afectados cambian de configuración; otros PE averiguan automáticamente sobre cambios y se adaptan. (Lesserre & Kompella, 2007)

2.3.4.1. Funciones

Un PE que participa en un VPLS dado, debe ser capaz de anunciar otros PE en VPLS que también son miembros de VPLS. Un PE también debe tener un medio para declarar que ya no participa en un VPLS.

Para hacer ambas cosas, el PE debe tener un medio para identificar un VPLS y un medio por el cual comunicarse con todos los demás PE. (Lesserre & Kompella, 2007)

2.3.4.2. Especificación del Protocolo

El mecanismo específico para autodescubrimiento se basa en L2VPN sobre túneles y VPN IP sobre BGP/MPLS; usa comunidades extendidas de BGP para identificar miembros de un VPLS. Un mecanismo de autodescubrimiento más genérico es BGP. La comunidad extendida específica utilizada es la Ruta Objetivo (Route Target RT), cuyo formato se describe en atributos BGP. Usa la semántica de objetivos de ruta; su uso en VPLS es idéntico.

Como se ha supuesto que los VPLS están completamente engranados, una sola ruta RT objetivo es suficiente para un VPLS dada, y en efecto, que RT es el identificador para VPLS.

Un PE anuncia (típicamente a través de I-BGP) que pertenece a VPLS sus NLRI con Route Target RT, y actúa sobre esto aceptando NLRI de otros PE que tienen Ruta de destino RT. Un PE anuncia que ya no participa en VPLS retirando todos los NLRI que había anunciado con Route Target RT. (Lesserre & Kompella, 2007)

2.3.4.3. Señalización

Una vez hecho el descubrimiento, cada par de PE en un VPLS debe ser capaz de establecer (y derribar) pseudo-rutas entre sí, es decir, intercambio (swap) de demultiplexores. Este proceso se conoce como señalización. La señalización también se usa para transmitir ciertas características del pseudo-rutas que un PE establece para un VPLS dado.

Un demultiplexor se usa para distinguir entre varios diferentes flujos de tráfico transportados por un túnel, cada flujo posiblemente representa un servicio diferente. En el caso de VPLS, el demultiplexor no solo dice a qué VPLS específico pertenece un paquete, también

identifica el ingreso PE. La información anterior se usa para reenviar el paquete; la última información se utiliza para el aprendizaje de direcciones MAC. El demultiplexor descrito aquí es una etiqueta MPLS. Sin embargo, hay que tener en cuenta que los túneles de PE a PE no necesitan ser MPLS túneles. (Lesserre & Kompella, 2007)

2.3.4.4. Conceptos

El VPLS BGP NLRI descrito a continuación, con un nuevo AFI y SAFI se usa para intercambiar miembros de VPLS y demultiplexores.

Un VPLS BGP NLRI tiene los siguientes elementos de información: un VE ID, un VE desplazamiento de bloque, un tamaño de bloque VE y una base de etiqueta.

Un PE que participa en un VPLS debe tener al menos un ID de VE. Si el PE es el VE, generalmente tiene un ID de VE.

Los identificadores VE generalmente son asignados por el administrador de la red. Su alcance es local para un VPLS. Una ID de VE determinada debe pertenecer a una sola PE, a menos que un CE sea multitarjeta.

El formato del VPLS NLRI se proporciona a continuación. El AFI es el L2VPN AFI (a ser asignado por IANA), y el SAFI es el SAFI VPLS. (Lesserre & Kompella, 2007)



Figura 7. BGP NLRI para información de VPLS

2.3.4.5. Capacidades de PE de Señalización

El siguiente atributo extendido, "Layer2 Info Extended Comunidad ", se utiliza para señalar información de control sobre pseudo-rutas que se configurarán para un VPLS dado. Esta información incluye el tipo de encapsulación (tipo de encapsulación en los pseudo-rutas), control Banderas (información de control con respecto a las pseudo-rutas) y el Máximo Unidad de transmisión (MTU) para ser utilizado en las pseudo-rutas.



Figura 8. Comunidad ampliada de capa 2 INFO

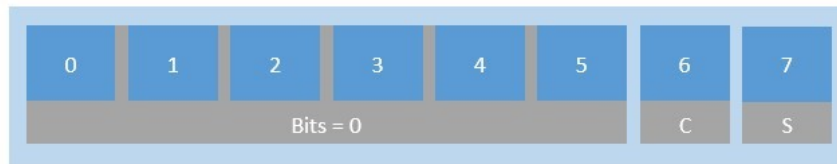


Figura 9. Comunidad ampliada de capa 2 INFO

Con referencia a la figura 8, los siguientes bits en las Banderas de control están definidos; los bits restantes del 0 al 5, *debe* ponerse a cero al enviar y *debe* ser ignorado al recibir esta comunidad. (Lesserre & Kompella, 2007)

Significado del nombre:

- C: Si se establece en 1 (0), la palabra de control *debe* (no) estar presente cuando enviando paquetes VPLS a este PE
- S: Si se establece en 1 (0), la entrega secuenciada de fotogramas es (no) requerido al enviar paquetes VPLS a este PE

2.3.5. Plano de Datos

Esta sección trata sobre dos aspectos del plano de datos para PE y u-PE que implementan VPLS: encapsulación y reenvío.

2.3.5.1. Encapsulación

Las tramas Ethernet recibidas de los dispositivos CE están encapsuladas para transmisión a través de la red de paquetes conmutados que conecta los PE.

2.3.5.2. Reenvío

Los paquetes VPLS se clasifican como pertenecientes a un servicio determinado y asociada a la tabla de reenvío basada en la interfaz sobre el cual el paquete es recibido. Los paquetes se envían en el contexto de instancia de servicio basada en la dirección MAC de destino. El anterior la asignación está determinada por la configuración. Este último es el foco de esta sección. (Lesserre & Kompella, 2007)

2.3.5.3. Aprendizaje de direcciones MAC

La característica distintiva clave de VPLS es que es un servicio multipunto. Esto significa que todo el Servicio Proveedor de red debe aparecer como un solo puente lógico de aprendizaje para cada VPLS que la red de SP soporta. Los puertos lógicos para el "puente" SP son los puertos del cliente, así como los pseudo-rutas en un VE. Así como un puente de aprendizaje aprende direcciones MAC en sus puertos, el puente SP debe aprender las direcciones MAC en sus VE's.

El aprendizaje consiste en asociar direcciones MAC de origen de paquetes con los puertos (lógicos) a los que llegan; esta asociación es la Forwarding Information Base (FIB). El FIB se usa para reenviar paquetes. Por ejemplo, supongamos que el puente recibe un paquete con la dirección MAC de origen S en el puerto (lógico) P. Si posteriormente, el bridge recibe un paquete con la dirección MAC de destino S, sabe que debería enviar el paquete al puerto P.

Si un VE aprende una dirección MAC de origen S en el puerto lógico P, luego ve S en un puerto diferente P', entonces el VE *debe* actualizar su FIB para reflejar el nuevo puerto P'. (Lesserre & Kompella, 2007)

2.3.5.4. Tiempo de vida

Los VPLS PE DEBEN tener un mecanismo de caducación para eliminar una dirección MAC asociado con un puerto lógico, más o menos lo mismo que los puentes de aprendizaje. Esto es necesario para que una dirección MAC se pueda volver a aprender si se "mueve" desde un puerto lógico a otro puerto lógico, ya sea porque estación a la que pertenece esa dirección MAC realmente se ha movido o porque de un cambio de topología en la LAN que causa que esta dirección MAC llegar a un nuevo puerto. Además, el tiempo de vida reduce el tamaño de una tabla MAC VPLS solo para las direcciones MAC activas, en lugar de todas las MAC direcciones en ese VPLS.

La "edad" de una dirección MAC de origen S en un puerto lógico P es el momento que fue visto por última vez como una fuente MAC en el puerto P. Si la edad excede el tiempo de vida T, S *debe* eliminarse del FIB. Esto por supuesto significa que cada vez que se ve a S como una dirección MAC de origen en el puerto P, La edad de S se reinicia.

Una implementación *debe* proporcionar un botón configurable para establecer el envejecimiento tiempo T en base a VPLS. Además, una implementación *puede* acelerar el tiempo de vida de todos los anuncios de MAC en un VPLS si detecta alguna situación, como un cambio de topología de Spanning Tree en ese VPLS. (Lesserre & Kompella, 2007)

2.3.5.5. Inundación

Cuando un puente recibe un paquete a un destino que no está en su FIB, inunda el paquete en todos los otros puertos. Del mismo modo, un VE inundará los paquetes a un destino desconocido para todos los demás VE's en el VPLS. (Lesserre & Kompella, 2007)

2.3.5.6. Transmisión y multidifusión

Hay una dirección MAC de difusión bien conocida. Una trama de Ethernet cuya dirección MAC de destino es la dirección MAC de difusión que se debe enviar a todas las estaciones en ese VPLS. Esto se puede lograr de la misma manera que se usa para las inundaciones.

También hay un conjunto fácilmente reconocible de direcciones MAC de "multidifusión".

Las tramas Ethernet con una dirección MAC de multidifusión de destino *pueden* ser transmitidas a todas las estaciones; un VE también puede usar ciertas técnicas para restringir la transmisión de tramas de multidifusión a un conjunto más pequeño de receptores, aquellos que han indicado interés en el grupo de multidifusión correspondiente. (Lesserre & Kompella, 2007)

2.3.5.7. Desvío "Split Horizon"

Cuando un PE capaz de inundar (por ejemplo, PEx) recibe una trama Ethernet de difusión, o una con una dirección MAC de destino desconocido, debe inundar la trama. Si la trama llegó desde un CE adjunto, PEx debe enviar una copia de la trama a cada CE adjunta, así como a todas las otras PE que participan en el VPLS. Si, por otro lado, la trama llegó de otro PE (por ejemplo, PEy), PEx debe enviar una copia del paquete solo a los CE adjuntos. PEx *no debe* enviar la trama a otros PE, ya que PEy ya lo habría hecho. Esta noción se ha denominado reenvío de "horizonte dividido" y es una consecuencia de que los PE estén lógicamente engranados completamente para VPLS. Las reglas de reenvío de horizonte dividido se aplican a los paquetes de difusión y multidifusión, así como a los paquetes a una dirección MAC desconocida. (Lesserre & Kompella, 2007)

2.3.5.8. Aprendizaje calificado y no calificado

La clave para el aprendizaje MAC de Ethernet normal suele ser simplemente la dirección MAC (6 octetos). Esto se llama "aprendizaje no calificado". Sin embargo, también es posible que la clave para el aprendizaje incluya la etiqueta VLAN cuando esté presente; esto se llama "aprendizaje calificado".

En el caso de VPLS, el aprendizaje se realiza en el contexto de una instancia de VPLS, que normalmente corresponde a un cliente. Si el cliente usa etiquetas de VLAN, se pueden hacer las mismas distinciones de aprendizaje calificado y no calificado. Si la clave para aprender dentro de un VPLS es solo la dirección MAC, entonces este VPLS está funcionando bajo un aprendizaje no calificado. Si la clave para aprender es (etiqueta de VLAN del cliente + dirección MAC), este VPLS está funcionando bajo el aprendizaje calificado.

Elegir entre el aprendizaje calificado y el no calificado implica varios factores, el más importante de los cuales es si uno quiere un solo dominio de difusión global (no calificado) o un dominio de difusión por VLAN (calificado). Este último hace que las inundaciones y las transmisiones sean más eficientes, pero requiere tablas MAC más grandes. Estas consideraciones se aplican igualmente al reenvío de Ethernet normal y a VPLS. (Lesserre & Kompella, 2007)

2.3.5.9. Opciones de implementación

Al implementar una red que admite VPLS, el SP debe decidir qué funciones admite el dispositivo compatible con VPLS más cercano al cliente (el VE). El caso predeterminado descrito en este documento es que el VE es un PE. Sin embargo, hay una serie de razones por las que el VE podría ser un dispositivo que realiza todas las funciones de Capa 2 (como el aprendizaje e

inundación de direcciones MAC) y un conjunto limitado de funciones de Capa 3 (como la comunicación a su PE), pero, por ejemplo, no hace un descubrimiento en toda regla y una señalización de PE a PE. Tal dispositivo se llama "u-PE".

Como ambos casos tienen beneficios, a uno le gustaría poder "mezclar y combinar" estos escenarios. El mecanismo de señalización presentado aquí permite esto. Por ejemplo, en una red de proveedores dada, un PE puede estar conectado directamente a dispositivos CE, otro puede estar conectado a u-PE que está conectado a un CE, y un tercero puede estar conectado directamente a un cliente a través de algunas interfaces y a U-PE sobre otros.

Todos estos PE realizan descubrimiento y señalización de la misma manera. Cómo lo hacen aprendiendo y reenviando depende de si hay u-PE; sin embargo, este es un asunto local y no está señalizado. (Lesserre & Kompella, 2007)

2.4. EVPN Ethernet Virtual Private Network

2.4.1. Terminología

- Dominio de difusión: en una red en puente, el dominio de difusión corresponde a una LAN virtual (VLAN), donde una VLAN es típicamente representado por una sola ID de VLAN (VID) pero puede ser representado por varios VID donde se usa Shared VLAN Learning (SVL).
- Bridge Table: una instancia de un dominio de difusión en una MAC-VRF.
- EVI: una instancia de EVPN que abarca los dispositivos Provider Edge (PE) participando en ese EVPN.

- MAC-VRF: una tabla de enrutamiento y reenvío virtual para acceso a medios Direcciones de control (MAC) en un PE.
- Segmento Ethernet (ES): cuando el sitio del cliente (dispositivo o red) es conectado a uno o más PE a través de un conjunto de enlaces Ethernet, ese conjunto de enlaces se conoce como un 'segmento de Ethernet'.
- Identificador de segmento de Ethernet (ESI): es un identificador único distinto de cero que identifica un segmento de Ethernet.
- Etiqueta Ethernet: una etiqueta Ethernet identifica un dominio de difusión en particular, por ejemplo, una VLAN. Una instancia de EVPN consiste en uno o más dominios de difusión.
- LACP: protocolo de control de agregación de enlaces.
- MP2MP: multipunto a multipunto
- MP2P: multipunto a punto
- P2MP: punto a multipunto
- P2P: punto a punto.
- Single-Active Redundancy Mode: cuando solo un PE individual, entre todos los PE conectados a un segmento de Ethernet están autorizados a reenviar tráfico hacia/desde ese segmento de Ethernet para una VLAN dada, luego el segmento Ethernet está definido para operar en redundancia Single-Active modo.
- All-Active Redundancy Mode: cuando todos los PE conectados a un segmento Ethernet pueden reenviar el tráfico de unidifusión conocido a/desde ese segmento Ethernet para una VLAN dada, entonces el segmento Ethernet es definido para operar en el modo de redundancia All-Active. (Sajassi, y otros, 2015)

2.4.2. EVPN basado en BGP MPLS

Una instancia de EVPN comprende los dispositivos Edge del cliente (CE) que están conectados al proveedor Dispositivos de borde (PE) que forman el borde de la infraestructura de MPLS. Un CE puede ser un host, un enrutador o un conmutador. Los PE brindan virtual Conectividad en puente de Capa 2 entre los CE. Puede haber múltiples Instancias de EVPN en la red del proveedor.

Los PE pueden estar conectados por una ruta conmutada por etiqueta MPLS (LSP) infraestructura, que proporciona los beneficios de la tecnología MPLS, tales como re-encaminamiento rápido. Los PE también pueden estar conectados por una infraestructura IP, en cuyo caso IP / GRE, tunelización u otro túnel IP se puede utilizar entre los PE.

En una EVPN, el aprendizaje MAC entre PE no ocurre en el plano de datos (como sucede con el puente tradicional en VPLS), sucede en el plano de control. El aprendizaje del plano de control ofrece un mayor control sobre el proceso de aprendizaje de MAC, como restringir quién aprende qué, y la capacidad de aplicar políticas. Además, el plano de control elegido para anunciar información de accesibilidad MAC es multiprotocolo (MP) BGP (similar a IP VPN's). Esto proporciona flexibilidad y la capacidad de preservar la "virtualización" o el aislamiento de grupos de agentes que interactúan (hosts, servidores, máquinas virtuales) desde el uno al otro. En EVPN, los PE publican las direcciones MAC aprendidas de los CE que están conectados a ellos, junto con una etiqueta MPLS, hacia otros PE en el plano de control utilizando Multiprotocol BGP (MP-BGP). El aprendizaje del plano de control permite equilibrar la carga del tráfico hacia y desde CE que son multihomed a múltiples PE. Esto es además de cargar equilibrio a través del núcleo de

MPLS a través de múltiples LSP entre el mismo par de PE. En otras palabras, permite que los CE se conecten a múltiples puntos activos de apego. También mejora los tiempos de convergencia en el evento de ciertas fallas de red.

Sin embargo, el aprendizaje entre PE y CE se hace por el mejor método adecuado para el CE: aprendizaje de plano de datos, IEEE 802.1x, la capa de enlace Protocolo de descubrimiento (LLDP), IEEE 802.1aq, Protocolo de resolución de direcciones (ARP), plano de administración u otros protocolos. Es una decisión local si la tabla de reenvío de Capa 2 un PE se completa con todas las direcciones de destino MAC conocidas por el plano de control, o si el PE implementa un esquema basado en caché. Por ejemplo, la tabla de reenvío de MAC se puede llenar solo con el Destinos MAC de los flujos activos que transitan un PE específico.

Los atributos de política de EVPN son muy similares a los de IP-VPN. Una instancia de EVPN requiere un Router Designado (RD) único por MAC-VRF y uno o más Destinos de Ruta (RT) únicos a nivel mundial. Un CE se conecta a un MAC-VRF en un PE, en una interfaz Ethernet que puede configurarse para una o más etiquetas Ethernet, por ejemplo, ID de VLAN. Algunos escenarios de implementación garantizan la exclusividad de los ID de VLAN en instancias EVPN: todos los puntos de conexión para una instancia de EVPN dada usan la misma ID de VLAN, y ninguna otra instancia de EVPN usa esta ID de VLAN. (Sajassi, y otros, 2015)

2.4.3. Segmento Ethernet ES

Cuando el sitio de un cliente está conectado a uno o más PE a través de un conjunto de enlaces Ethernet, este conjunto de enlaces Ethernet constituye un "Segmento Ethernet". Para un sitio multihomed, cada Segmento Ethernet (ES) se identifica por un identificador único distinto de

cero llamado Identificador de Segmento Ethernet (ESI). Un ESI está codificado como un entero de 10 octetos en línea con el octeto más significativo enviado primero. Los siguientes dos valores de ESI están reservados:

- ESI 0 denota un sitio con un solo hogar.
- ESI {0xFF} (repetido 10 veces) se conoce como MAX-ESI y está reservado.

En general, un segmento Ethernet DEBE tener un ESI no reservado que es único en toda la red (es decir, en todas las instancias de EVPN en todos los PE's). Si el CE (s) que constituye un segmento de Ethernet es (son) administrado por el operador de la red, entonces se debe garantizar la exclusividad de ESI; sin embargo, si el (los) CE (s) no se (son) administra (n), entonces el operador DEBE configurar un ESI único en toda la red para ese segmento de Ethernet. Es necesario para permitir el descubrimiento automático de segmentos Ethernet y elección del reenviador designado (DF).

En una red con CE gestionados y no gestionados, ESI tiene el siguiente formato:



Figura 10. Trama ESI

T (Tipo ESI) es un campo de 1 octeto (el octeto más significativo) que especifica el formato de los 9 octetos restantes (valor de ESI). (Sajassi, y otros, 2015)

2.4.4. ID de etiqueta de Ethernet (Ethernet Tag ID)

Un identificador de etiqueta Ethernet es un campo de 32 bits que contiene 12 bits o 24 bits que identifica un dominio de difusión particular (por ejemplo, una VLAN) en una instancia de EVPN. El identificador de 12 bits se llama la ID de VLAN (VID). Una instancia de EVPN consiste en uno o más dominios de difusión (una o más VLAN). Las VLAN se asignan a una determinada instancia EVPN por el proveedor del servicio EVPN. Una VLAN dada puede estar representado por múltiples VID. En tales casos, los PE participan en esa VLAN para una instancia determinada, EVPN son responsables para realizar la traducción de la identificación de VLAN a/desde un dispositivo CE localmente conectado.

Si una VLAN está representada por un solo VID en todos los dispositivos PE participando en esa VLAN para esa instancia de EVPN, entonces no hay necesidad de traducción VID en las PE. Además, alguna implementación de escenarios garantizan la exclusividad de los VID en todas las instancias de EVPN; todas los puntos de conexión para una instancia determinada de EVPN usan el mismo VID, y ninguna otra instancia de EVPN usa ese VID. Esto permite el RT (s) para cada instancia EVPN se derivará automáticamente del VID correspondiente.

Las siguientes subsecciones discuten la relación entre la difusión dominios (por ejemplo, VLAN), identificaciones de etiquetas Ethernet (por ejemplo, VID) y MAC-VRF como así como la configuración de la identificación de etiqueta Ethernet, en los diversos EVPN BGP rutas. (Sajassi, y otros, 2015)

El siguiente valor de identificación de etiqueta Ethernet está reservado:

- Ethernet Tag ID {0xFFFFFFFF} se conoce como MAX-ET.

2.4.4.1. Interfaz de servicio basada en VLAN

Con esta interfaz de servicio, una instancia de EVPN consiste en solo un dominio de difusión única (por ejemplo, una sola VLAN). Por lo tanto, hay un mapeo uno-a-uno entre un VID en esta interfaz y un MAC-VRF.

Como un MAC-VRF corresponde a una sola VLAN, consta de un solo tabla de puente correspondiente a esa VLAN. Si la VLAN está representada por múltiples VID (por ejemplo, un VID diferente por segmento de Ethernet por PE), entonces cada PE debe realizar la traducción VID para los marcos destinados a su (s) segmento (s) de Ethernet. En tales escenarios, las tramas de Ethernet transportadas a través de una red MPLS / IP DEBE permanecer etiquetado con el originando VID, y una traducción VID DEBE ser compatible con los datos ruta y DEBE realizarse en la disposición PE. La etiqueta de Ethernet ID en todas las rutas EVPN DEBE establecerse en 0. (Sajassi, y otros, 2015)

2.4.4.2. Interfaz de servicio de paquete VLAN

Con esta interfaz de servicio, una instancia de EVPN corresponde a múltiples dominios de difusión (por ejemplo, VLAN múltiples); sin embargo, solo una tabla de bridge se mantiene por MAC-VRF, lo que significa VLAN múltiples comparte la misma tabla de bridge. Esto implica que las direcciones MAC DEBEN ser únicas en todas las VLAN para ese EVI con el fin de este servicio para funcionar. En otras palabras, hay una asignación muchos a uno entre las VLAN y un MAC-VRF, y el MAC-VRF consiste en una sola tabla de bridge.

Además, una sola VLAN debe estar representada por un único VID: por ejemplo, no se permite ninguna traducción VID para este tipo de interfaz de servicio.

Los cuadros encapsulados en MPLS DEBEN permanecer etiquetados con el originador VID. La traducción de etiquetas NO está permitida. La identificación de la etiqueta de Ethernet en todas las rutas EVPN DEBEN establecerse en 0. (Sajassi, y otros, 2015)

2.4.4.3.Rutas BGP EVPN

Información de alcance de la capa de red (NLRI). El formato de EVPN NLRI es el siguiente:

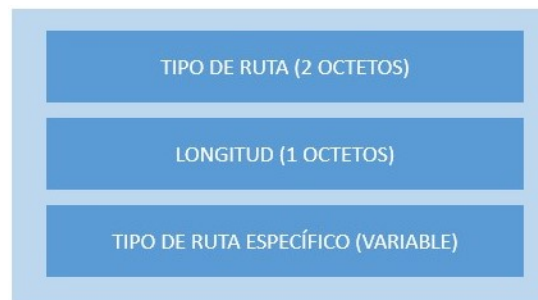


Figura 11. Trama NLRI v2

- El campo Tipo de ruta define la codificación del resto de la EVPN NLRI (tipo ruta específica EVPN NLRI).
- El campo Longitud indica la longitud en octetos del Tipo de ruta campo específico de EVPN NLRI.

Tipos de rutas:

- a) Ruta de descubrimiento automático de Ethernet (AD)
- b) Ruta de anuncio MAC/IP
- c) Ruta de etiqueta Ethernet multidifusión inclusiva
- d) Ruta del segmento de Ethernet

2.4.4.4. Ruta de descubrimiento automático

Un tipo de ruta Ethernet AD específico EVPN NLRI consiste en lo siguiente:



Figura 12. Ruta de auto-descubrimiento

A los efectos del procesamiento de la clave de ruta BGP, solo el Ethernet el identificador de segmento y la identificación de etiqueta Ethernet se consideran parte del prefijo en el NLRI. El campo Etiqueta MPLS debe tratarse como un atributo de ruta en lugar de ser parte de la ruta. (Sajassi, y otros, 2015)

2.4.4.5. Ruta de anuncio MAC/IP

El tipo de ruta de anuncio MAC/IP NLRI específico consiste en:

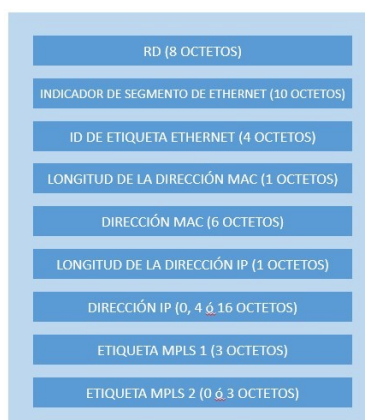


Figura 13. Ruta de anuncio

A los efectos del procesamiento de la clave de ruta BGP, solo la etiqueta Ethernet ID, longitud de la dirección MAC, dirección MAC, longitud de la dirección IP e IP. Los campos de dirección se consideran parte del prefijo en el NLRI. Los campos Identificador de segmento Ethernet, Etiqueta MPLS1 y Etiqueta MP2 deben ser tratados como atributos de ruta en lugar de ser parte de la "ruta". Las longitudes de direcciones IP y MAC están en bits. (Sajassi, y otros, 2015)

2.4.4.6. Ruta de etiqueta de Ethernet de multidifusión inclusiva

Un tipo de ruta de etiqueta Ethernet de multidifusión inclusiva EVPN NLRI específica consiste en lo siguiente:



Figura 14. Ruta de etiqueta de Ethernet de multidifusión inclusiva

2.4.4.7. Ruta del segmento de Ethernet

Un tipo de ruta de segmento Ethernet EVPN NLRI específico consiste en lo siguiente:

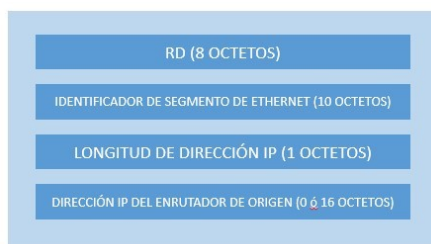


Figura 15. Ruta del segmento Ethernet

2.4.5. Funciones de Multihoming

Esta sección trata sobre las funciones, procedimientos y BGP asociado rutas utilizadas para soportar multihoming en EVPN. Esto cubre ambos escenarios de dispositivo multihomed (MHD) y red de multihome (MHN). (Sajassi, y otros, 2015)

2.4.5.1. Autodescubrimiento del segmento MultiHomed Ethernet

Los PE conectados al mismo segmento Ethernet pueden descubrir automáticamente entre sí con una configuración mínima o nula a través del intercambio de la ruta del Segmento Ethernet. (Sajassi, y otros, 2015)

2.4.5.2. Convergencia rápida

En EVPN, la accesibilidad de direcciones MAC se aprende a través del plano de control BGP sobre la red MPLS. Como tal, en ausencia de cualquier mecanismo de protección, el tiempo de convergencia de la red es una función del número de rutas de anuncios MAC/IP que debe retirar el PE que se encuentra con falla. Para entornos de gran escala, este esquema produce una convergencia lenta.

Para solventar este inconveniente, EVPN define un mecanismo eficiente y de señal rápida, a los nodos PE, la necesidad de actualizar su reenvío de tablas ante la ocurrencia de una falla en la conectividad a un segmento Ethernet.

Esto es que cada PE anuncie un conjunto de uno o más AD por rutas ES para cada segmento de Ethernet adjunto localmente. Un PE puede necesitar anunciar más de un Ethernet AD por ruta ES para una ES dada porque el ES puede estar en una multiplicidad de EVI y los RT para todos

estos EVI pueden no ajustarse en una sola ruta. Publicando un conjunto de rutas Ethernet AD por ES para el ES permite que cada ruta contenga un subconjunto del conjunto completo de RT's. Cada AD de Ethernet por ruta ES es diferente de las otras rutas en el conjunto por un Distinguidor de ruta diferente (RD). (Sajassi, y otros, 2015)

2.4.5.3. Construcción de AD de Ethernet por ruta de segmento de Ethernet

El Distinguidor de ruta (RD) *debe* ser un tipo 1. Los campos de valor comprenden una dirección IP (por lo general la dirección de bule invertido) seguido de un número invertido exclusivo para el PE.

El identificador de Segmento Ethernet *debe* ser una entidad de 10 octetos. La ruta Ethernet AD no es necesaria cuando el identificador de segmento se establece en 0. La identificación de la etiqueta Ethernet *debe* establecerse en MAX-ET. La etiqueta MPLS en NLRI *debe* establecerse en 0. La comunidad extendida de ESI Label *debe* incluirse en la ruta. Si se desea el modo de redundancia completamente activo, luego el bit "Single-Active" en los indicadores de la comunidad ampliada de ESI Label *debe* establecerse en 0 y la etiqueta MPLS en la comunidad extendida debe establecerse en una MPLS válida del valor de la etiqueta como la etiqueta ESI y *debe* tener el mismo valor en cada AD Ethernet.

Si se desea el modo de redundancia Single-Active, entonces el "Single-Active" el bit en las banderas de la comunidad extendida ESI LABEL *debe* establecerse en 1 y la etiqueta ESI *debería* establecerse en un valor de etiqueta MPLS válido. (Sajassi, y otros, 2015)

2.4.5.3.1. Destinos de ruta AD de Ethernet

Cada AD de Ethernet por ruta ES *debe* llevar uno o más destinos de ruta (RT). El conjunto de rutas AD de Ethernet por ES *debe* llevar el conjunto completo de RT para todas las instancias de EVPN a las que el Segmento de Ethernet pertenece. (Sajassi, y otros, 2015)

2.4.6. Determinación de la accesibilidad a direcciones MAC de unidifusión

Los PE reenvían paquetes que reciben en base a la dirección MAC de destino. Esto implica que los PE deben poder aprender cómo llegar a una dirección MAC dada, de unidifusión de destino. Existen dos componentes para el aprendizaje de direcciones MAC: “aprendizaje local” y “aprendizaje remoto”. (Sajassi, y otros, 2015)

2.4.6.1. Aprendizaje local

Un PE particular es capaz de aprender las direcciones MAC de los CE que están conectados a ella. Esto se conoce como aprendizaje local.

Los PE en una instancia de EVPN particular *deben* soportar el plano de datos local. Un PE debe ser capaz de aprender las direcciones MAC en el plano de datos cuando recibe paquetes, como los siguientes CE en la red:

- Solicitudes de DHCP
- Una solicitud de ARP para su propio MAC
- Una solicitud de ARP para un par

Alternativamente, los PE *pueden* aprender las direcciones MAC de los CE en el plano de control o a través de la integración del plano de gestión entre los PE y los CE.

Hay aplicaciones donde una dirección MAC que se puede alcanzar a través de un PE dado que en un segmento localmente conectado puede moverse de modo que sea alcanzable a través de otro PE en otro segmento. Esto se conoce como Movilidad MAC. (Sajassi, y otros, 2015)

2.4.6.2. Aprendizaje remoto

Un PE en particular debe ser capaz de determinar cómo enviar tráfico a direcciones MAC que pertenecen o están detrás de los CE conectados a otros PE, es decir, a CE remotos o hosts detrás de CE remotos, llamadas a tales MAC como direcciones MAC remotas.

Para que un PE aprenda direcciones remotas en el plano de control, cada PE publica la dirección MAC que aprende de sus CE adjuntos localmente en el plano de control, a todos los otros PE en esta instancia EVPN, utilizando MP-BGP y, específicamente, la ruta de publicación MAC/IP. (Sajassi, y otros, 2015)

2.4.7. Movilidad MAC

Es posible para un host determinado o estación final, pasar de un segmento Ethernet a otro; esto es denominado (Movilidad MAC), y es diferente de la situación de multihoming en la que puede alcanzar una dirección MAC dada a través de múltiples PE para el mismo segmento de Ethernet. En un movimiento MAC, ahí serían dos conjuntos de rutas de anuncios MAC/IP, en un conjunto con el nuevo segmento Ethernet y un conjunto con el segmento Ethernet anterior y la dirección MAC parece ser accesible a través de cada uno de estos segmentos.

Para permitir que todos los PE en la instancia EVPN estén correctos, se debe determinar la ubicación de la dirección MAC, todos los anuncios de que sea alcanzable a través del segmento Ethernet anterior *debe* ser retirado por los PE, para el segmento Ethernet anterior que lo anunciaba.

Si el aprendizaje local se realiza utilizando el plano de datos, estos PE no son capaces de detectar que la dirección MAC se ha movido a otro segmento Ethernet y la recepción de rutas de anuncios MAC/IP, con el atributo de comunidad extendida MAC Mobility, otros PE sirven como disparador para que estos PE retiren sus anuncios. Si el aprendizaje local se realiza utilizando el control o la gestión, estas interacciones sirven como el disparador de estos PE para retirar sus anuncios.

En una situación donde hay múltiples movimientos MAC, posiblemente entre los mismos dos segmentos Ethernet, puede haber retiros múltiples y re-anuncios. Para asegurar que todos los PE en la instancia de EVPN reciben todos estos correctamente a través de la infraestructura BGP intermedia, introduciendo un número de secuencia en el atributo de comunidad extendida MAC Mobility es necesaria.

Para procesar los eventos de movilidad correctamente, una implementación *debe* manejar escenarios en los que se produce el envoltorio de número de secuencia.

Cada evento de movilidad MAC para una dirección MAC dada contendrá un número de secuencia que se establece utilizando las siguientes reglas: (Sajassi, y otros, 2015)

- Un PE que anuncia una dirección MAC por primera vez lo anuncia sin ningún atributo de comunidad extendida MAC Mobility.

- Un PE que detecta una dirección MAC localmente unida para la cual anteriormente recibió una ruta de publicación MAC/IP con un identificador de segmento Ethernet que anuncia la dirección MAC en una ruta MAC/IP publicitaria etiquetada con una comunidad extendida MAC Mobility con una secuencia número uno mayor que la secuencia.
- Un PE que detecta una dirección MAC localmente unida para la cual localmente recibe una ruta de anuncio MAC/IP con el mismo identificador de segmento Ethernet no cero, lo anuncia con:
 - Ningún atributo de comunidad extendida de MAC Mobility, si la ruta no lleva dicho atributo.
 - Un atributo de comunidad extendida MAC Mobility con el número de secuencia igual o más alto de los números de secuencia.
- Un PE detecta una dirección MAC localmente unida para la cual anteriormente recibió una ruta de anuncio MAC/IP con el mismo cero.

2.5. Fundamentación

2.5.1. Aspectos Generales

En esta investigación es necesario el acople de hardware y software para simular un ambiente idóneo de la topología de red MPLS.

2.5.1.1. Hardware

Para realizar la simulación dentro del entorno GSN3 es necesario conocer las características de hardware que los dispositivos de red poseen en un ambiente real para poder representarlas con la mayor similitud posible. Los dispositivos de red que se eligieron para esta investigación son:

- ✓ Router MX 80: Con todas las funciones para aplicaciones empresariales y de proveedores de servicios en instalaciones con espacio y energía limitados. Los enrutadores MX80 están optimizados para WAN empresarial, Interconexión de centros de datos (DCI), agregación de sucursales y aplicaciones de campus.

Equipados con tres puertos 10GbE incorporados y dos ranuras para tarjeta de interfaz modular (MIC), los enrutadores MX80 permiten una conectividad de red flexible. También se encuentra disponible una ranura MS-MIC en el MX80 para soporte de servicios dedicados. 8 MB boot flash; 4 GB y NAND flash storage; 2 GB de DDR2 RAM. Los enrutadores MX80 permiten una escala rentable de "pago a medida que crece" para satisfacer los requisitos cambiantes del mercado. Esta escala y flexibilidad son evidentes en la gran variedad de casos de uso de la Serie MX que se han demostrado en las redes más exigentes del mundo. (Juniper, s.f.)



Figura 16. Router MX 80

- ✓ Cisco C2691: Estos modelos ofrecen desempeño extendido, alta densidad, desempeño de seguridad mejorado y soporte de aplicaciones concurrentes para satisfacer las crecientes demandas de las sucursales.

Posee un módulo de red, dos AIM's tres WIC's, dos interfaces FastEthernet memoria 256 MB DRAM, flash memory 128MB



Figura 17. Cisco C2691

- ✓ Ordenador: Toshiba Satélite L55 B5338 con procesador Core i5, disco duro de estado sólido SSD de 500GB, memoria 16GB RAM DDR3.



Figura 18. Ordenador

2.5.1.2. Software

2.5.1.2.1. JUNOS (Juniper Network Operating System)

Junos OS (más formalmente el sistema operativo de red de Juniper) es el sistema operativo basado en FreeBSD utilizado en los enrutadores de hardware de Juniper Networks. Es un sistema operativo que se utiliza en los dispositivos de enrutamiento, conmutación y seguridad de Juniper. Entre los beneficios de JUNOS se incluye:

- Diseño modular: todos los procesos y componentes de una configuración de red de Juniper están protegidos entre sí. Un módulo que falla no tendrá efecto en el resto del sistema.
- Compatibilidad con un solo tren: cada conmutador, enrutador u otro producto de Juniper ejecuta el mismo sistema JUNOS. El sistema está construido para una interoperabilidad simple en todo el sistema.



Figura 19. Logo JunOS

Las versiones JUNOS capaces de soportar las dos tecnologías de VPN's para el estudio de esta investigación con el vendor Juniper son:

- ✓ Junos 14.2R6

- ✓ Junos 16.1R1
- ✓ Junos 15.1.x
- ✓ Junos VMX

La interfaz de línea de comandos (CLI) de Junos es la interfaz de software utilizada para acceder al dispositivo de red. Desde aquí, se configura el dispositivo, supervisa sus operaciones y ajusta la configuración según sea necesario. El CLI de JunOS tiene dos modos:

- Modo Operacional: En este modo se muestra el estado actual del dispositivo. En el modo operativo, se ingresa comandos para monitoreo y / o para solucionar problemas del sistema operativo Junos, dispositivos y conectividad de red.
- Modo de Configuración: este modo le permite configurar el dispositivo. Una configuración se almacena como una jerarquía de declaraciones de configuración. En este modo, se introduce sentencias para configurar todas las propiedades del dispositivo, incluidas las interfaces, información general de enrutamiento, protocolos de enrutamiento, acceso de usuarios y varias propiedades del sistema y hardware

Cuando ingresa al modo de configuración, en realidad está viendo y cambiando un archivo llamado la configuración candidata. El archivo de configuración candidato le permite hacer cambios de configuración sin causar cambios operacionales a la operación actual configuración, llamada la configuración activa. El enrutador o switch no implementa los cambios que agregó al archivo de configuración candidato hasta que los confirma, lo que activa la configuración en el dispositivo. Configuraciones candidatas le permiten para modificar su configuración sin causar un daño potencial a su red actual operaciones. (Juniper, s.f.)

2.5.1.2.2. IOS (Cisco IOS)

IOS es un paquete de funciones de enrutamiento, conmutamiento, trabajo de internet y telecomunicaciones que se integra estrechamente con un sistema operativo multitarea.

Poseen diferentes tipos de IOS para la necesidad del tipo de cliente, ya sea desde cuentas para empresas pequeñas como para empresas multinacionales.



Figura 20. Logo Cisco IOS

La interfaz de línea de comandos de IOS (IOS CLI) proporciona un conjunto fijo de comandos de múltiples palabras. El conjunto disponible se determina mediante el "modo" y el nivel de privilegios del usuario actual. El modo "Global configuration" proporciona comandos para cambiar la configuración del sistema y el modo "interface configuration" a su vez, proporciona comandos para cambiar la configuración de una interfaz específica. A todos los comandos se les asigna un nivel de privilegios, de 0 a 15, y pueden ser accedidos por usuarios con los privilegios necesarios. A través de la CLI, se pueden definir los comandos disponibles para cada nivel de privilegio. (Cisco, s.f.)

2.5.1.2.3. GNS3

GNS3 es utilizado por cientos de miles de ingenieros de redes en todo el mundo para emular, configurar, probar y solucionar problemas de redes virtuales y reales. GNS3 le permite ejecutar

una pequeña topología que consiste en solo unos pocos dispositivos en su computadora portátil, a aquellos que tienen muchos dispositivos alojados en múltiples servidores o incluso alojados en la nube.

GNS3 es un simulador gráfico de red que te permite diseñar topologías de red complejas y poner en marcha simulaciones sobre ellos. (GNS3, s.f.)



Figura 21. Logo GNS3

Para permitir completar simulaciones, GNS3 está estrechamente vinculada con:

- Dynamips, un emulador de IOS que permite a los usuarios ejecutar binarios/ imágenes IOS de Cisco Systems.
- Dynagen, un front-end basado en texto para Dynamips
- Qemu y VirtualBox, para permitir utilizar máquinas virtuales como un firewall PIX.
- VPCS, un emulador de PC con funciones básicas de networking
- IOU (IOS on Unix), compilaciones especiales de IOS provistas por Cisco para correr directamente en sistemas UNIX y derivados.

2.5.1.2.4. Motores de Simulación

2.5.1.2.4.1. QEMU

QEMU es un emulador de procesadores basado en la traducción dinámica de binarios (conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped). QEMU también tiene capacidades de virtualización dentro de un sistema operativo, ya sea GNU/Linux, Windows, o cualquiera de los sistemas operativos admitidos; de hecho es la forma más común de uso. Esta máquina virtual puede ejecutarse en cualquier tipo de Microprocesador o arquitectura (x86, x86-64, PowerPC, MIPS, SPARC, etc.). Está licenciado en parte con la LGPL y la GPL de GNU.

El objetivo principal es emular un sistema operativo dentro de otro sin tener que reparticionar el disco duro, empleando para su ubicación cualquier directorio dentro de éste. (Anónimo, QEMU, s.f.)



Figura 22. Logo QEMU

2.5.1.2.4.2. VMware player

VMware Player, es un paquete de software de virtualización para computadoras x64 que ejecutan Microsoft Windows o Linux, suministrado de forma gratuita por VMware, Inc. Utiliza el mismo núcleo de virtualización que VMware Workstation.

Muchas máquinas virtuales listas para usar (VM) que se ejecutan en VMware Player, Workstation y otro software de virtualización están disponibles para fines específicos, ya sea para compra o de forma gratuita. Por ejemplo, un "dispositivo de navegador" basado en Linux gratuito con el navegador Firefox instalado está disponible y se puede usar para una navegación segura por la Web; Si está infectado o dañado, puede ser descartado y reemplazado por una copia limpia. Las máquinas virtuales pueden configurarse para restablecerse después de cada uso sin la necesidad de volver a crearlas desde el archivo original. Los proveedores de sistemas operativos con licencias comerciales generalmente requieren que las instalaciones tengan licencia; Las máquinas virtuales con dichos sistemas operativos instalados no pueden distribuirse sin restricciones. Las máquinas virtuales listas para usar con los sistemas operativos Microsoft o Apple instalados, en particular, no se distribuyen, a excepción de las versiones de evaluación. (Anónimo, VMWare, s.f.)



Figura 23. Logo VMware Player

2.5.1.2.5. Herramientas de Análisis

2.5.1.2.5.1. Ostinato

Ostinato es un creador de paquetes, generador de tráfico de red y analizador con una interfaz gráfica de usuario amigable. También una potente API de Python para la automatización de pruebas de red. Cree y envíe paquetes de varios flujos con diferentes protocolos a diferentes velocidades.

Ostinato tiene como objetivo proporcionar un generador de tráfico y una herramienta de prueba de red para cada ingeniero y desarrollador de redes, algo que no es posible hoy en día con el equipo de prueba de red comercial existente. Con la herramienta adecuada, los desarrolladores e ingenieros de redes pueden hacer mejor su trabajo y mejorar la calidad de los productos de redes.

(Srivats, s.f.)



Figura 24. Logo Ostinato

2.5.1.2.5.2. Wireshark

Wireshark, antes conocido como Ethereal, es un analizador de protocolos para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y

protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de los protocolos de la forma humana.

La funcionalidad que proporciona es similar a la de tcpdump, pero agrega una interfaz gráfica y muchas opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de una red (usualmente una red Ethernet, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo. También incluye una versión basada en texto llamada tshark.

Permite examinar los datos de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios de cada paquete. Wireshark incluye un lenguaje completo para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

Wireshark es software libre, y se distribuye en la mayoría de sistemas operativos Unix y compatibles, incluyendo Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Android y Mac OS X, así como en Microsoft Windows. (Anónimo, Wireshark, s.f.)

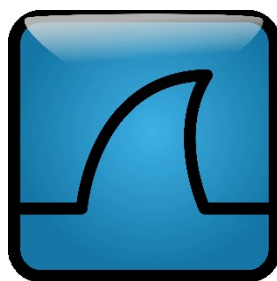


Figura 25. Logo Wireshark

2.5.1.2.5.3. Tiny Core Linux

Tiny Core Linux, o TCL, es un sistema operativo minimalista centrado en proveer un sistema base con el núcleo Linux. La distribución es especialmente notable por su reducido tamaño (de 11 a 16 megabytes) y su minimalismo, y posee un repositorio que contiene más de 3200 extensiones los cuales proveen funciones adicionales a TCL. Tiny Core Linux es un software libre de código abierto, bajo la licencia GNU GPLv2. (Shingledecker, s.f.)



Figura 26. Logo Tiny Core Linux.

CAPÍTULO 3

PROTOTIPO DE IMPLEMENTACIÓN DE VPLS Y EVPN

3.1. Especificación de Requerimientos

La simulación tendrá una configuración mixta en relación a los dispositivos de red, es decir, que el backbone de la topología será del proveedor Juniper, mientras que los dispositivos de acceso serán del proveedor Cisco. Esta configuración únicamente facilitará la recreación de una red real, debido a que la distribución de imágenes de sistemas operativos tanto para Cisco como para Juniper es de carácter licenciado, por tanto las imágenes escogidas y con las características necesarias para la realización de esta investigación serán las siguientes:

- JUNIPER: Junos VMX-14.1R1.10
- CISCO: c2691-js-mz.122-15.T13

Para poder realizar la simulación de las imágenes antes mencionadas; el software de simulación de redes GNS3 es el idóneo para dicho trabajo, puesto que esta plataforma permite la simulación directa e indirectamente de las imágenes en mención. La versión en la que se implementará el prototipo de red y con la que se presenta convergencia con las imágenes es la siguiente:

- GNS3 versión 2.1.4

Como se mencionó anteriormente sobre las imágenes de los sistemas operativos de los dispositivos de red; es necesaria la emulación directa e indirecta de dichos OS, a continuación se detalla:

- Simulación directa: GNS3 incluye un emulador (appliance) de IOS (Cisco) el mismo que permite ejecutar binarios/imágenes sin necesidad de configuraciones adicionales, este se denomina Dynamips.
- Simulación indirecta: GNS3 no posee emuladores para todos los sistemas operativos, por lo que añade la funcionalidad de la utilización de máquinas virtuales, esto permite que por medio de motores de virtualización tales como VMware, VirtualBox, Qemu, etc., se puedan simular en el mismo ambiente, otro tipo de sistemas operativos.

Los motores de virtualización que se utilizarán en esta investigación son los siguientes:

- VMware Player: Este motor de virtualización permitirá la optimización de hardware.
- Qemu: Este motor de virtualización permitirá la generación del emulador (appliance) para la imagen de Junos.

GNS3 presenta la característica para optimizar el rendimiento de hardware de la PC, esto lo hace mediante la utilización de una máquina virtual desarrollada por VMware denominada GNS3 VM (versión 2.1.9, kernel GNU/LINUX 4.4.0-31-generic x86_64), esta máquina virtual tiene como SO Ubuntu 14.04.5 LTS y soporta KVM (Kernel-Virtual Machine). Éste feature permite otorgar recursos de hardware (memoria RAM, números de procesadores, memoria ROM) para tener una mejor performance en la simulación de todos los dispositivos de red que se incluyan en la topología.

La arquitectura que tendrá el ambiente de simulación es el siguiente:

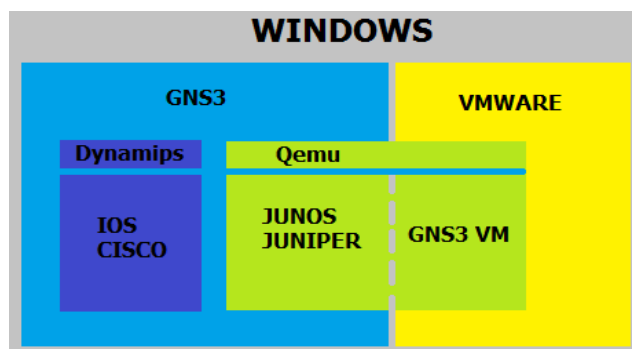


Figura 27. Arquitectura de simulación

3.2. Especificación de escenarios

La topología de una VPN de capa 2 permite conectar sitios ubicados en lugares distintos geográficamente hablando utilizando un puente virtual. Esto consiste en dispositivos de borde de cliente (CE) (host, enrutador, conmutador), conectados a uno o más enrutadores de borde (PE). Los enrutadores PE también pueden incluir un conmutador de borde MPLS (MES) que actúa en el borde de la infraestructura MPLS

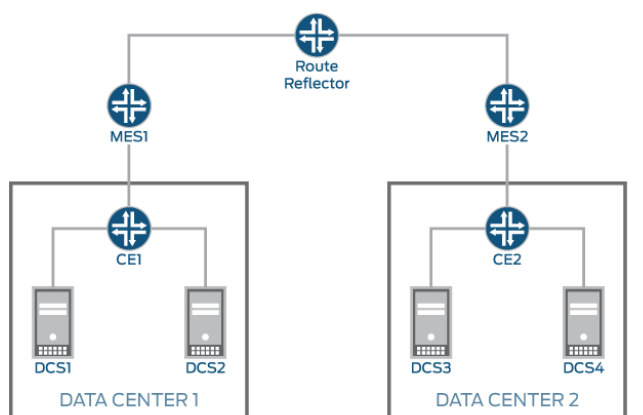


Figura 28. Topología EVPN

Este escenario en concepto de elementos necesarios, serán reflejados a lo largo de esta investigación, es decir, dispositivos CE y PE, donde los dispositivos PE conformarán el Core o backbone MPLS y los CE conformaran los clientes finales.

3.3. Preparación del entorno de simulación

Como se muestra en la figura 27 sobre la arquitectura de simulación, el entorno lo conforma el software de simulación GNS3, los motores de virtualización, herramientas de análisis y las imágenes de los sistemas operativos: todo esto sobre el sistema operativo Windows. A continuación se detalla el desarrollo de las configuraciones y pasos necesarios para la recreación del escenario de pruebas.

3.3.1. GNS3

La versión GNS3-2.1.4-all-in-one es descargable desde la página oficial de GNS3 (<https://www.gns3.com>). Con este archivo se instalan todas las características y appliance por defecto, sin embargo, es necesario la configuración de motores de virtualización para mejora el rendimiento de hardware y el acople del nuevo appliance para la imagen de Junos.

3.3.1.1. Instalación

Lanzar el archivo de instalación descargado desde la página oficial y seguir los pasos del asistente de instalación:



Figura 29. Instalación GNS3 [Paso 1]

GNS3 es un software gratuito de código abierto distribuido bajo la Licencia Pública General de GNU Versión 3, clic en el botón Acepto para continuar con la instalación:

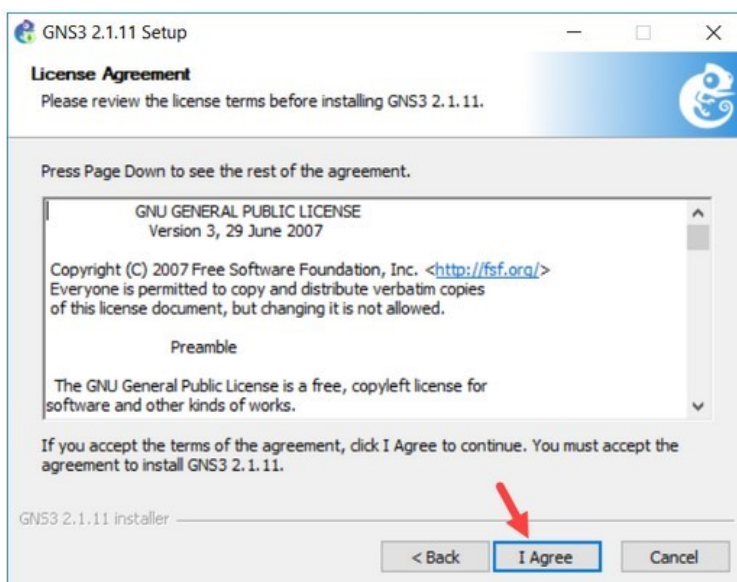


Figura 30. Instalación GNS3 [Paso 2]

Seleccione la carpeta Menú Inicio para el acceso directo GNS3. El valor predeterminado es la carpeta GNS3. Haga clic en **Siguiente>** para continuar con la instalación:

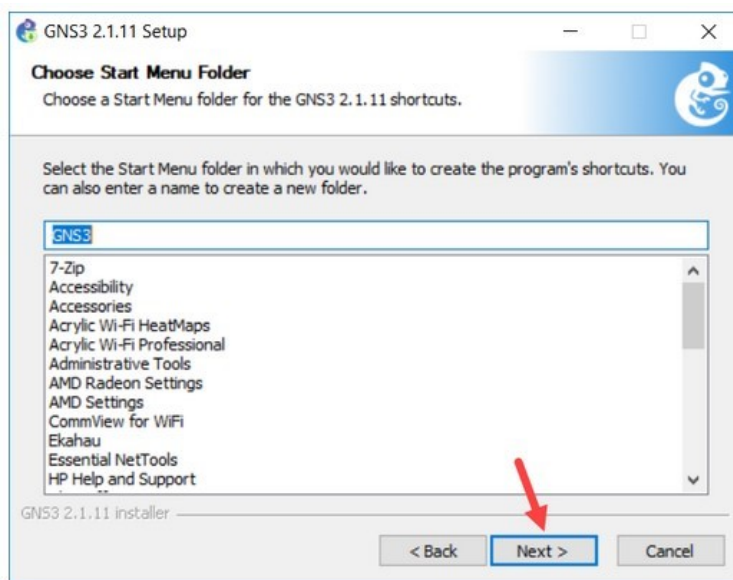


Figura 31. Instalación GNS3 [Paso 3]

GNS3 viene con varios prerrequisitos y software opcional. Por defecto, la mayoría del software está seleccionado para la instalación, pero puede decidir instalar solo un software específico.

Dejar todas las selecciones de software en su selección predeterminada y haga clic en **Siguiente>** para continuar con la instalación:

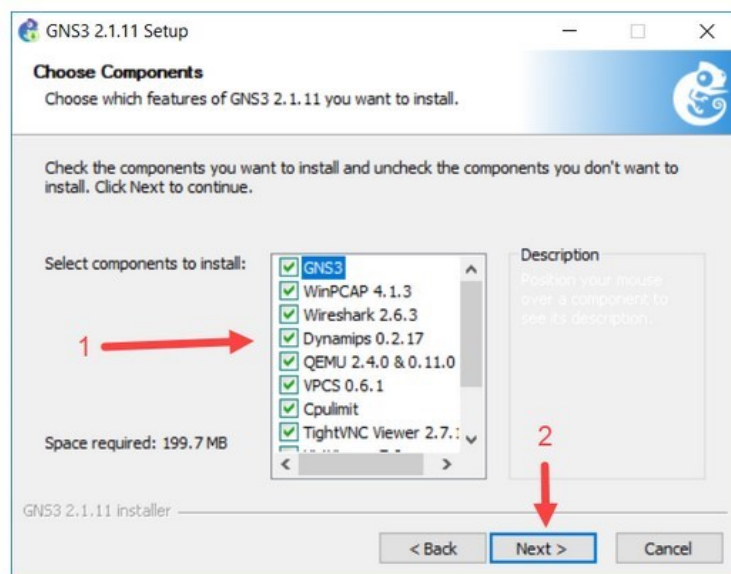


Figura 32. Instalación GNS3 [Paso 4]

Elegir una ubicación de instalación. La ubicación predeterminada es C: \ Archivos de programa \ GNS3. Luego hacer clic en Instalar:

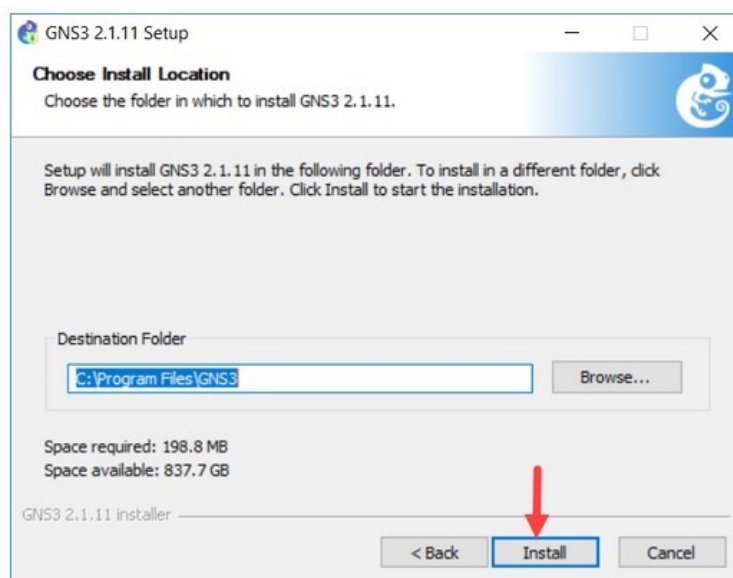


Figura 33. Instalación GNS3 [Paso 5]

A continuación se irán instalando los complementos como se muestra en la figura 32. Estos complementos serán de utilidad para instanciación de appliance y análisis de datos de la topología de red a utilizar en esta investigación.

3.3.1.2. Configuración Básica

GNS3 tiene la característica de utilizar servidores locales o remotos para mejorar el rendimiento de hardware, por lo que es necesario crear adaptadores de red del tipo loopback, de esta manera en las preferencias de servidores en GNS3 se direcciona hacia la IP del adaptador de loopback; para lo antes mencionado se debe seguir los siguientes pasos:

3.3.1.2.1. Creación de interfaz loopback

Para la creación de las interfaces de loopback, es necesario ir al administrador de dispositivos en propiedades del sistema, luego agregar hardware heredado como se muestra en la figura 34.

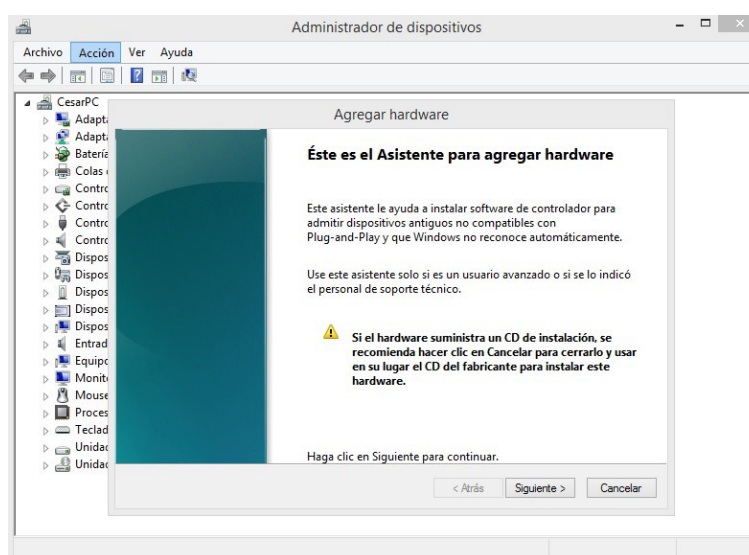


Figura 34. Interfaz Loopback [Paso 1]

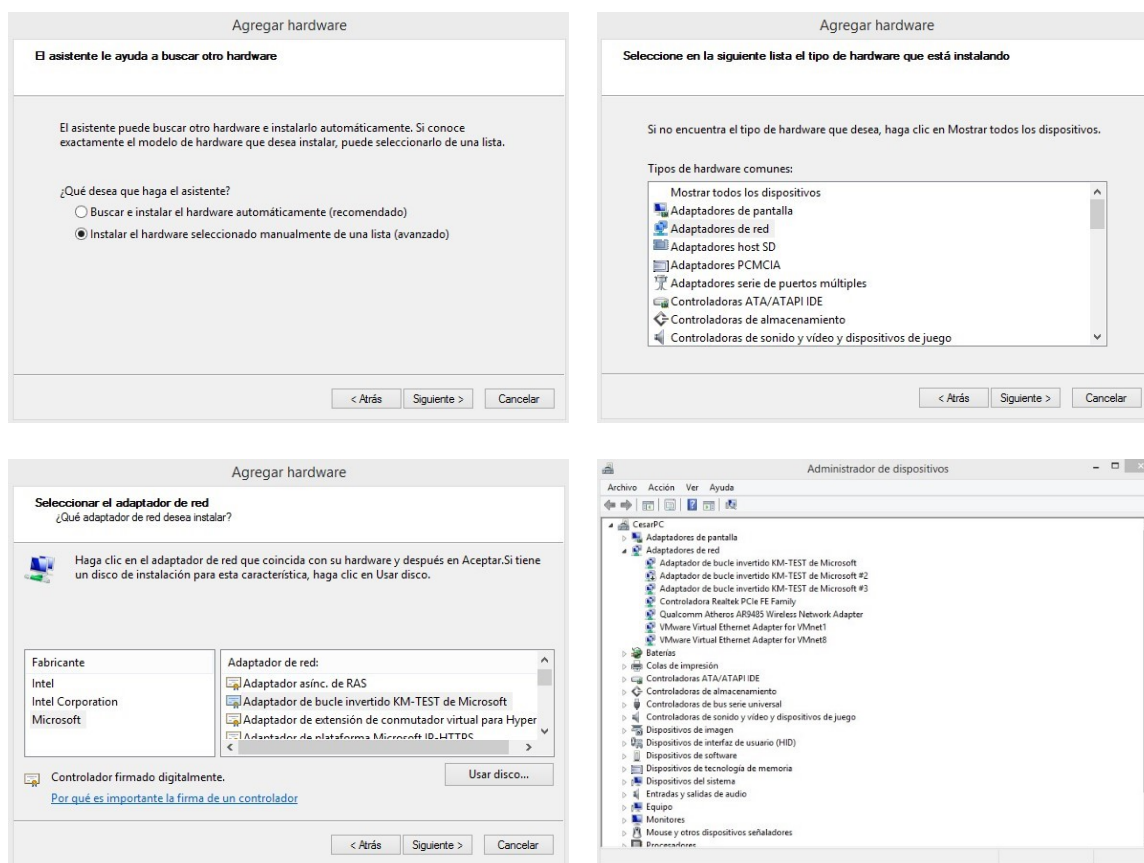


Figura 35. Interfaz Loopback [Pasos 2-4]

3.3.1.2.2. Configuración de servidores

GNS3 puede asociar servidores para mejorar el rendimiento de hardware, eso lo hace por medio de una subred dentro del sistema operativo, es decir, se coloca una dirección IP a la interfaz loopback y esta se configura en las preferencias de GNS3.

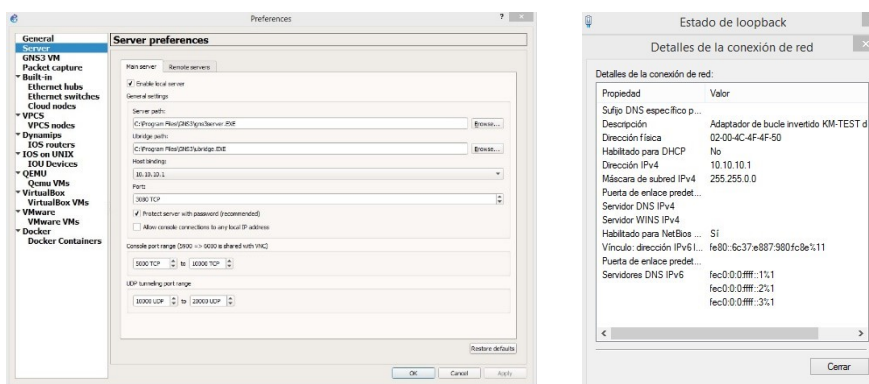


Figura 36. Preferencias de Servidores GNS3.

La IP asignada es 10.10.10.1/24, esta sub red escogida es únicamente para que el motor de virtualización VMWare tenga alcanzabilidad, se explica en 3.3.2.

3.3.2. Motores de virtualización

Los motores de virtualización que se utilizará para esta investigación son:

- QEMU
- VMWARE

Debido a que GNS3 no posee un appliance para la versión VMX de Junos, es necesario crear un nuevo appliance, los pasos para lo antes mencionado son los siguientes:

3.3.2.1. VMware

Este motor de virtualización nos permitirá arrancar un complemento para mejorar el rendimiento de hardware.

Desde las preferencias de GNS3 en la sección GNS3VM, contiene un link, el mismo que nos servirá para descargar este motor de virtualización listo para ser instalado en la arquitectura de simulación (ver figura 27):

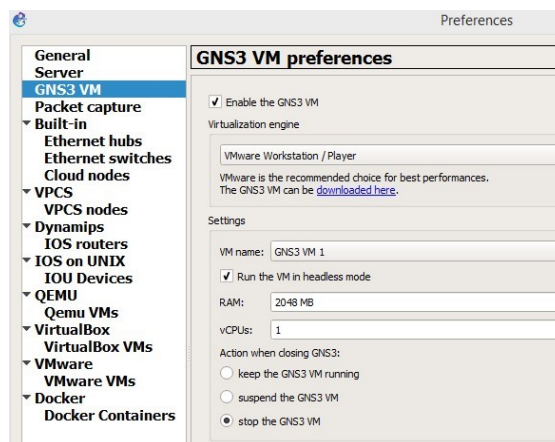


Figura 37. Link de descarga GNS3 VM

El archivo descargado es el siguiente:

- GNS3 VM.ova

Este archivo lo importamos en VMware y realizaremos lo siguiente

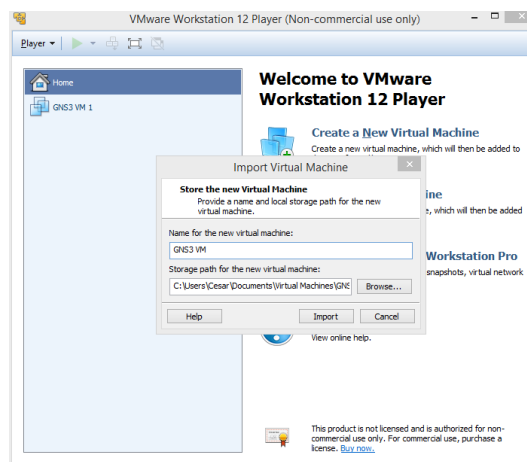
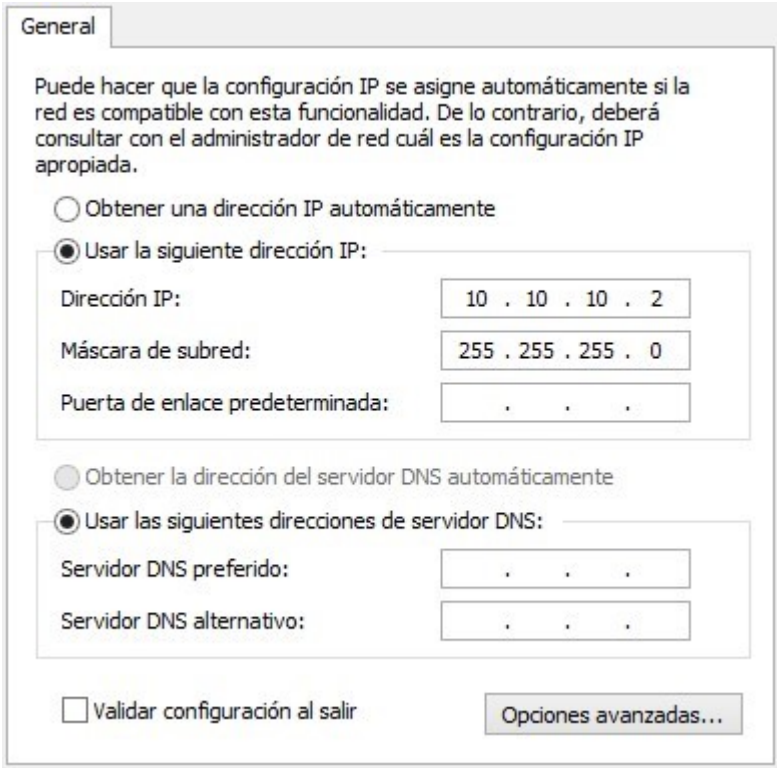


Figura 38. Importación de GNS3 VM en VMware

En la primera ejecución de la máquina virtual va a ser necesario la configuración de red para lograr una alcanzabilidad con los complementos que presenta GNS3, para esto en primera instancia se debe configurar el adaptador de red VMware Virtual Ethernet Adapter for VMnet1, como muestra la figura 39.



General

Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.

Obtener una dirección IP automáticamente

Usar la siguiente dirección IP:

Dirección IP: 10 . 10 . 10 . 2

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: . . .

Obtener la dirección del servidor DNS automáticamente

Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: . . .

Servidor DNS alternativo: . . .

Validar configuración al salir

Opciones avanzadas...

Figura 39. Configuración de red del adaptador VMnet1

La IP asignada es la siguiente:

- 10.10.10.2/24

De esta manera el motor de virtualización VMware tendrá alcanzabilidad con GNS3, sin embargo el motor de virtualización GNS3 VM también debe estar en la misma sub red para que

pueda ser utilizado por GNS3, por lo antes mencionado, la configuración de red de GNS3 VM es la siguiente:

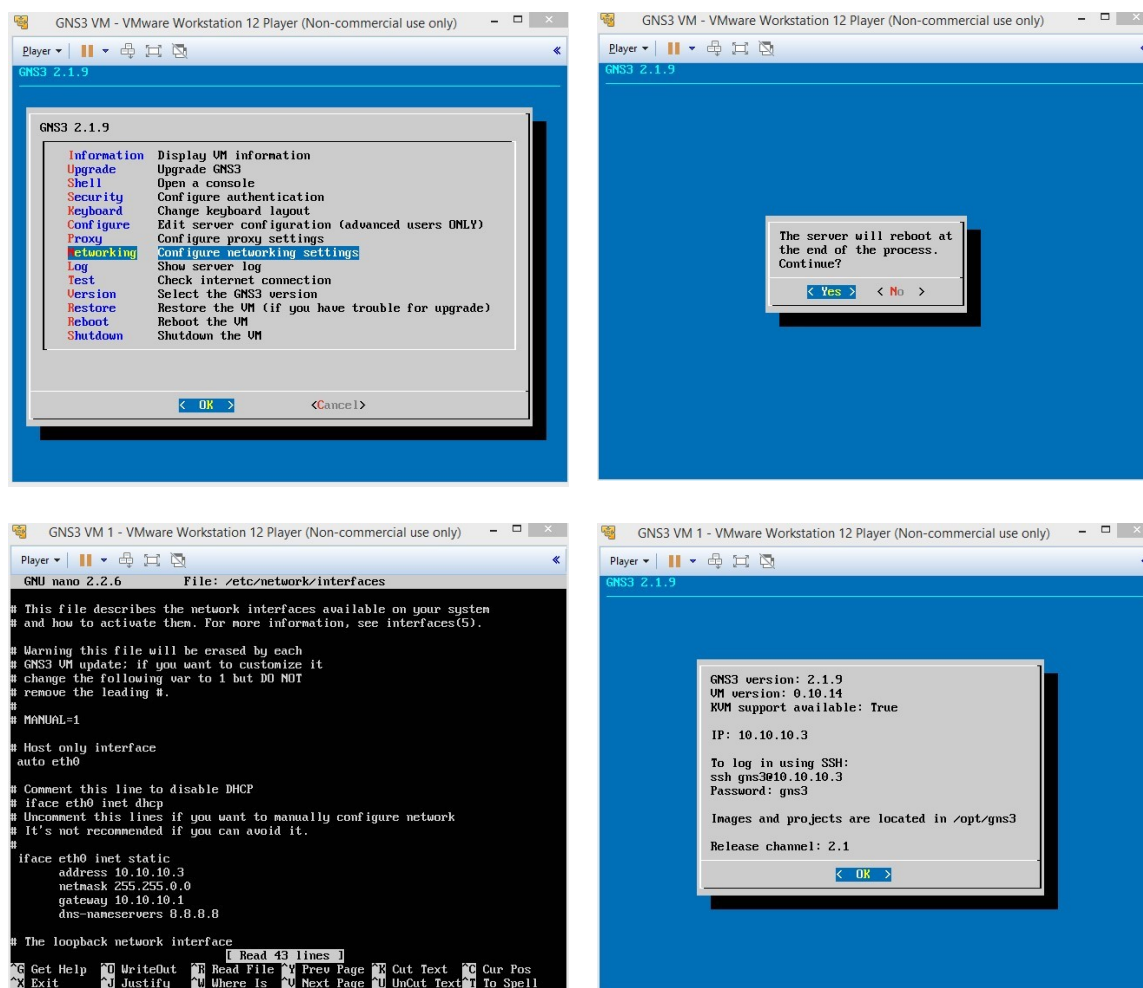


Figura 40. Pasos para la configuración de red de GNS3 VM [Pasos 1-4]

La IP configurada es la siguiente:

- 10.10.10.3/24

De esta manera la arquitectura de simulación queda en razón de red de la siguiente manera:

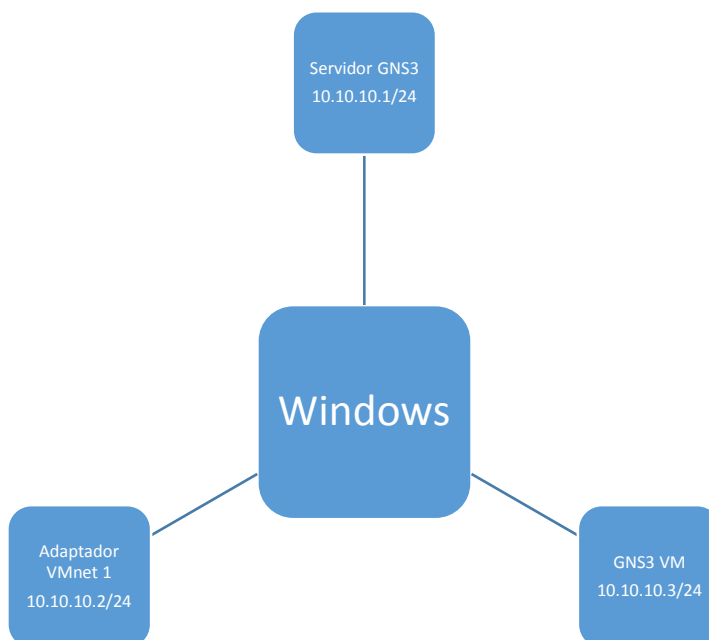


Figura 41. Configuración de red de la arquitectura de simulación

3.3.2.2. QEMU

El motor de virtualización QEMU se instala junto con el paquete de instalación de GNS3, por tanto la configuración a explicar es la siguiente:

GNS3 por defecto tiene instalado dos appliance de Juniper:

- Juniper vMX vCP, y
- Juniper vMX vFP

Estas imágenes de JunOS no son posibles descargarlas desde la página oficial de Juniper, ya que para hacerlo es necesario tener adquirida una licencia por pago. Por lo que es indispensable

virtualizar una imagen de JunOS que sea factible su descarga de manera gratuita; para esto se eligió la versión VMX-14.1R1.10, la misma que contiene los módulos necesarios para esta investigación.

Para lograr esta virtualización descargamos la imagen de JunOS desde esta página Web <https://lisrohete.tk/sro/vmx-jinstall-vmx-14-1r1-10-domestic.html>.

El archivo adquirido es el siguiente:

- jinstall-vmx-14.1R1.10-domestic.img

Para realizar la virtualización es necesario que GNS3 VM ya se encuentre habilitado y en línea, como se muestra en la figura 42.

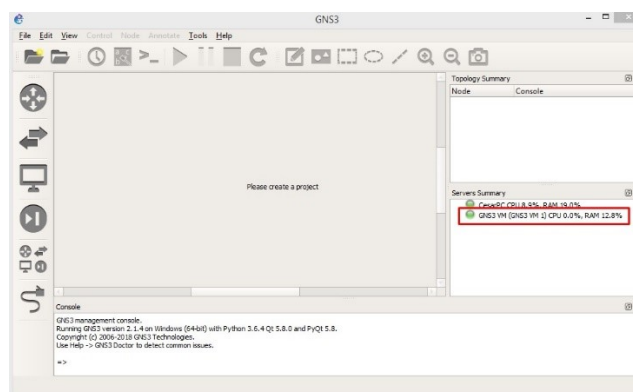


Figura 42. GNS3 VM en línea

En preferencias, sección Qemu VM, se agrega un nuevo template con el servidor GNS3 VM como primer paso.

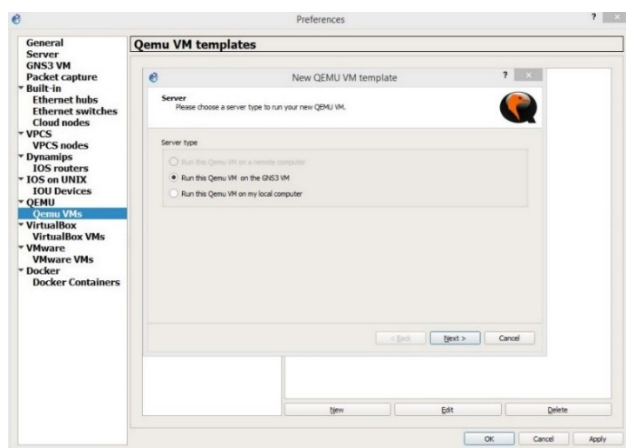


Figura 43. QEMU VM template

Seguido de los siguiente, donde se especifica el nombre, requerimientos de hardware, consola de gestión y la asociación del disco de imagen.

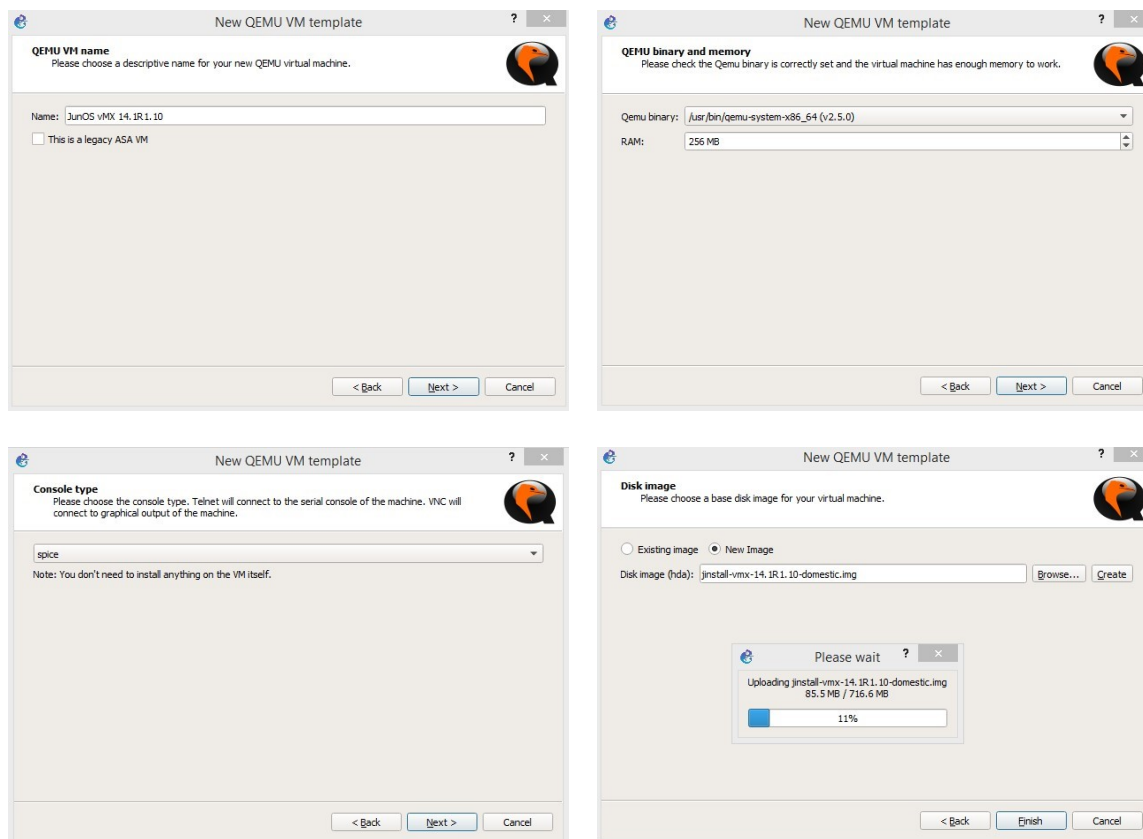


Figura 44. Creación del nuevo template Qemu VM [Pasos 2-4]

De esta manera ya podemos contar con la imagen lista para ser utilizada en el ambiente GNS3 como cualquier otro dispositivo con appliance.

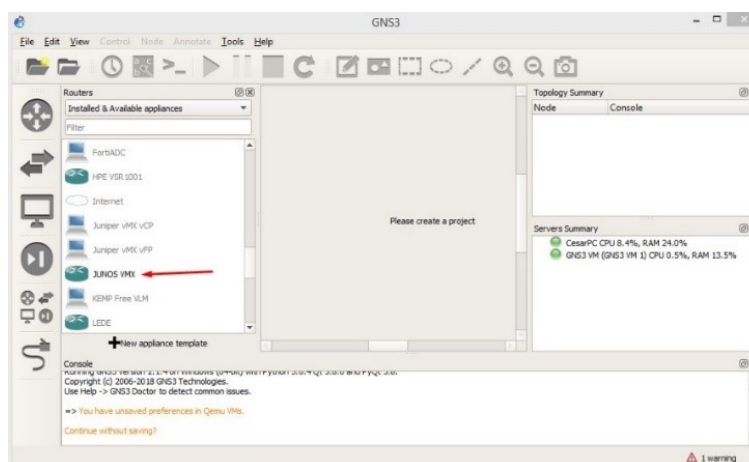


Figura 45. Appliance JunOS vMX 14.1R1.10

3.3.3. Herramientas de análisis

Las herramientas de análisis para la verificación de tramas y rendimiento de la red son:

- Wireshark
- Ostinato

Wireshark es instalado junto con el paquete de GNS3 por lo que no va a ser necesario la explicación de instalación del mismo. GNS3 no contiene el appliance de Ostinato por lo que es necesario realizar el mismo procedimiento que con la imagen JunOS descrito en la sección 3.2.2.2. La imagen del software es descargada desde la página oficial de Ostinato <https://ostinato.org/> y la versión es la 0.9. Como resultado del mismo procedimiento tenemos lo mostrado en la siguiente figura:

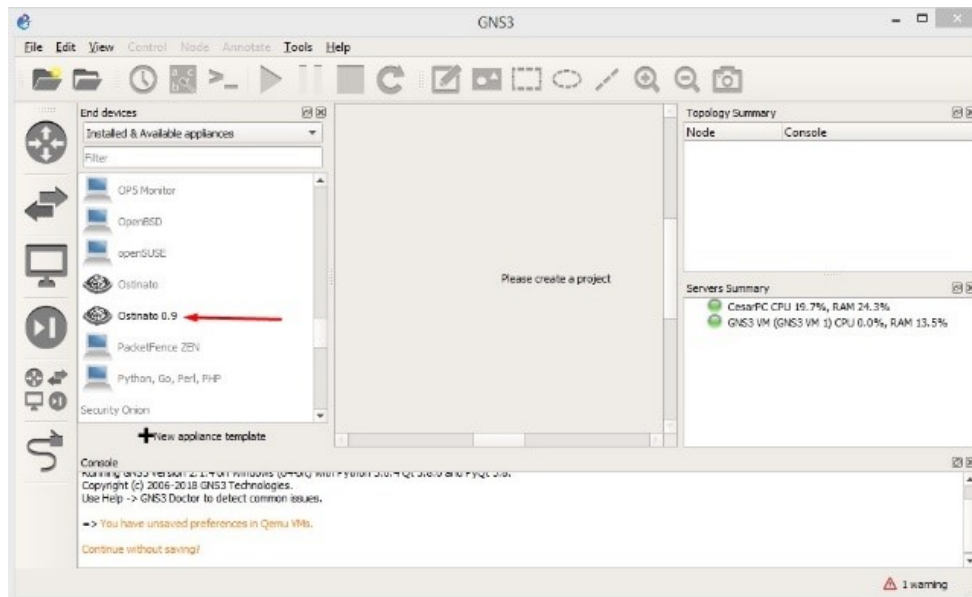


Figura 46. Appliance Ostinato 0.9

CAPÍTULO 4

IMPLEMENTACIÓN Y ANÁLISIS DE RENDIMIENTO

4.1. Topología de Red

El escenario de pruebas para la evaluación de rendimiento de las tecnologías EVPN/VPLS, consta de 6 enrutadores que soportan VPN's de capa 2 y permiten la convergencia de MPLS, BGP y los protocolos que hacen posible el encaminamiento de la información de alcanzabilidad como LDP, OSPF, RSVP.

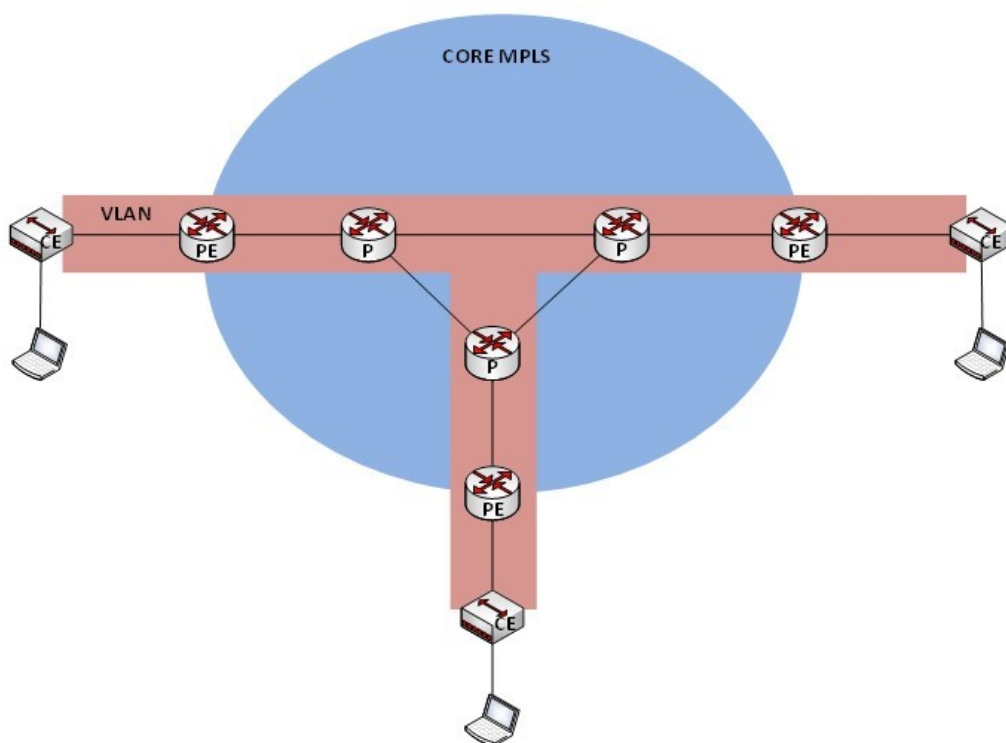


Figura 47. Topología de Red

Para este fin, la topología como se muestra en la figura 47, tiene tres enrutadores que forman un anillo y a su vez conforman parte del Core de la red MPLS denominados Provider Routers (P), los otros tres enrutadores a más de formar parte del Core, son los denominados routers de borde (PE – Provider Edge), los mismos que cargan con toda la ingeniería que hace posible la alcanzabilidad de extremo a extremo.

Los enrutadores de borde únicamente intercambian información de alcanzabilidad por medio de BGP, por lo que es necesario tener dispositivos de acceso para la conectividad con el cliente final, estos dispositivos son los denominados Customer Edge (CE), estos dispositivos no necesitan tener la ingeniería de VPN's y únicamente trabajan en función de VLAN's. En la siguiente tabla se muestra el Router y el papel que cumple en la topología:

Tabla 1.
Papel de Routers

Router	Función	SO
CE	Acceso	IOS
PE	Router de Borde	JunOS
P	Core	JunOS

4.1.1. Direccionamiento Físico

La imagen de JunOS cargada en la sección 3.3.2.2 presenta una inconsistencia en cuanto a la nomenclatura de la asignación de interfaces de red, para esto y con la ayuda de la simulación de varios escenarios, se determina que el template Qemu creado debe tener un número de cinco interfaces; el orden y nomenclatura de las interfaces de red es el siguiente:

Tabla 2.
Asignación de interfaces

Interfaz Física	Interfaz Lógica
e0/0	N/A
e1/0	N/A
e2/0	ge0/0/0
e3/0	ge0/0/1
e4/0	ge0/0/2

Dada la tabla 1, el direccionamiento físico para el escenario en el que se evalúan las dos tecnologías de VPN's, es la siguiente:

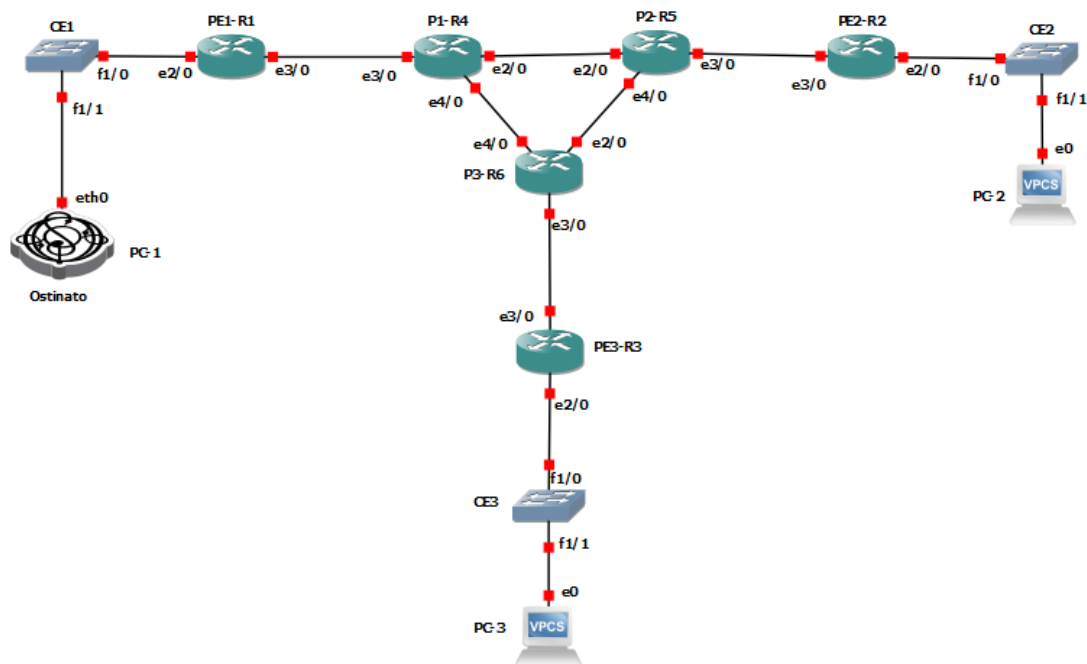


Figura 48. Direccionamiento Físico.

4.1.2. Direccionamiento Lógico

Del mismo modo que para el direccionamiento físico, el direccionamiento lógico va a ser el mismo para las dos tecnologías de VPN en el Core de la red, salvo para el nivel acceso el direccionamiento lógico cambia, lo antes mencionado se detalla en las siguientes secciones.

4.1.3. Direccionamiento lógico VPLS

El direccionamiento lógico para el entorno VPLS es el siguiente:

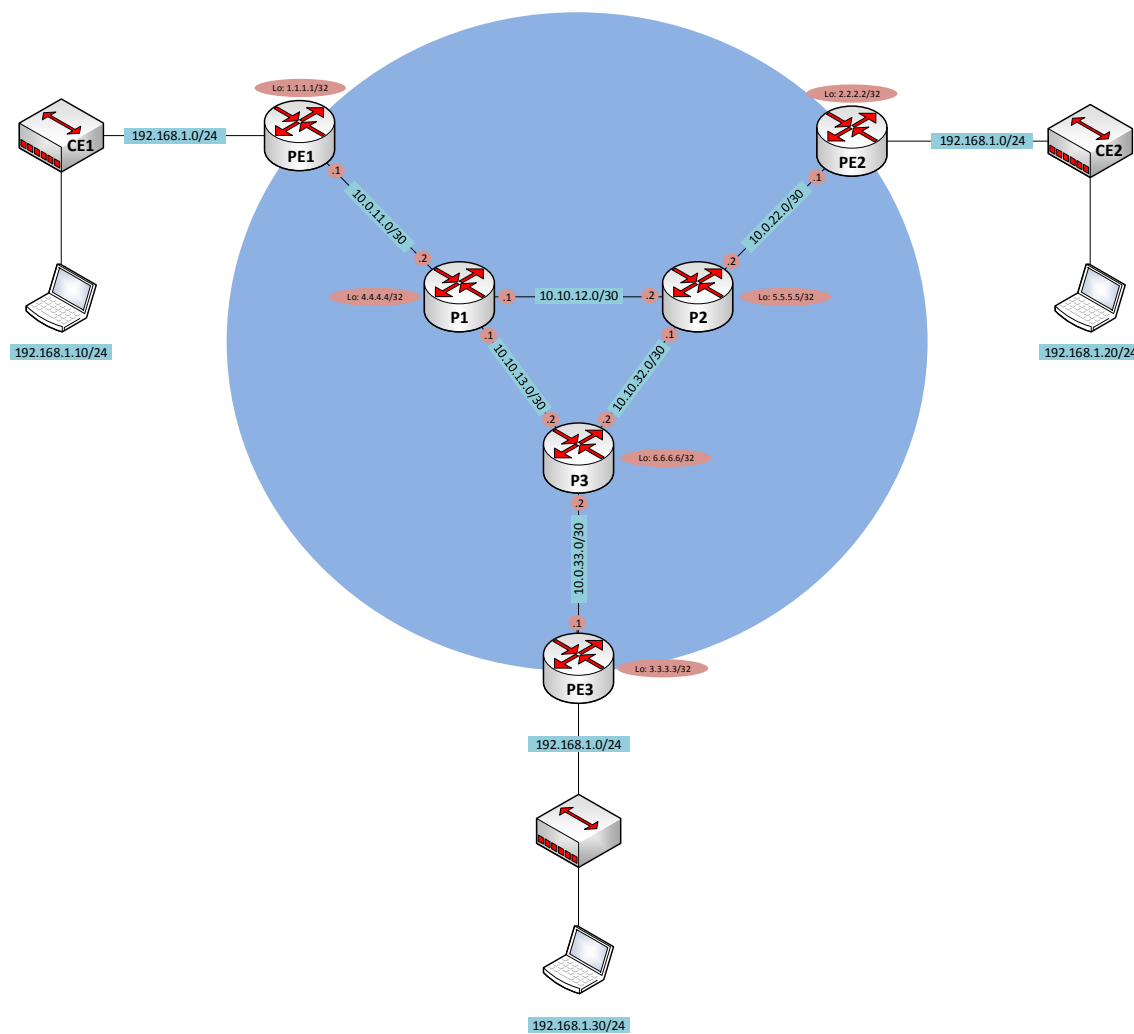


Figura 49. Direccionamiento Lógico VPLS

4.1.3.1. Direccionamiento lógico EVPN

El direccionamiento lógico para el entorno EVPN es el siguiente:

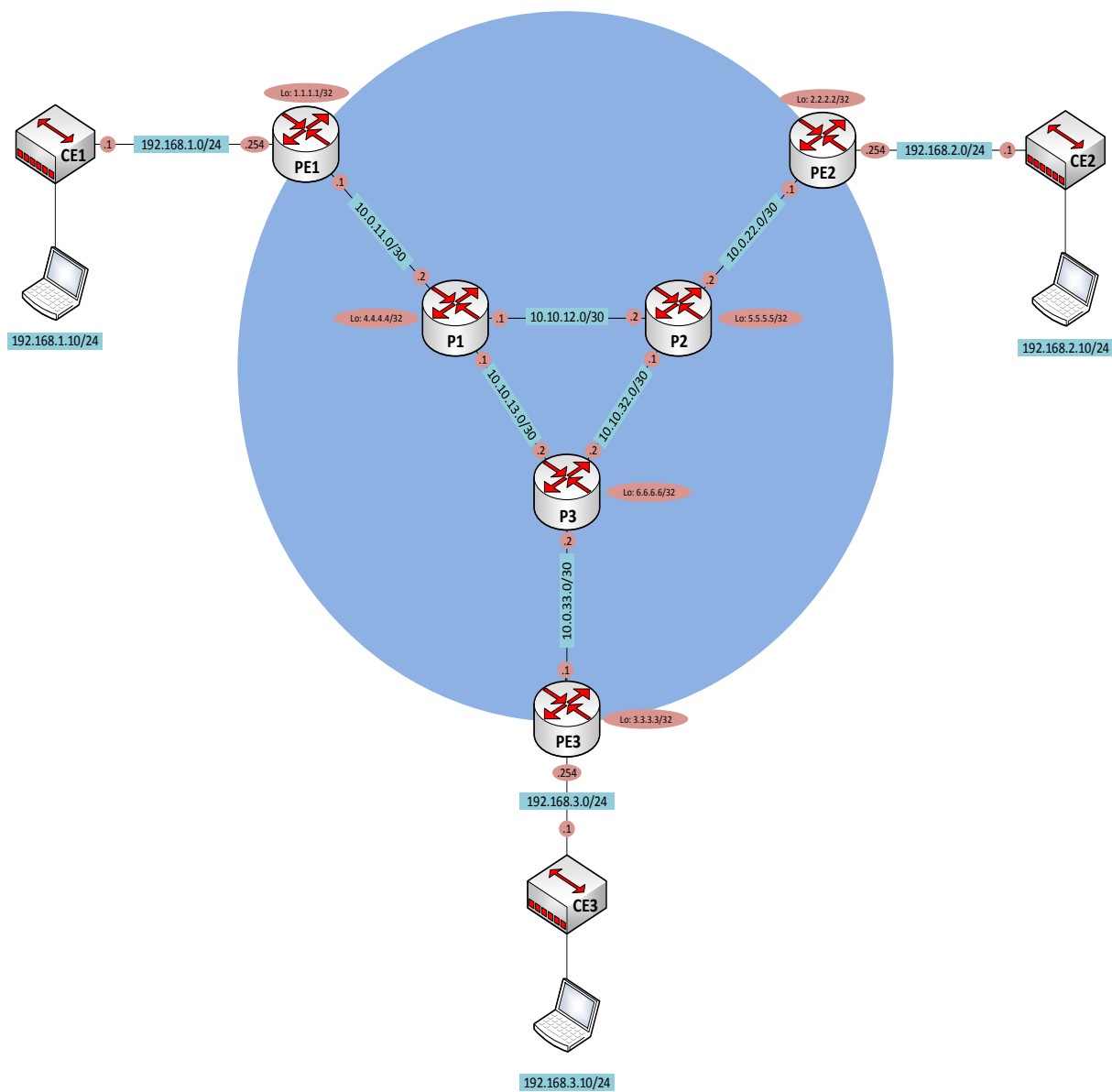


Figura 50. Direccionamiento Lógico EVPN

4.2. Lineamientos generales

Los lineamientos generales especifican los protocolos tanto para capa 2 como para capa 3 para lograr conectividad entre todos los routers, así como para el intercambio de información de alcanzabilidad.

Los protocolos para lograr lo antes mencionado son los siguientes:

- El protocolo IGP para lograr alcanzabilidad es OSPF, en el cual se define una sola área para todos los routers, ÁREA 0.
- MPLS, como medio de transporte entre las entidades de extremo.
- LDP, es el protocolo que define el Core de la red MPLS.
- BGP, protocolo para alcanzabilidad de extremo a extremo con sistema autónomo número AS 100.
- VLAN, las dos tecnologías de VPN utilizan encapsulación VLAN para lograr la alcanzabilidad de extremo a extremo con dispositivos de red capa 2 y 3, el número de la vlan asignada es 10.

A más de los protocolos que permiten la conectividad en el Core, es necesario la creación de instancias de enrutamiento que nos permitan creación de conexiones virtuales de capa 2, éstas son VPLS y EVPN. Como se muestra en la figura 48, a cada Router se le asigna un número el mismo que sirve para la asignación de direcciones IP a interfaces de loopback, las mismas que son de utilidad para la convergencia de MPLS, BGP, OSPF, VPLS y EVPN; en la siguiente tabla se indica las direcciones IP de loopback asignadas a cada Router.

Tabla 3.
Interfaces loopback

Router	Dirección IP
PE1	1.1.1.1/32
PE2	2.2.2.2/32
PE3	3.3.3.3/32
P1	4.4.4.4/32
P2	5.5.5.5/32
P3	6.6.6.6/32

Dados los lineamientos en esta sección y el direccionamiento físico mostrado en la figura 48. Los protocolos que se emplearan en las interfaces de los routers, se muestran en la siguiente tabla.

Tabla 4.
Lineamientos generales

PROTOCOLO						
ROUTER	INTERFACE	MPLS	LDP	BGP	OSPF	VPLS/EVPN
PE1-R1	GE-0/0/0	-	-	-	-	-
	GE-0/0/1	X	X	X	X	X
PE2-R2	GE-0/0/0	-	-	-	-	-
	GE-0/0/1	X	X	X	X	X
PE3-R3	GE-0/0/0	-	-	-	-	-
	GE-0/0/1	X	X	X	X	X

CONTINÚA

P1-R4	GE-0/0/0	X	X	X	X	-
	GE-0/0/1	X	X	X	X	-
	GE-0/0/2	X	X	X	X	-
P2-R5	GE-0/0/0	X	X	X	X	-
	GE-0/0/1	X	X	X	X	-
	GE-0/0/2	X	X	X	X	-
P3-R6	GE-0/0/0	X	X	X	X	-
	GE-0/0/1	X	X	X	X	-
	GE-0/0/2	X	X	X	X	-

4.3. Información de alcanzabilidad

Debido a que los escenarios de distribución física y lógica tanto para VPLS y EVPN son los mismos en el Core de la red, la información de alcanzabilidad será la misma en ambos casos, esto se evidencia en la tabla de enrutamiento OSPF; como referencia se ha tomado al dispositivo PE1-R1 para la comparación entre las dos tecnologías.

VPLS	EVPN
inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden) + = Active Route, - = Last Active, * = Both	inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden) + = Active Route, - = Last Active, * = Both
1.1.1.1/32 * [Direct/0] 00:14:30 > via lo0.0	1.1.1.1/32 * [Direct/0] 00:04:36 > via lo0.0
2.2.2.2/32 * [OSPF/10] 00:04:32, metric 3 > to 10.0.11.2 via ge-0/0/1.0	2.2.2.2/32 * [OSPF/10] 00:00:29, metric 3 > to 10.0.11.2 via ge-0/0/1.0
3.3.3.3/32 * [OSPF/10] 00:04:32, metric 3 > to 10.0.11.2 via ge-0/0/1.0	3.3.3.3/32 * [OSPF/10] 00:00:08, metric 3 > to 10.0.11.2 via ge-0/0/1.0
4.4.4.4/32 * [OSPF/10] 00:06:39, metric 1 > to 10.0.11.2 via ge-0/0/1.0	4.4.4.4/32 * [OSPF/10] 00:00:34, metric 1 > to 10.0.11.2 via ge-0/0/1.0
5.5.5.5/32 * [OSPF/10] 00:04:40, metric 2 > to 10.0.11.2 via ge-0/0/1.0	5.5.5.5/32 * [OSPF/10] 00:00:34, metric 2 > to 10.0.11.2 via ge-0/0/1.0
6.6.6.6/32 * [OSPF/10] 00:04:40, metric 2 > to 10.0.11.2 via ge-0/0/1.0	6.6.6.6/32 * [OSPF/10] 00:00:08, metric 2 > to 10.0.11.2 via ge-0/0/1.0
10.0.11.0/30 * [Direct/0] 00:14:30 > via ge-0/0/1.0	10.0.11.0/30 * [Direct/0] 00:01:24 > via ge-0/0/1.0
10.0.11.1/32 * [Local/0] 00:14:30 Local via ge-0/0/1.0	10.0.11.1/32 * [Local/0] 00:01:41 Local via ge-0/0/1.0
10.0.22.0/30 * [OSPF/10] 00:04:40, metric 3 > to 10.0.11.2 via ge-0/0/1.0	10.0.22.0/30 * [OSPF/10] 00:00:34, metric 3 > to 10.0.11.2 via ge-0/0/1.0
10.0.33.0/30 * [OSPF/10] 00:04:40, metric 3 > to 10.0.11.2 via ge-0/0/1.0	10.0.33.0/30 * [OSPF/10] 00:00:08, metric 3 > to 10.0.11.2 via ge-0/0/1.0
10.10.12.0/30 * [OSPF/10] 00:06:39, metric 2 > to 10.0.11.2 via ge-0/0/1.0	10.10.12.0/30 * [OSPF/10] 00:00:34, metric 2 > to 10.0.11.2 via ge-0/0/1.0
10.10.13.0/30 * [OSPF/10] 00:06:39, metric 2 > to 10.0.11.2 via ge-0/0/1.0	10.10.13.0/30 * [OSPF/10] 00:00:08, metric 2 > to 10.0.11.2 via ge-0/0/1.0
10.10.32.0/30 * [OSPF/10] 00:04:40, metric 3 > to 10.0.11.2 via ge-0/0/1.0	10.10.32.0/30 * [OSPF/10] 00:00:29, metric 3 > to 10.0.11.2 via ge-0/0/1.0
224.0.0.5/32 * [OSPF/10] 00:14:31, metric 1 MultiRecv	224.0.0.5/32 * [OSPF/10] 00:04:59, metric 1 MultiRecv

Figura 51. Tabla inet.0 PE1-R1 (VPLS vs EVPN).

4.4. Información L2VPN

En esta sección se evidencia la diferencia que existe en las tablas de enrutamiento a nivel de capa 2 y capa 3 entre las dos tecnologías de VPN.

4.4.1. VPLS

En los enrutadores de borde se generan tablas a nivel de capa 2 gracias a la señalización BGP y las instancias de ruteo creadas con el protocolo VPLS. En las figuras siguientes se muestran las tablas de capa 2 en los enrutadores de borde.

```
vpls.l2vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1:10:1:1/96
*[L2VPN/170/-101] 00:24:30, metric2 1
  Indirect

2.2.2.2:10:2:1/96
*[BGP/170] 00:12:46, localpref 100, from 2.2.2.2
  AS path: I, validation-state: unverified
  > to 10.0.11.2 via ge-0/0/1.0, Push 299824

3.3.3.3:10:3:1/96
*[BGP/170] 00:12:42, localpref 100, from 3.3.3.3
  AS path: I, validation-state: unverified
  > to 10.0.11.2 via ge-0/0/1.0, Push 299840
```

Figura 52. Tabla vpls.l2vpn.0 PE1-R1

```
vpls.l2vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1:10:1:1/96
*[BGP/170] 00:22:20, localpref 100, from 1.1.1.1
  AS path: I, validation-state: unverified
  > to 10.0.22.2 via ge-0/0/1.0, Push 299792

2.2.2.2:10:2:1/96
*[L2VPN/170/-101] 00:31:33, metric2 1
  Indirect

3.3.3.3:10:3:1/96
*[BGP/170] 00:22:13, localpref 100, from 3.3.3.3
  AS path: I, validation-state: unverified
  > to 10.0.22.2 via ge-0/0/1.0, Push 299840
```

Figura 53. Tabla vpls.l2vpn.0 PE2-R2

```
vpls.l2vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1:10:1:1/96
*[BGP/170] 00:19:39, localpref 100, from 1.1.1.1
  AS path: I, validation-state: unverified
  > to 10.0.33.2 via ge-0/0/1.0, Push 299792

2.2.2.2:10:2:1/96
*[BGP/170] 00:19:35, localpref 100, from 2.2.2.2
  AS path: I, validation-state: unverified
  > to 10.0.33.2 via ge-0/0/1.0, Push 299840

3.3.3.3:10:3:1/96
*[L2VPN/170/-101] 00:25:51, metric2 1
  Indirect
```

Figura 54. Tabla vpls.l2vpn.0 PE3-R3

4.4.2. EVPN

A diferencia de VPLS, EVPN tiene la característica de que puede crear VRF's, esto con el fin de que varias instancias de una tabla de enrutamiento puedan existir en un Router y trabajar simultáneamente. En las figuras siguientes se muestran las tablas de VRF en los enrutadores de borde.

```
vrf.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.0/24      * [Direct/0] 00:09:16
> via irb.0
192.168.1.254/32   * [Local/0] 00:12:15
Local via irb.0
192.168.2.0/24     * [BGP/170] 00:08:00, localpref 100, from 2.2.2.2
AS path: I, validation-state: unverified
> to 10.0.11.2 via ge-0/0/1.0, Push 16, Push 299808(top)
192.168.2.254/32  * [EVPN/7] 00:08:00, metric2 1
> to 10.0.11.2 via ge-0/0/1.0, Push 299776, Push 299808(top)
192.168.3.0/24    * [BGP/170] 00:07:56, localpref 100, from 3.3.3.3
AS path: I, validation-state: unverified
> to 10.0.11.2 via ge-0/0/1.0, Push 16, Push 299824(top)
192.168.3.254/32 * [EVPN/7] 00:07:56, metric2 1
> to 10.0.11.2 via ge-0/0/1.0, Push 299776, Push 299824(top)
```

Figura 55. Tabla vrf.inet

```
vrf.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.0/24     * [BGP/170] 00:14:12, localpref 100, from 1.1.1.1
AS path: I, validation-state: unverified
> to 10.0.22.2 via ge-0/0/1.0, Push 16, Push 299792(top)
192.168.1.254/32  * [EVPN/7] 00:14:12, metric2 1
> to 10.0.22.2 via ge-0/0/1.0, Push 299776, Push 299792(top)
192.168.2.0/24    * [Direct/0] 00:15:17
> via irb.0
192.168.2.254/32  * [Local/0] 00:17:23
Local via irb.0
192.168.3.0/24    * [BGP/170] 00:14:03, localpref 100, from 3.3.3.3
AS path: I, validation-state: unverified
> to 10.0.22.2 via ge-0/0/1.0, Push 16, Push 299840(top)
192.168.3.254/32 * [EVPN/7] 00:14:03, metric2 1
> to 10.0.22.2 via ge-0/0/1.0, Push 299776, Push 299840(top)
```

Figura 56. Tabla vrf.inet.0 PE2-R2.

```
vrf.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.0/24     * [BGP/170] 00:18:10, localpref 100, from 1.1.1.1
AS path: I, validation-state: unverified
> to 10.0.33.2 via ge-0/0/1.0, Push 16, Push 299808(top)
192.168.1.254/32  * [EVPN/7] 00:18:10, metric2 1
> to 10.0.33.2 via ge-0/0/1.0, Push 299776, Push 299808(top)
192.168.2.0/24    * [BGP/170] 00:18:04, localpref 100, from 2.2.2.2
AS path: I, validation-state: unverified
> to 10.0.33.2 via ge-0/0/1.0, Push 16, Push 299824(top)
192.168.2.254/32 * [EVPN/7] 00:18:04, metric2 1
> to 10.0.33.2 via ge-0/0/1.0, Push 299776, Push 299824(top)
192.168.3.0/24    * [Direct/0] 00:19:18
> via irb.0
192.168.3.254/32 * [Local/0] 00:21:11
Local via irb.0
```

Figura 57. Tabla vrf.inet.0 PE3-R3.

4.5. Análisis de resultados

La topología de red está diseñada para ejecutar tres escenarios de pruebas, los mismos que nos determinarán qué tecnología tiene mejores prestaciones, por medio de los siguientes parámetros:

- i. Convergencia de la red total
- ii. Aprendizaje MAC
- iii. Supresión de tráfico BUM

4.5.1. Convergencia de la red total

Después de configurar la red, tener conectividad de extremo a extremo PE1-PE2-PE3, de acuerdo al direccionamiento físico indicado en la figura 48 para VPLS/EVPN. Se configura el cliente final de acuerdo a la siguiente tabla para verificar la conectividad.

Tabla 5.
Configuración de red de los Clientes Finales.

	VPLS	EVPN
PC-1	192.168.1.10/24	192.168.1.10/24 - GW192.168.1.254
PC-2	192.168.1.20/24	192.168.2.10/24 - GW192.168.2.254
PC-3	192.168.1.30/24	192.168.3.10/24 - GW192.168.3.254

La conectividad entre los clientes se refleja en las siguientes figuras, mediante la herramienta PING con el siguiente lineamiento:

- PC-1 ping PC-2
- PC-1 ping PC-3
- PC-2 ping PC-1
- PC-2 ping PC-3
- PC-3 ping PC-1
- PC-3 ping PC-2

```

VPLS                               EVPN
PING 192.168.3.10 (192.168.3.10): 56 data bytes
64 bytes from 192.168.3.10: seq=0 ttl=63 time=16.931 ms
64 bytes from 192.168.3.10: seq=1 ttl=63 time=18.245 ms
64 bytes from 192.168.3.10: seq=2 ttl=63 time=21.982 ms
64 bytes from 192.168.3.10: seq=3 ttl=63 time=15.746 ms
64 bytes from 192.168.3.10: seq=4 ttl=63 time=12.847 ms

PING 192.168.2.10 (192.168.2.10): 56 data bytes
64 bytes from 192.168.2.10: seq=0 ttl=63 time=19.623 ms
64 bytes from 192.168.2.10: seq=1 ttl=63 time=16.382 ms
64 bytes from 192.168.2.10: seq=2 ttl=63 time=14.395 ms
64 bytes from 192.168.2.10: seq=3 ttl=63 time=24.405 ms
64 bytes from 192.168.2.10: seq=4 ttl=63 time=14.634 ms
^^

PING 192.168.1.30 (192.168.1.30): 56 data bytes
64 bytes from 192.168.1.30: seq=0 ttl=64 time=21.597 ms
64 bytes from 192.168.1.30: seq=1 ttl=64 time=13.963 ms
64 bytes from 192.168.1.30: seq=2 ttl=64 time=29.240 ms
64 bytes from 192.168.1.30: seq=3 ttl=64 time=16.577 ms
64 bytes from 192.168.1.30: seq=4 ttl=64 time=19.716 ms

PING 192.168.1.20 (192.168.1.20): 56 data bytes
64 bytes from 192.168.1.20: seq=0 ttl=64 time=21.434 ms
64 bytes from 192.168.1.20: seq=1 ttl=64 time=22.839 ms
64 bytes from 192.168.1.20: seq=2 ttl=64 time=24.060 ms
64 bytes from 192.168.1.20: seq=3 ttl=64 time=14.781 ms
64 bytes from 192.168.1.20: seq=4 ttl=64 time=16.124 ms

```

Figura 58. Conectividad Ping PC-1 a PC-2 y PC-3

```

VPLS
PC-2> ping 192.168.1.10
84 bytes from 192.168.1.10 icmp_seq=1 ttl=63 time=24.578 ms
84 bytes from 192.168.1.10 icmp_seq=2 ttl=63 time=14.872 ms
84 bytes from 192.168.1.10 icmp_seq=3 ttl=63 time=18.167 ms
84 bytes from 192.168.1.10 icmp_seq=4 ttl=63 time=18.899 ms
84 bytes from 192.168.1.10 icmp_seq=5 ttl=63 time=13.192 ms

PC-2> ping 192.168.3.10
84 bytes from 192.168.3.10 icmp_seq=1 ttl=63 time=14.378 ms
84 bytes from 192.168.3.10 icmp_seq=2 ttl=63 time=14.282 ms
84 bytes from 192.168.3.10 icmp_seq=3 ttl=63 time=79.818 ms
84 bytes from 192.168.3.10 icmp_seq=4 ttl=63 time=22.109 ms
84 bytes from 192.168.3.10 icmp_seq=5 ttl=63 time=13.979 ms

EVPN
PC-2> ping 192.168.1.10
84 bytes from 192.168.1.10 icmp_seq=1 ttl=64 time=22.742 ms
84 bytes from 192.168.1.10 icmp_seq=2 ttl=64 time=18.635 ms
84 bytes from 192.168.1.10 icmp_seq=3 ttl=64 time=13.123 ms
84 bytes from 192.168.1.10 icmp_seq=4 ttl=64 time=19.402 ms
84 bytes from 192.168.1.10 icmp_seq=5 ttl=64 time=16.577 ms
^C
PC-2> ping 192.168.1.30
84 bytes from 192.168.1.30 icmp_seq=1 ttl=64 time=20.562 ms
84 bytes from 192.168.1.30 icmp_seq=2 ttl=64 time=36.077 ms
84 bytes from 192.168.1.30 icmp_seq=3 ttl=64 time=17.269 ms
84 bytes from 192.168.1.30 icmp_seq=4 ttl=64 time=16.676 ms
84 bytes from 192.168.1.30 icmp_seq=5 ttl=64 time=16.454 ms

```

Figura 59. Conectividad Ping PC-2 a PC-1 y PC-3

```
VPLS
PC-3> ping 192.168.1.10
84 bytes from 192.168.1.10 icmp_seq=1 ttl=63 time=18.139 ms
84 bytes from 192.168.1.10 icmp_seq=2 ttl=63 time=14.396 ms
84 bytes from 192.168.1.10 icmp_seq=3 ttl=63 time=12.097 ms
84 bytes from 192.168.1.10 icmp_seq=4 ttl=63 time=15.264 ms
84 bytes from 192.168.1.10 icmp_seq=5 ttl=63 time=12.276 ms

PC-3> ping 192.168.2.10
84 bytes from 192.168.2.10 icmp_seq=1 ttl=63 time=11.748 ms
84 bytes from 192.168.2.10 icmp_seq=2 ttl=63 time=11.635 ms
84 bytes from 192.168.2.10 icmp_seq=3 ttl=63 time=13.111 ms
84 bytes from 192.168.2.10 icmp_seq=4 ttl=63 time=11.993 ms
84 bytes from 192.168.2.10 icmp_seq=5 ttl=63 time=13.432 ms

EVPN
PC-3> ping 192.168.1.10
84 bytes from 192.168.1.10 icmp_seq=1 ttl=64 time=19.083 ms
84 bytes from 192.168.1.10 icmp_seq=2 ttl=64 time=14.552 ms
84 bytes from 192.168.1.10 icmp_seq=3 ttl=64 time=17.164 ms
84 bytes from 192.168.1.10 icmp_seq=4 ttl=64 time=38.943 ms
84 bytes from 192.168.1.10 icmp_seq=5 ttl=64 time=24.926 ms

PC-3> ping 192.168.1.20
84 bytes from 192.168.1.20 icmp_seq=1 ttl=64 time=18.560 ms
84 bytes from 192.168.1.20 icmp_seq=2 ttl=64 time=14.551 ms
84 bytes from 192.168.1.20 icmp_seq=3 ttl=64 time=12.752 ms
84 bytes from 192.168.1.20 icmp_seq=4 ttl=64 time=99.443 ms
84 bytes from 192.168.1.20 icmp_seq=5 ttl=64 time=13.256 ms
```

Figura 60. Conectividad Ping PC-3 a PC-1 y PC-2

La información de alcanzabilidad es totalmente transparente desde el punto de vista del cliente, esto se demuestra realizando un trazado desde un cliente hacia el otro con el siguiente lineamiento, con la ayuda de la herramienta tracert (traceroute):

- PC-1 tracert PC-2
- PC-1 tracert PC-3
- PC-2 tracert PC-1
- PC-2 tracert PC-3
- PC-3 tracert PC-1
- PC-3 tracert PC-1


```

VPLS gns3@box:~$ traceroute 192.168.1.20
traceroute to 192.168.1.20 (192.168.1.20), 30 hops max, 38 byte packets
 1 192.168.1.20 (192.168.1.20) 103.193 ms 77.008 ms 62.236 ms

gns3@box:~$ traceroute 192.168.1.20
traceroute to 192.168.1.20 (192.168.1.20), 30 hops max, 38 byte packets
 1 192.168.1.20 (192.168.1.20) 103.193 ms 77.008 ms 62.236 ms
 2 192.168.1.20 (192.168.1.20) 103.193 ms 77.008 ms 62.236 ms

EVPN gns3@box:~$ traceroute 192.168.2.10
traceroute to 192.168.2.10 (192.168.2.10), 30 hops max, 38 byte packets
 1 192.168.1.254 (192.168.1.254) 2.547 ms 1.276 ms 1.523 ms
 2 192.168.2.10 (192.168.2.10) 22.959 ms 17.579 ms 25.779 ms

gns3@box:~$ traceroute 192.168.3.10
traceroute to 192.168.3.10 (192.168.3.10), 30 hops max, 38 byte packets
 1 192.168.1.254 (192.168.1.254) 1.854 ms 3.027 ms 3.066 ms
 2 192.168.3.10 (192.168.3.10) 15.124 ms 37.442 ms 35.207 ms

```

Figura 61. Trazado PC-1 a PC-2 y PC-3

```

VPLS PC-2> trace 192.168.1.10
trace to 192.168.1.10, 8 hops max, press Ctrl+C to stop
 1 *192.168.1.10 25.630 ms (ICMP type:3, code:3, Destination port unreachable)

PC-2> trace 192.168.1.30
trace to 192.168.1.30, 8 hops max, press Ctrl+C to stop
 1 *192.168.1.30 41.571 ms (ICMP type:3, code:3, Destination port unreachable)

EVPN PC-2> trace 192.168.1.10
trace to 192.168.1.10, 8 hops max, press Ctrl+C to stop
 1 192.168.2.254 3.075 ms 2.259 ms 2.124 ms
 2 *192.168.1.10 15.945 ms (ICMP type:3, code:3, Destination port unreachable)
) *

PC-2> trace 192.168.3.10
trace to 192.168.3.10, 8 hops max, press Ctrl+C to stop
 1 192.168.2.254 12.028 ms 3.201 ms 7.339 ms
 2 *192.168.3.10 12.158 ms (ICMP type:3, code:3, Destination port unreachable)
)

```

Figura 62. Trazado PC-2 a PC-1 y PC-3

```

VPLS PC-3> trace 192.168.1.10
trace to 192.168.1.10, 8 hops max, press Ctrl+C to stop
 1 *192.168.1.10 15.437 ms (ICMP type:3, code:3, Destination port unreachable)

PC-3> trace 192.168.1.20
trace to 192.168.1.20, 8 hops max, press Ctrl+C to stop
 1 *192.168.1.20 19.350 ms (ICMP type:3, code:3, Destination port unreachable)

EVPN PC-3> trace 192.168.2.10
trace to 192.168.2.10, 8 hops max, press Ctrl+C to stop
 1 192.168.3.254 2.596 ms 2.121 ms 1.807 ms
 2 *192.168.2.10 9.358 ms (ICMP type:3, code:3, Destination port unreachable)

PC-3> trace 192.168.1.10
trace to 192.168.1.10, 8 hops max, press Ctrl+C to stop
 1 192.168.3.254 4.316 ms 1.988 ms 2.889 ms
 2 *192.168.1.10 17.987 ms (ICMP type:3, code:3, Destination port unreachable)
)

```

Figura 63. Trazado PC-3 a PC-1 y PC-2.

Generando tráfico ICMP con la herramienta PING desde el sitio del cliente PC-2 al sitio del cliente PC-3. Intencionalmente, fallamos una de las rutas en la red central (Core) y se observa un número de Paquetes ICMP perdidos antes de que la red vuelva a converger. Con este método, registramos el tiempo de convergencia de la red observando a VPLS y EVPN individualmente. Para lograr la medición del tiempo se utiliza la herramienta de sniffing Wireshark, la misma que captura todos los paquetes que transitan en una ruta definida para la medición. El escenario de pruebas y la metodología de pruebas es el siguiente:

1. El enlace para realizar el fallo es el conectado desde P2-R5 a P3-R6, lo denominaremos A.
2. Las capturas del tráfico se las realiza en el enlace que va de PE2-R2 a P2-R5, lo denominaremos B.
3. La interrupción o fallo del enlace tiene intervalos de 60s.

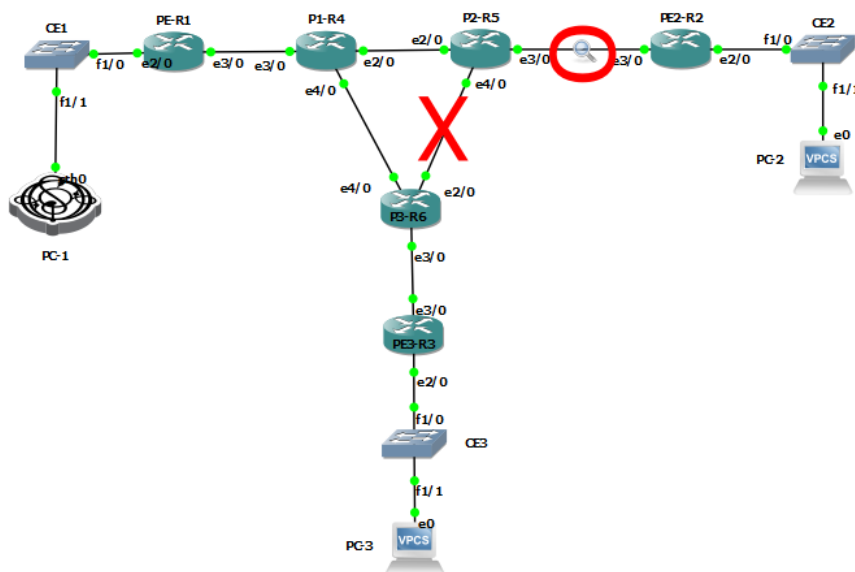


Figura 64. Escenario de pruebas -Convergencia-

En el escenario de pruebas se realizaron diez cortes o fallos del enlace A tanto para VPLS como para EVPN. Las mediciones son reflejadas con los mensajes “Echo (request)” y “Echo

(replay)”, donde se mide el tiempo desde que el enlace A es “cortado”, hasta que la respuesta “Echo (replay)” del cliente PC-3 es vista por el cliente PC-2. El tráfico capturado por el enlace B se muestra a continuación.

No.	Time	Source	Destination	Protocol	Length	Info
845	05:19:52,123973	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0xf830, seq=250/64000, ttl=63 (no response found!)
848	05:19:54,125006	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0xfa30, seq=251/64256, ttl=63 (no response found!)
851	05:19:56,126096	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0xfc30, seq=252/64512, ttl=63 (no response found!)
853	05:19:58,126493	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0xfe30, seq=253/64768, ttl=63 (no response found!)
860	05:20:00,129022	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0031, seq=254/65024, ttl=63 (no response found!)
864	05:20:02,128592	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0231, seq=255/65280, ttl=63 (no response found!)
868	05:20:04,132881	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0431, seq=256/1, ttl=63 (no response found!)
871	05:20:06,130786	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0631, seq=257/257, ttl=63 (no response found!)
875	05:20:08,132242	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0831, seq=258/513, ttl=63 (no response found!)
877	05:20:10,132897	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0a31, seq=259/769, ttl=63 (no response found!)
884	05:20:12,133361	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0c31, seq=260/1025, ttl=63 (no response found!)
887	05:20:14,136291	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x0e31, seq=261/1281, ttl=63 (no response found!)
891	05:20:16,136456	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1031, seq=262/1537, ttl=63 (no response found!)
915	05:20:27,164522	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1b31, seq=272/4097, ttl=63 (reply in 916)
916	05:20:27,177091	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x1b31, seq=272/4097, ttl=63 (request in 915)
918	05:20:28,188131	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1c31, seq=273/4353, ttl=63 (reply in 919)
919	05:20:28,202680	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x1c31, seq=273/4353, ttl=63 (request in 918)
921	05:20:29,243978	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1d31, seq=274/4609, ttl=63 (reply in 922)
922	05:20:29,251544	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x1d31, seq=274/4609, ttl=63 (request in 921)
925	05:20:30,255517	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1e31, seq=275/4865, ttl=63 (reply in 926)
926	05:20:30,264725	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x1e31, seq=275/4865, ttl=63 (request in 925)
930	05:20:31,268070	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1f31, seq=276/5121, ttl=63 (reply in 931)
931	05:20:31,279455	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x1f31, seq=276/5121, ttl=63 (request in 930)
935	05:20:32,284977	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x2031, seq=277/5377, ttl=63 (reply in 936)
936	05:20:32,292046	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x2031, seq=277/5377, ttl=63 (request in 935)
938	05:20:33,296073	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x2131, seq=278/5633, ttl=63 (reply in 939)
939	05:20:33,303855	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x2131, seq=278/5633, ttl=63 (request in 938)
No.	Time	Source	Destination	Protocol	Length	Info
888	05:20:15,366532	10.0.22.1	224.0.0.2	LDP	84	Hello Message
889	05:20:15,556711	2.2.2.2	3.3.3.3	TCP	104	[TCP Retransmission] 179 → 54799 [PSH, ACK] Seq=766 Ack=766 Win=16384 Len=38 TSval=63072 TSecr=67011
890	05:20:15,685012	10.0.22.2	224.0.0.5	OSPF	94	Hello Packet
891	05:20:16,136456	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1031, seq=262/1537, ttl=63 (no response found!)
892	05:20:16,315874	10.0.22.2	224.0.0.5	OSPF	166	LS Update
893	05:20:16,537475	2.2.2.2	5.5.5.5	LDP	104	Label Withdrawal Message
894	05:20:16,542114	5.5.5.5	2.2.2.2	LDP	104	Label Withdrawal Message
895	05:20:16,637866	5.5.5.5	2.2.2.2	LDP	104	Label Release Message
896	05:20:16,640446	2.2.2.2	5.5.5.5	LDP	104	Label Release Message
897	05:20:16,739959	5.5.5.5	2.2.2.2	TCP	66	56662 → 646 [ACK] Seq=881 Ack=899 Win=16384 Len=0 TSval=77721 TSecr=63175
898	05:20:16,934878	10.0.22.2	224.0.0.2	LDP	84	Hello Message
899	05:20:17,351349	10.0.22.1	224.0.0.5	OSPF	98	LS Acknowledge
900	05:20:20,129078	10.0.22.1	224.0.0.2	LDP	84	Hello Message
901	05:20:21,687155	10.0.22.2	224.0.0.2	LDP	84	Hello Message
902	05:20:21,865681	10.0.22.1	224.0.0.5	OSPF	94	Hello Packet
903	05:20:23,661100	10.0.22.2	224.0.0.5	OSPF	94	Hello Packet
904	05:20:24,006742	10.0.22.1	224.0.0.2	LDP	84	Hello Message
905	05:20:25,959399	10.0.22.2	224.0.0.5	OSPF	110	LS Update
906	05:20:26,008713	10.0.22.2	224.0.0.5	OSPF	214	LS Update
907	05:20:26,273455	10.0.22.2	224.0.0.2	LDP	84	Hello Message
908	05:20:26,568060	5.5.5.5	2.2.2.2	LDP	84	Keep Alive Message
909	05:20:26,663161	2.2.2.2	5.5.5.5	TCP	66	646 → 56662 [ACK] Seq=899 Ack=899 Win=16384 Len=0 TSval=64167 TSecr=78682
910	05:20:26,674547	2.2.2.2	5.5.5.5	LDP	84	Keep Alive Message
911	05:20:26,771928	5.5.5.5	2.2.2.2	TCP	66	56662 → 646 [ACK] Seq=899 Ack=917 Win=16384 Len=0 TSval=78702 TSecr=64168
912	05:20:26,987053	10.0.22.1	224.0.0.5	OSPF	118	LS Acknowledge
913	05:20:27,077069	5.5.5.5	2.2.2.2	LDP	104	Label Mapping Message
914	05:20:27,089091	2.2.2.2	5.5.5.5	LDP	104	Label Mapping Message
915	05:20:27,164522	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1b31, seq=272/4097, ttl=63 (reply in 916)
916	05:20:27,177091	192.168.3.10	192.168.2.10	ICMP	120	Echo (ping) reply id=0x1b31, seq=272/4097, ttl=63 (request in 915)
917	05:20:27,192773	5.5.5.5	2.2.2.2	TCP	66	56662 → 646 [ACK] Seq=937 Ack=955 Win=16384 Len=0 TSval=78743 TSecr=64208
918	05:20:28,188131	192.168.2.10	192.168.3.10	ICMP	124	Echo (ping) request id=0x1c31, seq=273/4353, ttl=63 (reply in 919)

Figura 65. Captura de paquetes EVPN

Se verifican los ítems desde el número 891 hasta el 915 donde el tiempo de reacción de la red es de 11s, desde la hora 05:20:16 hasta 05:20:27. Del mismo modo son validadas las nueve interrupciones restantes en el escenario de pruebas, en la siguiente tabla se muestran los resultados.

Tabla 6.
Convergencia EVPN

No. Interrupción	Tiempo de convergencia	
	[s]	
1	11	
2	6	
3	4	
4	5	
5	11	
6	10	
7	9	
8	7	
9	8	
10	5	

No.	Time	Source	Destination	Protocol	Length	Info
4037	16:53:55,481364	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xa3d3, seq=1000/59395, ttl=64 (reply in 4038)
4038	16:53:55,494549	192.168.1.30	192.168.1.20	ICMP	120	Echo (ping) reply id=0xa3d3, seq=1000/59395, ttl=64 (request in 4037)
4039	16:53:56,499373	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xa4d3, seq=1001/59651, ttl=64 (no response found)
4043	16:53:58,540223	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xa6d3, seq=1002/59907, ttl=64 (no response found)
4047	16:54:00,502402	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xa8d3, seq=1003/60163, ttl=64 (no response found)
4053	16:54:02,502164	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xaad3, seq=1004/60419, ttl=64 (no response found)
4058	16:54:04,502859	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xacd3, seq=1005/60675, ttl=64 (no response found)
4063	16:54:06,503352	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xae3d3, seq=1006/60931, ttl=64 (no response found)
4065	16:54:08,503056	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xb0d3, seq=1007/61187, ttl=64 (no response found)
4069	16:54:10,503767	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xb2d3, seq=1008/61443, ttl=64 (no response found)
4072	16:54:12,500208	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xb4d3, seq=1009/61699, ttl=64 (no response found)
4080	16:54:14,506903	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xb6d3, seq=1010/61955, ttl=64 (no response found)
4087	16:54:16,507471	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xb8d3, seq=1011/62211, ttl=64 (no response found)
4089	16:54:18,508528	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xbad3, seq=1012/62467, ttl=64 (no response found)
4093	16:54:20,508398	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xbcd3, seq=1013/62723, ttl=64 (no response found)
4099	16:54:22,509597	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xbdd3, seq=1014/62979, ttl=64 (no response found)
4103	16:54:24,513010	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xc0d3, seq=1015/63235, ttl=64 (no response found)
4108	16:54:26,512739	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xc2d3, seq=1016/63491, ttl=64 (no response found)
4111	16:54:28,513004	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xc4d3, seq=1017/63747, ttl=64 (no response found)
4115	16:54:30,514957	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xc6d3, seq=1018/64003, ttl=64 (no response found)
4121	16:54:32,516076	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xc8d3, seq=1019/64259, ttl=64 (no response found)
4124	16:54:34,518199	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xcad3, seq=1020/64515, ttl=64 (no response found)
4129	16:54:36,517576	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xcdd3, seq=1021/64771, ttl=64 (no response found)
4133	16:54:38,519905	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xcd3, seq=1022/65027, ttl=64 (no response found)
4141	16:54:40,519965	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xd0d3, seq=1023/65283, ttl=64 (no response found)
4146	16:54:42,533695	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xd2d3, seq=1024/4, ttl=64 (no response found)
4151	16:54:44,532690	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xd4d3, seq=1025/260, ttl=64 (no response found)
4153	16:54:46,531483	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xd6d3, seq=1026/516, ttl=64 (no response found)
4160	16:54:48,533839	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xd8d3, seq=1027/772, ttl=64 (no response found)
4166	16:54:50,534892	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xdad3, seq=1028/1028, ttl=64 (no response found)
4167	16:54:50,536168	192.168.1.20	192.168.1.30	ICMP	124	Echo (ping) request id=0xdad3, seq=1028/1028, ttl=64 (reply in 4160)

CONTINÚA

Se verifican los ítems desde el número 4039 hasta el 4100 donde el tiempo de reacción de la red es de 11s, desde la hora 16:53:56 hasta 16:54:50. Del mismo modo son validadas las nueve interrupciones restantes en el escenario de pruebas, en la siguiente tabla se muestran los resultados.

Tabla 7.
Convergencia VPLS

No. Interrupción	Tiempo de convergencia [s]
1	54
2	45
3	50
4	40
5	45
6	46
7	51
8	44
9	40
10	40

De este modo y con las diez interrupciones realizadas en los escenarios propuestos, los resultados son los siguientes.

Tabla 8.
Convergencia VPLS vs EVPN

Tecnología	Tiempo de Convergencia Promedio[s]
VPLS	45.5
EVPN	7.6

4.5.2. Movilidad MAC

El escenario de pruebas tanto para VPLS como para EVPN es el siguiente:

La topología inicia con tres clientes PC-1, PC-2 y PC-3, respectivamente en sus CE y PE como se muestra en la figura.

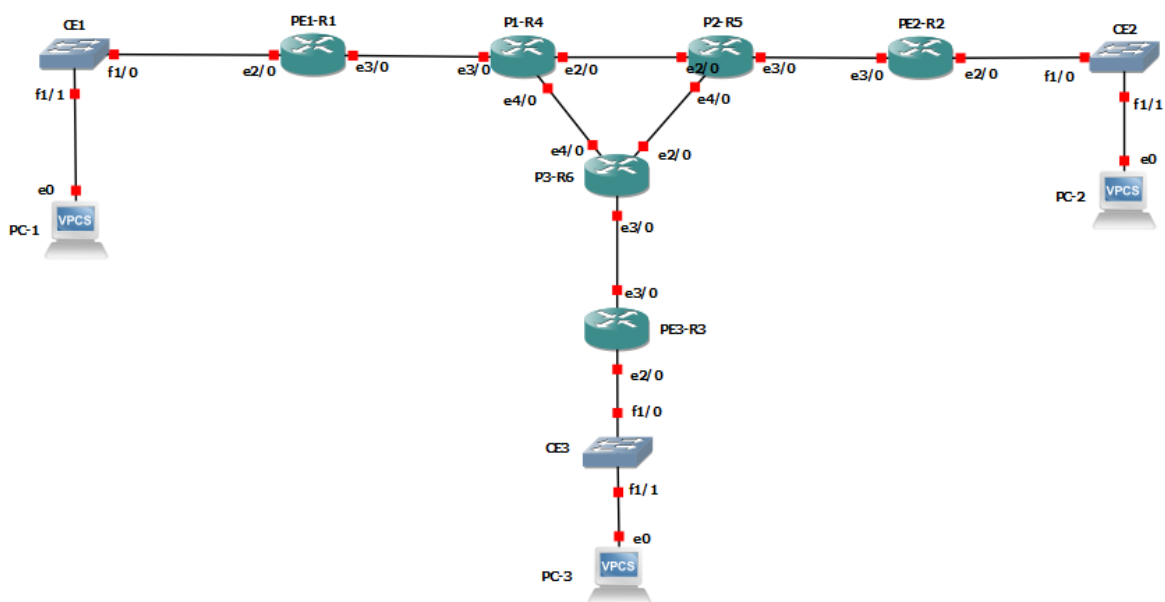


Figura 67. Escenario de pruebas Movilidad MAC

Para verificar la movilidad MAC que se produce o no en las tecnologías VPLS/EVPN, se desconectará la PC-1 y se reubicará la PC-2 hacia el CE1 quedando CE2 sin PC, mostrado en la siguiente figura.

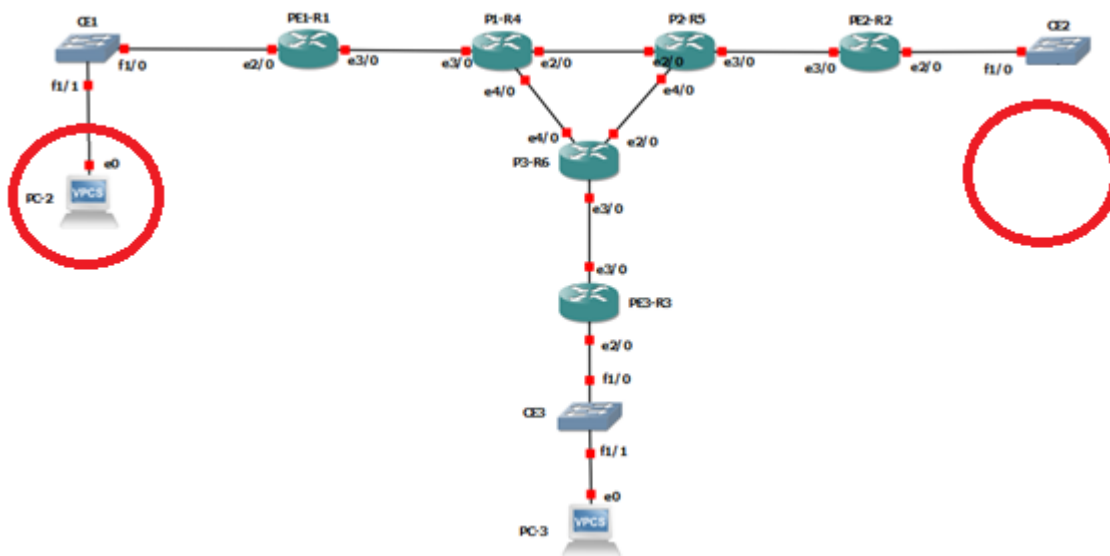


Figura 68. Movilidad MAC

Para demostrar los cambios efectuados por las dos tecnologías se realizará el análisis individualmente.

4.5.2.1.EVPN

Se identifica las direcciones MAC de cada dispositivo PC en cada PE, así tenemos:

- PC-1 00:50:79:66:68:01
- PC-2 00:50:79:66:68:00
- PC-3 00:50:79:66:68:02


```

root@R1-PE1> show evpn mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : evpna
Bridging domain : __evpna__, VLAN : 10
MAC          MAC          Logical      NH      RTR
address      flags      interface    Index  ID
00:50:79:66:68:00  DC
00:50:79:66:68:01  D    ge-0/0/0.0
00:50:79:66:68:02  DC
c0:01:06:91:f1:00  DC
c0:03:08:a5:00:00  DC

root@R2-PE2> show evpn mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : evpna
Bridging domain : __evpna__, VLAN : 10
MAC          MAC          Logical      NH      RTR
address      flags      interface    Index  ID
00:50:79:66:68:00  D    ge-0/0/0.0
00:50:79:66:68:01  DC
00:50:79:66:68:02  DC
c0:01:06:91:f1:00  D    ge-0/0/0.0
c0:03:08:a5:00:00  DC

root@R3-PE3> show evpn mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : evpna
Bridging domain : __evpna__, VLAN : 10
MAC          MAC          Logical      NH      RTR
address      flags      interface    Index  ID
00:50:79:66:68:00  DC
00:50:79:66:68:01  DC
00:50:79:66:68:02  D    ge-0/0/0.0
c0:01:06:91:f1:00  DC

```

Figura 69. Tabla-MAC EVPN

Definidas las direcciones MAC realizamos la reubicación del cliente PC-2 y capturamos los paquetes mediante el sniffer Wireshark en el enlace entre el PE1 – P1 obteniendo los siguientes resultados.

No.	Time	Source	Destination	Protocol	Length	Info
50	02:47:01,535151	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
68	02:47:12,125088	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
95	02:47:26,720220	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
96	02:47:26,721211	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
104	02:47:29,608664	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
123	02:47:37,276944	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
158	02:47:55,260790	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
159	02:47:55,262989	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
164	02:47:56,834972	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
181	02:48:02,313313	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
214	02:48:22,380076	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
218	02:48:22,719175	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
231	02:48:27,455135	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
237	02:48:29,993329	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
248	02:48:33,679724	2.2.2.2	1.1.1.1	BGP	162	UPDATE Message
250	02:48:33,797630	2.2.2.2	1.1.1.1	BGP	251	UPDATE Message, UPDATE Message
270	02:48:38,501764	1.1.1.1	2.2.2.2	BGP	135	UPDATE Message
271	02:48:38,506799	1.1.1.1	3.3.3.3	BGP	135	UPDATE Message
275	02:48:38,638074	1.1.1.1	2.2.2.2	BGP	177	UPDATE Message, UPDATE Message
278	02:48:38,759554	1.1.1.1	3.3.3.3	BGP	177	UPDATE Message, UPDATE Message
294	02:48:46,750959	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
305	02:48:51,689028	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
309	02:48:54,268356	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
317	02:48:54,740615	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
351	02:49:15,667453	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
359	02:49:19,692202	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
361	02:49:20,482841	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
365	02:49:21,082984	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message

Figura 70. Captura de paquetes enlace PE1-P1

En la figura 70 se tiene la captura de paquetes donde se realiza la señalización antes-durante y después de realizar la reubicación de la PC-2. Así tenemos que en los paquetes desde el No. 50 a 237 la reubicación aún no es ejecutada, a partir del paquete No. 248 al 278, se efectúa la reubicación del PC-2 y desde el paquete 294 la reubicación es convergida.

EVPN basándose en la señalización BGP envía mensajes de actualización de la información de accesibilidad de la capa de red NLRI hacia todos los PE existentes en la red, de esta manera actualizando la nueva ubicación de las direcciones MAC reubicadas.

Esta actualización se encuentra encapsulada dentro de los mensajes UPDATE mostrados en la figura 106, con este antecedente procedemos a des encapsular al mensaje UPDATE, con el fin de comprobar la movilidad MAC.

Este mensaje es enviado desde PE1 hacia PE2 indicando la dirección MAC que tenía hasta antes de realizar la reubicación de la PC, el NLRI contiene la información del Router distintivo PE2 al que pertenece la dirección MAC en cuestión.

```

> Frame 248: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits)
> Ethernet II, Src: LucentTe_71:c0:01 (00:05:86:71:c0:01), Dst: LucentTe_71:eb:01 (00:05:86:71:eb:01)
> Internet Protocol Version 4, Src: 2.2.2.2, Dst: 1.1.1.1
> Transmission Control Protocol, Src Port: 62456, Dst Port: 179, Seq: 77, Ack: 77, Len: 96
  # Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffffff
    Length: 96
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 73
  # Path attributes
    > Path Attribute - ORIGIN: IGP
    > Path Attribute - AS_PATH: empty
    > Path Attribute - LOCAL_PREF: 100
    > Path Attribute - EXTENDED_COMMUNITIES
    # Path Attribute - MP_REACH_NLRI
      > Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
      Type Code: MP_REACH_NLRI (14)
      Length: 44
      Address family identifier (AFI): Layer-2 VPN (25)
      Subsequent address family identifier (SAFI): EVPN (70)
      Next hop network address (4 bytes)
      Number of Subnetwork points of attachment (SNPA): 0
    # Network layer reachability information (35 bytes)
      # EVPN NLRI: MAC Advertisement Route
        Route Type: MAC Advertisement Route (2)
        Length: 33
        Route Distinguisher: 000102020202000a (2.2.2.2:10)
        > ESI: 00:00:00:00:00:00:00:00:00
        Ethernet Tag ID: 10
        MAC Address Length: 48
        MAC Address: Private_66:68:00 (00:50:79:66:68:00)
        IP Address Length: 0
        > IP Address: NOT INCLUDED
        MPLS Label Stack 1: 299776 (bottom)

```

Figura 71. Mensaje Update 162

En este mensaje se agrega información sobre la dirección IP que posee el PC reubicado. Esta información para el aprendizaje de MAC no es de mucha utilidad, es informativo.

```

  Network layer reachability information (16 bytes)
    BGP Prefix
      Prefix Length: 120
      Label Stack: 16 (bottom)
      Route Distinguisher: 2.2.2.2:1
      MP Reach NLRI IPv4 prefix: 192.168.2.20
Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffffff
  Length: 100
  Type: UPDATE Message (2)
  Withdrawn Routes Length: 0
  Total Path Attribute Length: 77
  Path attributes
    Path Attribute - ORIGIN: IGP
    Path Attribute - AS_PATH: empty
    Path Attribute - LOCAL_PREF: 100
    Path Attribute - EXTENDED_COMMUNITIES
    Path Attribute - MP_REACH_NLRI
      Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
      Type Code: MP_REACH_NLRI (14)
      Length: 48
      Address family identifier (AFI): Layer-2 VPN (25)
      Subsequent address family identifier (SAFI): EVPN (70)
      Next hop network address (4 bytes)
      Number of Subnetwork points of attachment (SNPA): 0
    Network layer reachability information (39 bytes)
      EVPN NLRI: MAC Advertisement Route
        Route Type: MAC Advertisement Route (2)
        Length: 37
        Route Distinguisher: 000102020202000a (2.2.2.2:10)
        ESI: 00:00:00:00:00:00:00:00
        Ethernet Tag ID: 10
        MAC Address Length: 48
        MAC Address: Private 66:68:00 (00:50:79:66:68:00)
        IP Address Length: 32
        IPv4 address: 192.168.2.20
        MPLS Label Stack 1: 299776 (bottom)

```

Figura 72. Mensaje Update 251.

En este mensaje la dirección MAC ya es registrado en el Router distintivo PE1 y es enviada hacia los routers PE2 y PE1 para que actualicen sus tablas MAC.

```

▷ Frame 275: 177 bytes on wire (1416 bits), 177 bytes captured (1416 bits)
▷ Ethernet II, Src: LucentTe_71:eb:01 (00:05:86:71:eb:01), Dst: LucentTe_71:c0:01 (00:05:86:71:c0:01)
▷ Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
▷ Transmission Control Protocol, Src Port: 179, Dst Port: 62456, Seq: 146, Ack: 358, Len: 111
└─ Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 46
  Type: UPDATE Message (2)
  Withdrawn Routes Length: 0
  Total Path Attribute Length: 23
  └─ Path attributes
    └─ Border Gateway Protocol - UPDATE Message
      Marker: ffffffffffffffffffffffffffffffff
      Length: 65
      Type: UPDATE Message (2)
      Withdrawn Routes Length: 0
      Total Path Attribute Length: 42
      └─ Path attributes
        └─ Path Attribute - MP_UNREACH_NLRI
          Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
          Type Code: MP_UNREACH_NLRI (15)
          Length: 38
          Address family identifier (AFI): Layer-2 VPN (25)
          Subsequent address family identifier (SAFI): EVPN (70)
          └─ Withdrawn routes (35 bytes)
            └─ EVPN NLRI: MAC Advertisement Route
              Route Type: MAC Advertisement Route (2)
              Length: 33
              Route Distinguisher: 0001010101000a (1.1.1.1:10)
              └─ ESI: 00:00:00:00:00:00:00:00
                Ethernet Tag ID: 10
                MAC Address Length: 48
                MAC Address: Private_66:68:00 (00:50:79:66:68:00)
                IP Address Length: 0
                └─ IP Address: NOT INCLUDED
                  MPLS Label Stack 1: 0 (bottom)

```

Figura 73. Mensaje Update 177

```

▷ Frame 278: 177 bytes on wire (1416 bits), 177 bytes captured (1416 bits)
▷ Ethernet II, Src: LucentTe_71:eb:01 (00:05:86:71:eb:01), Dst: LucentTe_71:c0:01 (00:05:86:71:c0:01)
▷ Internet Protocol Version 4, Src: 1.1.1.1, Dst: 3.3.3.3
▷ Transmission Control Protocol, Src Port: 179, Dst Port: 62993, Seq: 146, Ack: 96, Len: 111
└─ Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 65
  Type: UPDATE Message (2)
  Withdrawn Routes Length: 0
  Total Path Attribute Length: 42
  └─ Path attributes
    └─ Path Attribute - MP_UNREACH_NLRI
      Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
      Type Code: MP_UNREACH_NLRI (15)
      Length: 38
      Address family identifier (AFI): Layer-2 VPN (25)
      Subsequent address family identifier (SAFI): EVPN (70)
      └─ Withdrawn routes (35 bytes)
        └─ EVPN NLRI: MAC Advertisement Route
          Route Type: MAC Advertisement Route (2)
          Length: 33
          Route Distinguisher: 0001010101000a (1.1.1.1:10)
          └─ ESI: 00:00:00:00:00:00:00:00
            Ethernet Tag ID: 10
            MAC Address Length: 48
            MAC Address: Private_66:68:00 (00:50:79:66:68:00)
            IP Address Length: 0
            └─ IP Address: NOT INCLUDED
              MPLS Label Stack 1: 0 (bottom)

```

Figura 74. Mensaje Update 177 V2.

De esta manera EVPN cumple con actualizar las tablas MAC desde el instante en que un dispositivo terminal es reubicado desde un CE a otro CE. El resultado de este proceso que realiza EVPN a través del intercambio de NLRI queda reflejado en las tablas-MAC EVPN.

```

root@R1-PE1> show evpn mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : evpna
Bridging domain : __evpna__, VLAN : 10
MAC              MAC              Logical              NH      RTR
address          flags              interface            Index  ID
00:50:79:66:68:02 DC
c0:01:06:91:f1:00 D              ge-0/0/0.0
c0:03:08:a5:00:00 DC
root@R2-PE2> show evpn mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : evpna
Bridging domain : __evpna__, VLAN : 10
MAC              MAC              Logical              NH      RTR
address          flags              interface            Index  ID
00:50:79:66:68:02 DC
c0:01:06:91:f1:00 DC
c0:02:08:95:00:00 D              ge-0/0/0.0
c0:03:08:a5:00:00 DC
root@R3-PE3> show evpn mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : evpna
Bridging domain : __evpna__, VLAN : 10
MAC              MAC              Logical              NH      RTR
address          flags              interface            Index  ID
00:50:79:66:68:00 DC
00:50:79:66:68:02 D              ge-0/0/0.0
c0:01:06:91:f1:00 DC
c0:01:08:46:00:00 DC

```

Figura 75. Mensaje Update 177 V3

Este proceso implica que el Router PE3 sabe exactamente a que Router de borde debe enviar los paquetes debido a que su tabla se actualizó, es decir PC-3 puede enviar paquetes a PC-2 sin pérdidas, gracias al aprendizaje de MAC.

4.5.2.2.VPLS

Se identifica las direcciones MAC de cada dispositivo PC en cada PE, así tenemos:

- PC-1 00:50:79:66:68:01
- PC-2 00:50:79:66:68:00
- PC-3 00:50:79:66:68:02

```

root@R1-PE1> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls
Bridging domain : __vpls__, VLAN : NA
  MAC          MAC          Logical          NH    RTR
  address      flags      interface      Index  ID
00:50:79:66:68:00  D        lsi.1048576
00:50:79:66:68:01  D        ge-0/0/0.10
00:50:79:66:68:02  D        lsi.1048577
c0:01:08:c9:f1:00  D        ge-0/0/0.10
c0:02:08:d8:f1:00  D

root@R2-PE2> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls
Bridging domain : __vpls__, VLAN : NA
  MAC          MAC          Logical          NH    RTR
  address      flags      interface      Index  ID
00:50:79:66:68:00  D        ge-0/0/0.10
00:50:79:66:68:01  D        lsi.1048575
00:50:79:66:68:02  D        lsi.1048577
c0:01:08:c9:f1:00  D

root@R3-PE3> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls
Bridging domain : __vpls__, VLAN : NA
  MAC          MAC          Logical          NH    RTR
  address      flags      interface      Index  ID
00:50:79:66:68:00  D        lsi.1048577
00:50:79:66:68:01  D        lsi.1048575
00:50:79:66:68:02  D        ge-0/0/0.10
c0:01:08:c9:f1:00  D

```

Figura 76. Tabla-MAC VPLS.

Definidas las direcciones MAC realizamos la reubicación del cliente PC-2 y capturamos los paquetes mediante el sniffer Wireshark en el enlace entre el PE1 – P1 obteniendo los siguientes resultados.

No.	Time	Source	Destination	Protocol	Length	Info
216	03:32:44,856485	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
249	03:32:52,885802	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
261	03:32:56,019629	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
297	03:33:05,207095	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
331	03:33:14,393513	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
357	03:33:20,344791	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
369	03:33:23,089965	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
416	03:33:35,586302	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
435	03:33:40,642667	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
460	03:33:45,371988	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
487	03:33:52,422451	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
528	03:34:03,312156	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
537	03:34:05,512101	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
561	03:34:10,001274	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
607	03:34:22,276046	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
634	03:34:28,885787	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
645	03:34:32,140295	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
676	03:34:39,339557	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
721	03:34:51,639478	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
752	03:34:59,220089	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
757	03:35:00,683176	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
779	03:35:05,843171	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
826	03:35:17,598823	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
869	03:35:28,918357	3.3.3.3	1.1.1.1	BGP	85	KEEPALIVE Message
877	03:35:30,538260	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message
882	03:35:31,720711	1.1.1.1	2.2.2.2	BGP	85	KEEPALIVE Message
928	03:35:44,177989	2.2.2.2	1.1.1.1	BGP	85	KEEPALIVE Message
972	03:35:56,063718	1.1.1.1	3.3.3.3	BGP	85	KEEPALIVE Message

Figura 77. Captura de paquetes enlace PE1-P1

En la figura 77 se observa que no existe ningún tipo de mensaje del protocolo BGP asociado al aprendizaje MAC esto implicaría que la tabla MAC no sufriría ningún tipo de cambio, salvo sobre los dispositivos que se retiran de la red como sería el caso del PC-1, esto se demuestra con las tablas MAC VPLS mostradas a continuación


```

root@R1-PE1> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls
Bridging domain : __vpls__, VLAN : NA
MAC          MAC          Logical      NH      RTR
address      flags      interface    Index   ID
00:50:79:66:68:00 D          ge-0/0/0.10
00:50:79:66:68:02 D          lsi.1048577
c0:01:08:c9:f1:00 D          ge-0/0/0.10

root@R2-PE2> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls
Bridging domain : __vpls__, VLAN : NA
MAC          MAC          Logical      NH      RTR
address      flags      interface    Index   ID
00:50:79:66:68:00 D          lsi.1048576
c0:01:08:c9:f1:00 D          lsi.1048576

root@R3-PE3> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls
Bridging domain : __vpls__, VLAN : NA
MAC          MAC          Logical      NH      RTR
address      flags      interface    Index   ID
00:50:79:66:68:00 D          lsi.1048576
00:50:79:66:68:02 D          ge-0/0/0.10
c0:01:08:c9:f1:00 D          lsi.1048576

```

Figura 78. Tabla-MAC VPLS

En la tabla MAC se puede observar que en el Router PE3 la interfaz lógica no cambia al igual que en el Router PE2, por lo que se determina que no existe aprendizaje de MAC, por tanto, el Router PE3 al no modificar su tabla MAC sabe que debe enviar los paquetes a PE2 por lo que existirá pérdida de paquetes hasta que PE2 le diga que no tiene esa dirección MAC en su tabla, y BGP tenga que enviar mensajes de difusión para aprender de nuevo la ubicación de ese destino.

4.5.3. Supresión de Tráfico BUM

Para verificar el comportamiento que presenta cada tecnología ante una respuesta de tráfico por inundación, el escenario de pruebas es el siguiente:

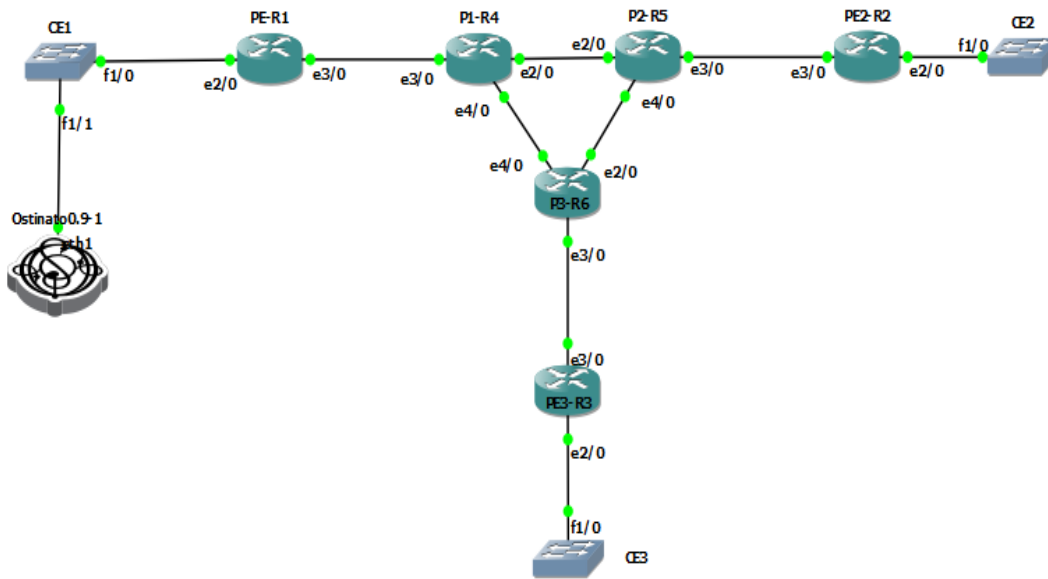


Figura 79. Escenario de pruebas tráfico BUM

Mediante la herramienta Ostinato, generamos flujos del tipo ARP con múltiples destinos y múltiples fuentes, tanto para el escenario de VPLS como en EVPN.

Los lineamientos generales para generar los flujos son los siguientes:

- Tipo de protocolo: ARP
- Direcciones MAC fuente: 22:22:22:22:22:22 (incremento en 1)
- Direcciones MAC destino: 11:11:11:11:11:11 (incremento en 1)
- Direcciones IP fuente: 192.168.1.10 (incremento en 1)
- Direcciones IP destino: 192.168.1.20 VPLS – 192.168.2.10 EVPN (incremento en 1)
- Número de paquetes por segundo: 100

En Ostinato configuramos en el puerto 0 ether1 un nuevo flujo como se muestra a continuación:

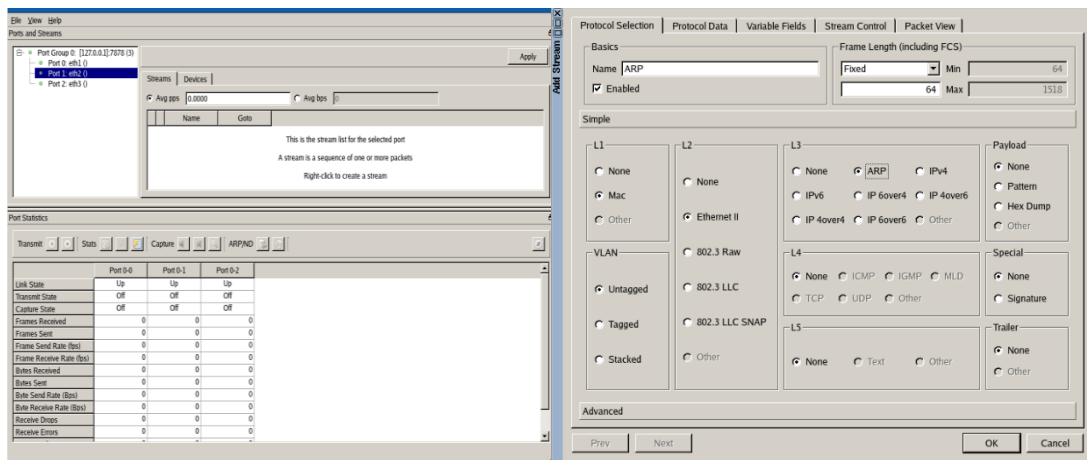


Figura 80. Configuración tipo de Protocolo

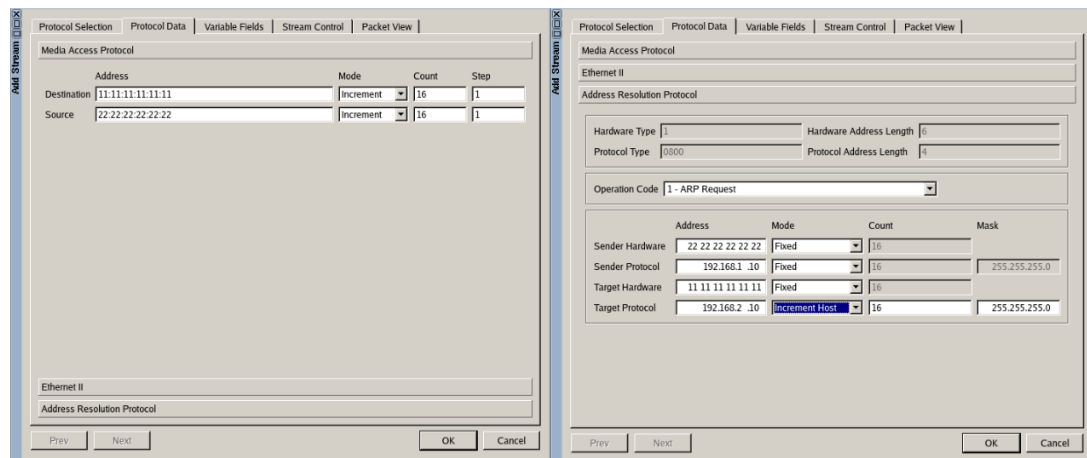


Figura 81. Configuración direcciones MAC e IP.

Ahora bien, con el escenario definido se realiza la inundación del tráfico ARP en dos ocasiones para verificar que sucede con el aprendizaje MAC; en el enlace PE1-P1 se realiza la captura del tráfico con Wireshark.

4.5.3.1.EVPN

En la primera inundación de tráfico ARP se obtienen los siguientes resultados:

No.	Time	Source	Destination	Protocol	Length	Info
138	22:43:29,563568	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.20? Tell 192.168.1.10
139	22:43:30,563939	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.21? Tell 192.168.1.10
143	22:43:31,566624	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.22? Tell 192.168.1.10
145	22:43:32,565444	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.23? Tell 192.168.1.10
146	22:43:33,566713	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.24? Tell 192.168.1.10
147	22:43:34,580873	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.25? Tell 192.168.1.10
151	22:43:35,576509	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.26? Tell 192.168.1.10
152	22:43:36,576804	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.27? Tell 192.168.1.10
153	22:43:37,577436	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.28? Tell 192.168.1.10
157	22:43:38,577686	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.29? Tell 192.168.1.10
372	22:47:23,856368	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.20? Tell 192.168.1.10
376	22:47:24,857224	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.21? Tell 192.168.1.10
377	22:47:25,857343	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.22? Tell 192.168.1.10
378	22:47:26,858372	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.23? Tell 192.168.1.10
382	22:47:27,858211	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.24? Tell 192.168.1.10
383	22:47:28,858128	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.25? Tell 192.168.1.10
384	22:47:29,858716	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.26? Tell 192.168.1.10
385	22:47:30,859036	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.27? Tell 192.168.1.10
389	22:47:31,860008	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.28? Tell 192.168.1.10
390	22:47:32,860254	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.29? Tell 192.168.1.10
391	22:47:33,860668	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.30? Tell 192.168.1.10
395	22:47:34,863837	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.31? Tell 192.168.1.10
396	22:47:35,863565	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.32? Tell 192.168.1.10
397	22:47:36,863947	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.33? Tell 192.168.1.10
401	22:47:37,864649	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.34? Tell 192.168.1.10
402	22:47:38,865470	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.35? Tell 192.168.1.10
403	22:47:39,866474	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.2.20? Tell 192.168.1.10

Figura 82. Inundación ARP -antes-

En la segunda inundación se obtienen los siguientes resultados:

No.	Time	Source	Destination	Protocol	Length	Info
1256	22:47:21,408907	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029
1259	22:47:23,876929	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1260	22:47:24,648541	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029
1262	22:47:24,867822	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1263	22:47:25,885889	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1266	22:47:26,869600	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1271	22:47:27,785095	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029
1272	22:47:27,872059	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1273	22:47:28,873790	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1274	22:47:29,866313	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1276	22:47:30,872338	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1278	22:47:30,946619	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029
1281	22:47:31,877567	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1283	22:47:32,893762	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1284	22:47:33,872686	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1285	22:47:34,068294	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029
1288	22:47:34,879415	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1290	22:47:35,878217	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1292	22:47:36,886777	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1294	22:47:37,282781	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029
1296	22:47:37,886234	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1299	22:47:38,880938	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1301	22:47:39,884772	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1305	22:47:40,475636	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029
1306	22:47:40,899967	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1307	22:47:41,888234	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1308	22:47:42,878946	LucentTe_71:a2:01	LucentTe_71:33:01	MPLS	82	MPLS Label Switched Packet
1310	22:47:43,573133	f1:00:81:00:00:0a	cc:cd:c0:01:0a:a4	STP	86	Conf. Root = 32768/0/c0:01:08:46:00:01 Cost = 0 Port = 0x8029

Figura 83. Inundación ARP-después-

```

root@R1-PE1> show evpn mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : evpna
Bridging domain : __evpna__, VLAN : 10

```

MAC address	MAC flags	Logical interface	NH Index	RTR ID
22:22:22:22:22:22	D	ge-0/0/0.0		
22:22:22:22:22:23	D	ge-0/0/0.0		
22:22:22:22:22:24	D	ge-0/0/0.0		
22:22:22:22:22:25	D	ge-0/0/0.0		
22:22:22:22:22:26	D	ge-0/0/0.0		
22:22:22:22:22:27	D	ge-0/0/0.0		
22:22:22:22:22:28	D	ge-0/0/0.0		
22:22:22:22:22:29	D	ge-0/0/0.0		
22:22:22:22:22:2a	D	ge-0/0/0.0		
22:22:22:22:22:2b	D	ge-0/0/0.0		
22:22:22:22:22:2c	D	ge-0/0/0.0		
22:22:22:22:22:2d	D	ge-0/0/0.0		
22:22:22:22:22:2e	D	ge-0/0/0.0		
22:22:22:22:22:2f	D	ge-0/0/0.0		
22:22:22:22:22:30	D	ge-0/0/0.0		
22:22:22:22:22:31	D	ge-0/0/0.0		
c0:01:11:25:f1:00	D	ge-0/0/0.0		
c0:02:11:34:f1:00	DC		1048578	1048578

Figura 84. Tabla EVPN MAC

EVPN opera sobre el plano de control es decir realiza el aprendizaje de MAC usando MP-BGP. MP-BGP anuncia rutas de tipo 2, es decir, rutas de host MAC / IP que representan cada cliente. Se observa que cada vez que el dispositivo PE recibe una nueva ruta MAC, la dirección MAC ya se encuentra instalada en la tabla MAC. La próxima vez que el dispositivo del cliente envíe la solicitud ARP para esta dirección MAC, este requerimiento no inunda en la red central o llega los otros, Por lo que el enrutador de borde PE funciona como un proxy y responde a la solicitud ARP con su propia dirección MAC. Este es el tráfico BUM suprimido en una red EVPN.

4.5.3.2.VPLS

En la primera inundación de tráfico ARP se obtienen los siguientes resultados:

No.	Time	Source	Destination	Protocol	Length	Info
12	22:09:12,255782	22:22:22:22:22:23	Private_11:11:12	ARP	64	Who has 192.168.1.21? Tell 192.168.1.10
13	22:09:13,256528	22:22:22:22:22:24	Private_11:11:13	ARP	64	Who has 192.168.1.22? Tell 192.168.1.10
14	22:09:14,256939	22:22:22:22:22:25	Private_11:11:14	ARP	64	Who has 192.168.1.23? Tell 192.168.1.10
18	22:09:15,257580	22:22:22:22:22:26	Private_11:11:15	ARP	64	Who has 192.168.1.24? Tell 192.168.1.10
19	22:09:16,258225	22:22:22:22:22:27	Private_11:11:16	ARP	64	Who has 192.168.1.25? Tell 192.168.1.10
20	22:09:17,258590	22:22:22:22:22:28	Private_11:11:17	ARP	64	Who has 192.168.1.26? Tell 192.168.1.10
24	22:09:18,259676	22:22:22:22:22:29	Private_11:11:18	ARP	64	Who has 192.168.1.27? Tell 192.168.1.10
25	22:09:19,261394	22:22:22:22:22:2a	Private_11:11:19	ARP	64	Who has 192.168.1.28? Tell 192.168.1.10
26	22:09:20,260848	22:22:22:22:22:2b	Private_11:11:1a	ARP	64	Who has 192.168.1.29? Tell 192.168.1.10
30	22:09:21,260950	22:22:22:22:22:2c	Private_11:11:1b	ARP	64	Who has 192.168.1.30? Tell 192.168.1.10
31	22:09:22,261305	22:22:22:22:22:2d	Private_11:11:1c	ARP	64	Who has 192.168.1.31? Tell 192.168.1.10
32	22:09:23,262087	22:22:22:22:22:2e	Private_11:11:1d	ARP	64	Who has 192.168.1.32? Tell 192.168.1.10
36	22:09:24,263207	22:22:22:22:22:2f	Private_11:11:1e	ARP	64	Who has 192.168.1.33? Tell 192.168.1.10
37	22:09:25,263892	22:22:22:22:22:30	Private_11:11:1f	ARP	64	Who has 192.168.1.34? Tell 192.168.1.10
38	22:09:26,264588	22:22:22:22:22:31	Private_11:11:20	ARP	64	Who has 192.168.1.35? Tell 192.168.1.10
42	22:09:27,267617	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.1.20? Tell 192.168.1.10
43	22:09:28,267832	22:22:22:22:22:23	Private_11:11:12	ARP	64	Who has 192.168.1.21? Tell 192.168.1.10
44	22:09:29,268286	22:22:22:22:22:24	Private_11:11:13	ARP	64	Who has 192.168.1.22? Tell 192.168.1.10
48	22:09:30,268788	22:22:22:22:22:25	Private_11:11:14	ARP	64	Who has 192.168.1.23? Tell 192.168.1.10
49	22:09:31,270171	22:22:22:22:22:26	Private_11:11:15	ARP	64	Who has 192.168.1.24? Tell 192.168.1.10
50	22:09:32,270599	22:22:22:22:22:27	Private_11:11:16	ARP	64	Who has 192.168.1.25? Tell 192.168.1.10
51	22:09:33,270502	22:22:22:22:22:28	Private_11:11:17	ARP	64	Who has 192.168.1.26? Tell 192.168.1.10
55	22:09:34,271176	22:22:22:22:22:29	Private_11:11:18	ARP	64	Who has 192.168.1.27? Tell 192.168.1.10
56	22:09:35,271635	22:22:22:22:22:2a	Private_11:11:19	ARP	64	Who has 192.168.1.28? Tell 192.168.1.10
57	22:09:36,272242	22:22:22:22:22:2b	Private_11:11:1a	ARP	64	Who has 192.168.1.29? Tell 192.168.1.10
61	22:09:37,273104	22:22:22:22:22:2c	Private_11:11:1b	ARP	64	Who has 192.168.1.30? Tell 192.168.1.10
62	22:09:38,272854	22:22:22:22:22:2d	Private_11:11:1c	ARP	64	Who has 192.168.1.31? Tell 192.168.1.10
63	22:09:39,273885	22:22:22:22:22:2e	Private_11:11:1d	ARP	64	Who has 192.168.1.32? Tell 192.168.1.10

Figura 85. Inundación ARP -antes-.

En la segunda inundación se obtienen los siguientes resultados:

No.	Time	Source	Destination	Protocol	Length	Info
105	22:10:00,288249	22:22:22:22:22:23	Private_11:11:12	ARP	64	Who has 192.168.1.21? Tell 192.168.1.10
106	22:10:01,290432	22:22:22:22:22:24	Private_11:11:13	ARP	64	Who has 192.168.1.22? Tell 192.168.1.10
110	22:10:02,289105	22:22:22:22:22:25	Private_11:11:14	ARP	64	Who has 192.168.1.23? Tell 192.168.1.10
111	22:10:03,289568	22:22:22:22:22:26	Private_11:11:15	ARP	64	Who has 192.168.1.24? Tell 192.168.1.10
112	22:10:04,290547	22:22:22:22:22:27	Private_11:11:16	ARP	64	Who has 192.168.1.25? Tell 192.168.1.10
116	22:10:05,290165	22:22:22:22:22:28	Private_11:11:17	ARP	64	Who has 192.168.1.26? Tell 192.168.1.10
117	22:10:06,291701	22:22:22:22:22:29	Private_11:11:18	ARP	64	Who has 192.168.1.27? Tell 192.168.1.10
118	22:10:07,290822	22:22:22:22:22:2a	Private_11:11:19	ARP	64	Who has 192.168.1.28? Tell 192.168.1.10
122	22:10:08,291926	22:22:22:22:22:2b	Private_11:11:1a	ARP	64	Who has 192.168.1.29? Tell 192.168.1.10
123	22:10:09,291641	22:22:22:22:22:2c	Private_11:11:1b	ARP	64	Who has 192.168.1.30? Tell 192.168.1.10
124	22:10:10,292278	22:22:22:22:22:2d	Private_11:11:1c	ARP	64	Who has 192.168.1.31? Tell 192.168.1.10
128	22:10:11,294890	22:22:22:22:22:2e	Private_11:11:1d	ARP	64	Who has 192.168.1.32? Tell 192.168.1.10
129	22:10:12,293686	22:22:22:22:22:2f	Private_11:11:1e	ARP	64	Who has 192.168.1.33? Tell 192.168.1.10
130	22:10:13,294407	22:22:22:22:22:30	Private_11:11:1f	ARP	64	Who has 192.168.1.34? Tell 192.168.1.10
134	22:10:14,294383	22:22:22:22:22:31	Private_11:11:20	ARP	64	Who has 192.168.1.35? Tell 192.168.1.10
135	22:10:15,296448	22:22:22:22:22:22	Private_11:11:11	ARP	64	Who has 192.168.1.20? Tell 192.168.1.10
136	22:10:16,296878	22:22:22:22:22:23	Private_11:11:12	ARP	64	Who has 192.168.1.21? Tell 192.168.1.10
140	22:10:17,298035	22:22:22:22:22:24	Private_11:11:13	ARP	64	Who has 192.168.1.22? Tell 192.168.1.10
141	22:10:18,297729	22:22:22:22:22:25	Private_11:11:14	ARP	64	Who has 192.168.1.23? Tell 192.168.1.10
142	22:10:19,297669	22:22:22:22:22:26	Private_11:11:15	ARP	64	Who has 192.168.1.24? Tell 192.168.1.10
146	22:10:20,298097	22:22:22:22:22:27	Private_11:11:16	ARP	64	Who has 192.168.1.25? Tell 192.168.1.10
147	22:10:21,298436	22:22:22:22:22:28	Private_11:11:17	ARP	64	Who has 192.168.1.26? Tell 192.168.1.10
148	22:10:22,298967	22:22:22:22:22:29	Private_11:11:18	ARP	64	Who has 192.168.1.27? Tell 192.168.1.10
152	22:10:23,299296	22:22:22:22:22:2a	Private_11:11:19	ARP	64	Who has 192.168.1.28? Tell 192.168.1.10
153	22:10:24,300321	22:22:22:22:22:2b	Private_11:11:1a	ARP	64	Who has 192.168.1.29? Tell 192.168.1.10
154	22:10:25,300989	22:22:22:22:22:2c	Private_11:11:1b	ARP	64	Who has 192.168.1.30? Tell 192.168.1.10
158	22:10:26,301854	22:22:22:22:22:2d	Private_11:11:1c	ARP	64	Who has 192.168.1.31? Tell 192.168.1.10
159	22:10:27,302184	22:22:22:22:22:2e	Private_11:11:1d	ARP	64	Who has 192.168.1.32? Tell 192.168.1.10

Figura 86. Inundación ARP-después-.

```

root@R1-PE1> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
          SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls
Bridging domain : __vpls__, VLAN : NA
  MAC          MAC          Logical          NH    RTR
  address      flags      interface      Index  ID
  22:22:22:22:22:22  D      ge-0/0/0.10
  22:22:22:22:22:23  D      ge-0/0/0.10
  22:22:22:22:22:24  D      ge-0/0/0.10
  22:22:22:22:22:25  D      ge-0/0/0.10
  22:22:22:22:22:26  D      ge-0/0/0.10
  22:22:22:22:22:27  D      ge-0/0/0.10
  22:22:22:22:22:28  D      ge-0/0/0.10
  22:22:22:22:22:29  D      ge-0/0/0.10
  22:22:22:22:22:2a  D      ge-0/0/0.10
  22:22:22:22:22:2b  D      ge-0/0/0.10
  22:22:22:22:22:2c  D      ge-0/0/0.10
  22:22:22:22:22:2d  D      ge-0/0/0.10
  22:22:22:22:22:2e  D      ge-0/0/0.10
  22:22:22:22:22:2f  D      ge-0/0/0.10
  22:22:22:22:22:30  D      ge-0/0/0.10
  22:22:22:22:22:31  D      ge-0/0/0.10
  c0:01:08:c9:f1:00  D      ge-0/0/0.10

```

Figura 87. Tabla VPLS MAC

VPLS al ser una tecnología que opera en el plano de datos, tiene como medio de aprendizaje a los tipos de despacho convencionales como son Broadcast, Unicast, Multicast hacia los clientes, por lo que el tráfico BUM que transita por el Core de la red es inevitable.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Las herramientas usadas en esta investigación tanto para la simulación y análisis de la topología de red en la comparativa de las tecnologías de VPN's, aportan una confiable apreciación de los resultados obtenidos, esto gracias al ambiente amigable que prestan dichos programas y la versatilidad en el uso/configuración del mismo para la simulación aproximada de una red real. El uso de las herramientas de simulación (GNS3) y motor virtualización (VMware), brindan un potencial de rendimiento de hardware importante para la implementación de una red mixta en cuanto a proveedores de equipos de networking, esto presta una sensación de manejar un ambiente real en el entorno de configuración y pruebas.
- Todas las pruebas realizadas se efectuaron en base a una topología de red sobre la que es posible crear las instancias necesarias para implementar las tecnologías de VPN's bajo las mismas condiciones y de ese modo tener una apreciación de resultados confiable.
- Juniper y Cisco prestan con el soporte en documentación referente al CLI (interfaz de línea de comandos) en sus sistemas operativos JUNOS e IOS respectivamente, por lo que no resulta confuso el aprendizaje de los comandos para la configuración de los dispositivos de red.

- A nivel de rendimiento se verificó que el tiempo total de convergencia de red requerido en el caso de VPLS es de aproximadamente 45.6 segundos en promedio, mientras que EVPN tarda aproximadamente 7.6 segundos en promedio para que la red converja, mostrados en la tabla 8. Considerando que la red escogida para el escenario de pruebas comprende en el Core de apenas 3 enrutadores los tiempos de convergencia son significativos en un indicador de 6 a 1. Por tanto, en ambientes tales como los tendría un proveedor de servicios de internet (ISP) se deben tomar en cuenta que el número de enrutadores tiene gran importancia dado que el tiempo de tolerancia a fallos debe ser bajo. Otro aspecto evaluado en esta investigación es la capacidad que tiene EVPN frente a VPLS en cuanto a la movilidad MAC, la representación en esta investigación fue la de un cliente con una computadora moviéndose de un Router de borde hacia otro, pero si esto se aplica en un centro de datos, implicaría la capacidad de tener redundancia en una eventualidad de fallo sin pérdida de paquetes, y tiempo en la actualización de sus tablas de alcanzabilidad.
- Se pudo validar la diferencia que existe en relación al manejo del tráfico BUM en el Core de la red, verificando que, gracias al aprendizaje MAC de EVPN a través de la señalización BGP sin la necesidad de usar el plano de datos, se puede evitar la inundación de tráfico innecesario (ARP), creando un Proxy ARP ficticio y de este modo mejorar el rendimiento, de este modo una red de datos utilizando EVPN reducirá el consumo de ancho de banda, en relación al volumen de tráfico BUM generado por la cantidad de enrutadores, en comparación con VPLS. EVPN también tiene la capacidad de admitir una arquitectura con un plano de control centralizado, esto debido a que trabaja sobre MP-BGP.
- En cuanto a funcionalidad en el campo de VPN's, las dos tecnologías ofrecen el mismo objetivo referente a la conectividad de dos o más clientes ubicados geográficamente en

distintos lugares. Sin embargo, haciendo referencia a los parámetros establecidos para la comparación de dichas tecnologías; EVPN presenta mejor respuesta a fallos y tienen mejor rendimiento en la utilización del ancho de banda.

5.2 Recomendaciones

- En base a la comparativa realizada en esta investigación sobre las tecnologías de VPN's, se recomienda la implementación de EVPN, ya que aumenta la eficiencia de utilización del enlace y proporciona soporte para entornos virtualizados.
- Se recomienda para futuros estudios sobre tecnologías de networking, el uso del simulador GNS3, con el motor de virtualización VMware sobre una red donde la instancia GSN3VM tenga autonomía en cuanto a hardware, con esto se podrían realizar escenarios de pruebas con un número considerable de enrutadores, conmutadores, terminales finales, etc. Y de este modo profundizar el aprendizaje en el área de Redes y Comunicación de Datos.

BIBLIOGRAFÍA

Andrew, G. (2007). Cisco Secute Virtual Private Network. Cisco Press.

Anónimo. (2014). Cómo funciona MPLS. Obtenido de Cómo funciona MPLS:
<https://www.networkworld.es/telecomunicaciones/como-funciona-mpls>

Anónimo. (s.f.). QEMU. Obtenido de <https://es.wikipedia.org/wiki/QEMU>

Anónimo. (s.f.). VMWare. Obtenido de <https://es.wikipedia.org/wiki/VMware>

Anónimo. (s.f.). Wireshark. Obtenido de <https://es.wikipedia.org/wiki/Wireshark>

Auben Networks. (s.f.). MPLS Layer 2 VPN / Layer 2 Circuit. Obtenido de
<http://www.auben.net/index.php/tecnologias/g-mpls-e-ingenieria-de-trafico/mpls-layer-2-vpn-layer-2-circuit>

Cisco. (s.f.). Cisco Overview. Obtenido de <https://www.cisco.com/>

Cosoi, E. (s.f.). Redes PRivadas Virtuales: Un mundo de acceso a las librerías del mundo. Obtenido de https://scielo.conicyt.cl/scielo.php?pid=S0370-41062005000200014&script=sci_arttext

Garcia, A. (2002). MPLS - Multiprotocol Label Switching V Foro Tecnológico @asLAN: Banda Ancha y su Entorno.

GNS3. (s.f.). Releaso Notes. Obtenido de <https://docs.gns3.com/release.html>

Icaran, M. (s.f.). Estudio y configuración de una VPN MPLS MP-BGP. USM España.

Juniper. (s.f.). Junos OS Overview. Obtenido de <https://www.juniper.net/>

Lesserre, M., & Kompella, V. (2007). Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling. IETF RFC 4762.

Rosen, E., & Rakhter, Y. (2006). BGP/MPLS IP Virtual Private Networks (VPN's). IETF RFC 4364.

Sajassi, A., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., & Heendrickx, W. (2015). BGP MPLS-Based Ethernet VPN. IETF RFC 7432.

Sánchez, A., & Grzegorz, K. (2015). MPLS in the SDN Era. O'Reilly.

Scalability testing of legacy MPLS-based Virtual Private Networks. (s.f.). Obtenido de https://www.researchgate.net/publication/271551766_BGP_performance_analysis_for_large_scale_VPN

Shingledecker, R. (s.f.). Tiny Core Linux. Obtenido de <https://distro.ibiblio.org/tinycorelinux/>

Srivats, P. (s.f.). Ostinato Network Traffic Generator and Analyzer. Obtenido de <https://ostinato.org/>

Tanenbaun, A. (2011). Redes de Computadoras. 4ta. Edición.

Townsley, W., Valencia, A., Rubens, A., & Pall, G. (1999). Layer Two Tunneling Protocol L2TP. IETF RFC 2661.