Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

DISEÑO Y SIMULACIÓN DE SERVICIOS L3VPN Y L2VPN ENFOCADA PARA EMPRESAS PRIVADAS DENTRO DE UNA RED MPLS

Jhonnser Jehonadab Escobar Lucero
Asesorado por el Ing. Julio César Solares Peñate

Guatemala, febrero de 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DISEÑO Y SIMULACIÓN DE SERVICIOS L3VPN Y L2VPN ENFOCADA PARA EMPRESAS PRIVADAS DENTRO DE UNA RED MPLS

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JHONNSER JEHONADAB ESCOBAR LUCERO
ASESORADO POR EL ING. JULIO CÉSAR SOLARES PEÑATE

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, FEBRERO DE 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. José Francisco Gómez Rivera (a.i.)
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Ing. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Ing. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. José Aníbal Silva de los Angeles
EXAMINADOR	Ing. Helmunt Federico Chicol Cabrera
EXAMINADOR	Ing. Marvin Marino Hernández Fernández
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO Y SIMULACIÓN DE SERVICIOS L3VPN Y L2VPN ENFOCADA PARA EMPRESAS PRIVADAS DENTRO DE UNA RED MPLS

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería de Mecánica Eléctrica, con fecha 4 de mayo de 2021.

Jhonnser Jehonadab Escobar Lucero

Guatemala, 27 de febrero de 2023

Señor
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Estimado Ingeniero:

Por este medio me permito dar aprobación al trabajo de Graduación titulado **DISEÑO Y SIMULACIÓN DE SERVICIOS L3VPN Y L2VPN ENFOCADA PARA EMPRESAS PRIVADAS DENTRO DE UNA RED MPLS**, desarrollado por el estudiante **Jhonnser Jehonadab Escobar Lucero**, ya que considero que cumple con los requisitos establecidos.

Por lo tanto, el autor de este trabajo y yo como asesor, nos hacemos responsables del contenido y conclusiones del mismo.

Sin otro en particular, aprovecho la oportunidad para saludarlo.

ID Y ENSEÑAD A TODOS

Ing. Julio César Solares Peñate
Asesor

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

Guatemala, 1 de marzo de 2023

Señor director
Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Señor director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **DISEÑO Y SIMULACIÓN DE SERVICIOS L3VPN Y L2VPN ENFOCADA PARA EMPRESAS PRIVADAS DENTRO DE UNA RED MPLS,** desarrollado por el estudiante **Jhonnser Jehonadab Escobar Lucero**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS

Ing. Julio César Solares Peñate
Coordinador de Electrónica

ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

REF. EIME 53.2023.

El director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del asesor, con el Visto Bueno del coordinador de área, al trabajo de graduación del estudiante Jhonnser Jehonadab Escobar Lucero: "**DISEÑO Y SIMULACIÓN DE SERVICIOS L3VPN Y L2VPN ENFOCADA PARA EMPRESAS PRIVADAS DENTRO DE UNA RED MPLS**", procede a la autorización correspondiente.

Ing. Armando Alonso Rivera Carrillo

Guatemala, 3 de octubre del 2023.

Decanato
Facultad de Ingeniería
24189101- 24189102
secretariadecanato@ingenieria.usac.edu.gt

LNG.DECANATO.OI.065.2024

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO Y SIMULACIÓN DE SERVICIOS L3VPN Y L2VPN ENFOCADA PARA EMPRESAS PRIVADAS DENTRO DE UNA RED MPLS,** presentado por: **Jhonnser Jehonadab Escobar Lucero,** después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. José Francisco Gómez Rivera
Decano a.i.

Guatemala, febrero de 2024

JFGR/gaoc

ACTO QUE DEDICO A:

Mi madre	Deysi Lucero, por ser un excelente ejemplo de amor, sacrificio y esfuerzo. Gracias por brindarme tu apoyo incondicional
Mi padre	Romeo Escobar (q. e. p. d.), porque tu presencia siempre ha estado conmigo en todo momento, fuiste uno de los regalos más grandes de mi vida.
Mis hermanos	Emilsa, Enmilly y Osman Escobar, por aportarme grandes momentos de felicidad y demostrarme lo grandioso es que tener hermanos para compartir y aprender juntos.
Mi familia	Especialmente a mis tíos Israel y Yony Escobar, por su apoyo incondicional. A mi abuelo Rafael Escobar (q. e. p. d.), tías, primos y primas por su amor y cariño.
Mis compañeros	Por haber compartido tantas experiencias inolvidables, de corazón agradezco su amistad y apoyo. Gracias por ayudarme cuando los necesité.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi casa de estudios y por brindarme las herramientas para convertirme en un profesional.
Facultad de Ingeniería	Por ayudarme a mi formación profesional.
Los docentes	Infinitas gracias por ayudarme en mi formación profesional, por su vocación, docencia y paciencia al compartirme sus conocimientos y prepararme integralmente como profesional.
Mi asesor de tesis	Ing. Julio César Solares Peñate, por su asesoría y tiempo dedicado a este trabajo.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
$	Dólar
Gb	Giga Bite
Mb	Mega Bite

GLOSARIO

Adyacencia	Se forma cuando dos enrutadores vecinos intercambian información de enrutamiento sincronizando sus tablas.
Ancho de banda	Es la medida de datos y recursos de comunicación disponible expresados en bit/s.
ARP	Protocolo que proporciona asignación dinámica entre las direcciones IP y las direcciones MAC.
Backbone	Término que se utiliza para referirse a la parte central o principal de una red ISP.
Bit	Es la unidad mínima de información.
Dirección MAC	Una dirección exclusiva que se asigna a una interfaz de red.
Encabezado IP	Datos que identifican de forma exclusiva un paquete de Internet, contiene dirección de origen y destino.
Encapsular	Proceso en el cual el enrutador agrega etiquetas a los diferentes tipos de paquetes.
Enlace	Conexión a través de la cual se transmiten datos.

Enlace troncal	Es un enlace punto a punto de alta capacidad donde converge el tráfico IP de distintos clientes.
Enrutador	Un sistema que tiene más de una interfaz, ejecuta protocolos de enrutamiento y reenvía paquetes de datos entre las redes de los equipos.
Enrutamiento	Acción de encontrar la mejor ruta entre todas las posibles.
Ethernet	Es el estándar que permite conectar equipos de red mediante un cable.
IANA	Una organización que delega las direcciones IP registradas a los registros de Internet de todo el mundo.
ICMP	Protocolo que comprueba si existe comunicación bidireccional entre dos puntos.
MTU	El tamaño de la unidad de datos más grande, que puede transmitirse a través de un enlace.
Multidifusión	Un procedimiento de capa de red que se utiliza para enviar paquetes de datagramas en varios equipos en una red IP.
Nodo	Punto de intersección, unión o terminación de los distintos elementos que componen la red.

Paquete	Nombre dado a los bloques en los cuales es dividida la información antes de su envío.
Tabla de enrutamiento	Base de datos donde los enrutadores almacenan las mejores rutas.
Topología	Disposición física en la que se conectan los dispositivos a la red.
UDP	Un protocolo que utiliza un equipo para enviar datagramas a otros equipos de una red IP.
Vecindades	Relación establecida entre dispositivos que comparten un mismo segmento de la red.
VLAN	Técnica utilizada para dividir segmentos de red independientes dentro de una topología física.

RESUMEN

En el presente trabajo de graduación consta de siete capítulos donde se expresan los conceptos teóricos y prácticos para comprender el funcionamiento de los servicios L3VPN y L2VPN dentro de una red MPLS, donde la simulación permite a los administradores de red evaluar y validar diferentes configuraciones antes de implementarlas en los entornos reales.

En el primer capítulo se presenta todos los conceptos básicos sobre los protocolos de enrutamiento de estado de enlace y vector distancia más utilizados por los proveedores de red para brindar conectividad hacia distintas redes.

El segundo capítulo se presenta toda la terminología y los protocolos relacionados con el etiquetado MPLS que garantizan soluciones eficientes para el manejo de grandes cantidades de tráfico.

El tercer y cuarto capítulo se presentan los conceptos de las tecnologías L3VPN y L2VPN donde se detallan con base de ejemplos la forma correcta de configuración y los entornos donde se pueden aplicar para aumentar el rendimiento de la red corporativa.

Los capítulos restantes se realiza el diseño y simulación por software utilizando EVE-NG como programa de simulación que permite utilizar el sistema operativo de los equipos reales de Cisco System XE y XR para implementarlos sobre una topología MPLS para luego ejecutar las tecnologías L3VPN y L2VPN con los requerimientos más comunes de configuración.

OBJETIVOS

General

Diseñar y simular una red corporativa de servicios multiprotocolo donde incluya las instrucciones para la correcta configuración de las tecnologías de L3VPN3 y L2VPN sobre una red MPLS.

Específicos

1. Definir los conceptos teóricos sobre MPLS entendiendo su funcionamiento y la ventaja que conlleva la implementación dentro de las redes IP.

2. Presentar las características más importantes sobre los servicios de L3VPN Y L2VPN los cuales brinden seguridad al cliente que obtendrá un servicio de alto rendimiento de forma privada en la transmisión de su información.

3. Desarrollar una red corporativa a través de software que permita simular escenarios reales para la correcta configuración sobre los equipos de distribución de una red ISP.

INTRODUCCIÓN

El crecimiento de las redes corporativas surge a través de la necesidad de poder transmitir grandes cantidades de información en el menor tiempo posible desde la transmisión de pocos bits por segundo hasta alcanzar tasas alrededor de gigabits por segundo, para que este incremento de transferencia sea posible se ha creado una serie de protocolos estandarizados a lo largo del tiempo que permita obtener mejores resultados, tanto en la confiabilidad y estabilidad de los paquetes enviados; las redes corporativas permiten conectar todos los puntos remotos o sucursales de forma permanente, privada y segura a través de fibra óptica y mediante tecnología MPLS.

Los proveedores de servicios (ISP), son los responsables de transmitir la información requerida por los clientes en distintos puntos geográficos confiando que la transmisión no sufra degradaciones que puedan causar latencia o perdida de información en todo su trayecto.

Una de las soluciones que ha permitido a los proveedores de servicios mejorar el rendimiento de la red, es gracias a la implementación del servicio Multiprotocol Label Switching como son sus siglas MPLS, este es un estándar creado por la organización (IETF), que consiste en crear un método mucho más eficiente a la transmisión de paquetes dentro de una red que puede fácilmente integrar información relacionado con internet, voz y video.

Con la red MPLS integrada los proveedores de servicios ofrecen soluciones a clientes acorde a los requerimientos corporativos que ellos soliciten siendo los servicios más utilizados las VPN de capa 3 (L3VPN) y capa 2 (L2VPN),

son una forma de realizar túneles los cuales transportan tráfico de múltiples clientes a través de sus redes y donde es necesario mantener los protocolos de ambas capas separando el tráfico de otros clientes o del mismo proveedor de servicio.

1. FUNDAMENTOS DE RED

El concepto de red en el área de las telecomunicaciones se puede entender como una interconexión de dispositivos que se comunican entre si para compartir información y acceder a recursos compartidos, estos dispositivos pueden incluir computadoras, servidores, enrutadores, impresoras entre otros.

La conexión entre los equipos se basa en diversos protocolos y estándares que facilita la comunicación en la transmisión de datos y garantiza la entrega de la información de forma confiable entre los dispositivos conectados.

1.1. Modelo OSI

El modelo OSI, es el estándar ampliamente utilizado para poder explicar el proceso que deben de pasar los datos para que puedan ser enviados de una red a otra, esto se define mediante siete capas las cuales son las siguientes.

- Capa de aplicación: es la capa más alta del modelo OSI y es la responsable de interactuar directamente con las aplicaciones del usuario final, en esta capa se manejan los distintos protocolos como pueden ser el HTTP, SMTP, FTP entre otros.

- Capa 6 – Capa de presentación: una de las tareas de esta capa es brindar el formato adecuado a los datos para asegurar la comunicación entre los equipos de origen y destino.

- Capa 5 – Capa de sesión: en la capa se lleva a cabo las solicitudes de gestión, mantenimiento y finalización de la comunicación entre dos dispositivos.

- Capa 4 – Capa de transporte: la capa de transporte de encarga de la selección del protocolo del transporte, control de errores y control de flujo que garantiza una comunicación ordenada y sólida entre los sistemas finales.

- Capa 3 – Capa de red: es la capa encargada del enrutamiento del tráfico IP y determina el mejor camino tomando en cuenta diversos factores como ancho de banda, costo, numero de saltos entre otros. Adicional utiliza direcciones IP para identificar a todos los dispositivos conectados en la red.

- Capa 2 – Capa de enlace de datos: esta capa se encarga de proporcionar una comunicación confiable dentro de una red de área local, identificando todos los dispositivos de red conectados y utiliza la dirección MAC para enviar y entregar las tramas correspondientes del mismo dominio.

- Capa 1 – Capa física: se encarga de la envío y recepción de los datos a través de un medio de transmisión como lo puede ser hilo de cobre, fibra óptica, entre otros.

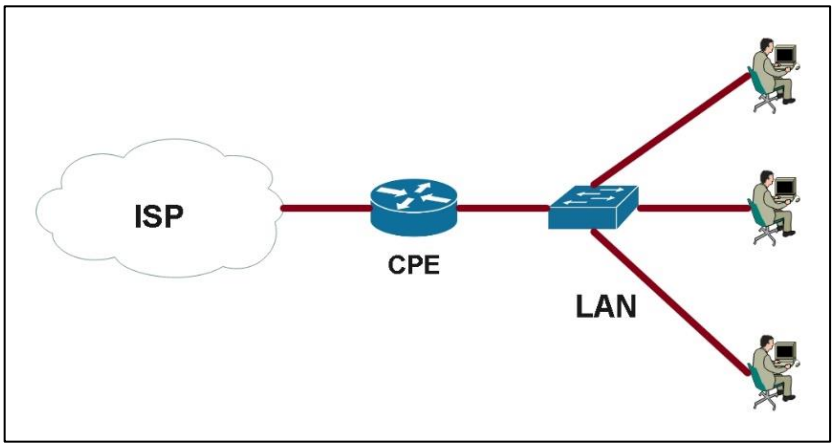
1.2. Clasificación de redes

Las redes se pueden clasificar de diferentes maneras basadas en su tamaño, usuarios conectados y alcance geográfico. Los tipos de redes más comunes y las que tienen relación con este estudio son las siguientes:

- Red de área local (LAN): es la infraestructura de red que es administrada por alguna entidad pública o privada conformada por un conjunto de dispositivos donde se encuentran bajo el mismo dominio de broadcast. Todos los dispositivos se encuentran directamente conectados por cables ethernet o de manera inalámbrica utilizando wifi.

Figura 1.

Red LAN

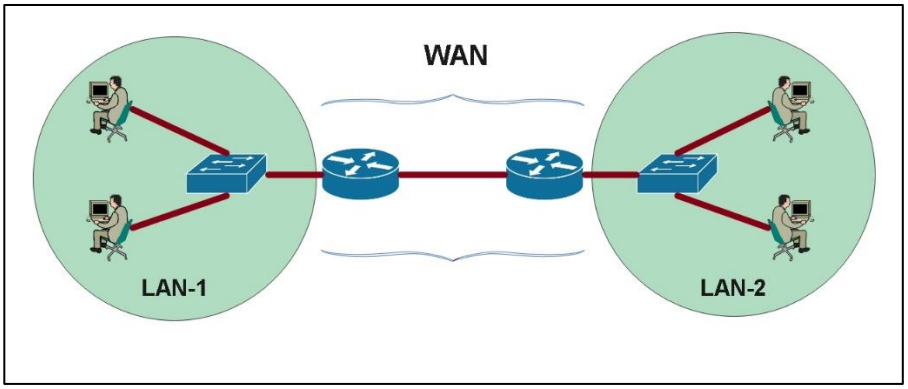


Nota. Ejemplo de red LAN. Elaboración propia, realizado con Edraw Max.

- Red de área amplia (WAN): es la infraestructura de red que proporciona acceso a otras redes LAN en un área geográfica extensa. Normalmente, la administración de las redes WAN está a cargo de proveedores de servicios (ISP).

Figura 2.

Red WAN

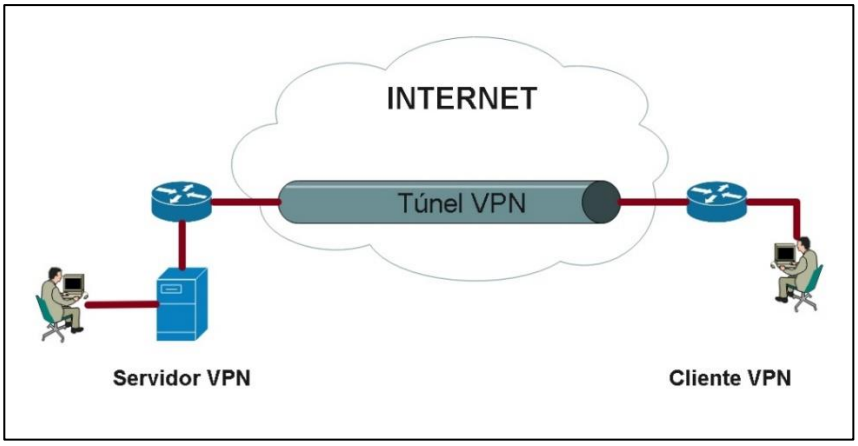


Nota. Ejemplo de red WAN. Elaboración propia, realizado con Edraw Max.

- Red privada virtual (VPN): es una conexión segura con niveles de encriptación entre redes o dispositivos a través de una red pública.

Figura 3.

Túnel VPN



Nota. Ejemplo de túnel VPN. Elaboración propia, realizado con Edraw Max.

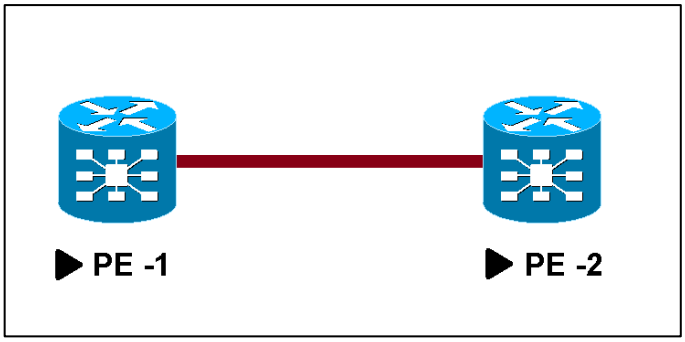
1.3. Topologías físicas de WAN

Por lo general, las WAN se interconectan mediante las siguientes topologías físicas:

- Punto a punto: las topologías físicas punto a punto conectan dos sitios directamente, por lo que facilita el envío de la información de los protocolos de capa 2 y capa 3, dado que todas las tramas solo pueden transferirse entre los dos nodos.

Figura 4.

Red punto a punto

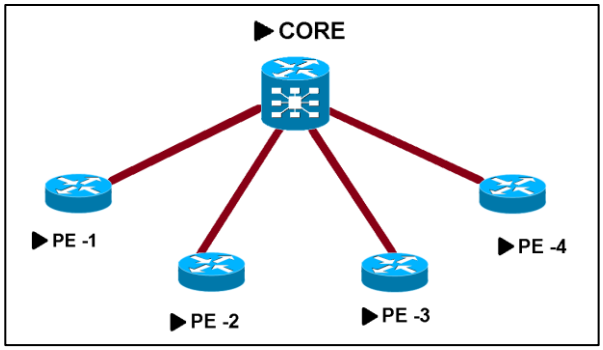


Nota. Ejemplo de red de punto a punto. Elaboración propia, realizado con Edraw Max.

- Hub-and-spoke: es el tipo de topología de red que utiliza varias redes WAN para conectar múltiples ubicaciones a través de una red central, este tipo de red es útil cuando las empresas cuentan con una oficina principal y varias sucursales lo que permite una administración centralizada y procesos de enrutamiento mejor organizados.

Figura 5.

Red Hub and Spoke

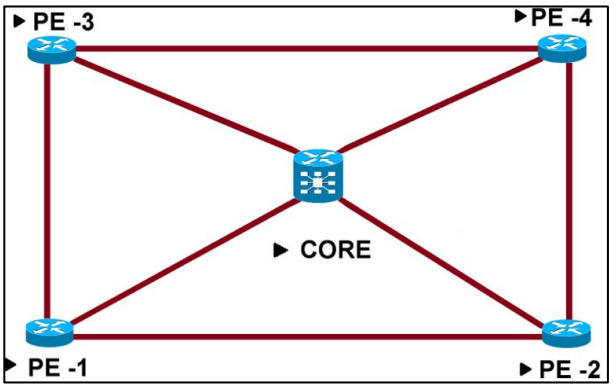


Nota. Ejemplo de red Hub and Spoke. Elaboración propia, realizado con Edraw Max.

- Malla: es la topología de red en donde cada dispositivo se encuentra directamente conectado a todos los demás presentes en la red, en una malla completa ofrece múltiples caminos de comunicación ofreciendo un alto nivel de redundancia en casos de fallas entre nodos.

Figura 6.

Red en malla



Nota. Ejemplo de red de malla. Elaboración propia, realizado con Edraw Max.

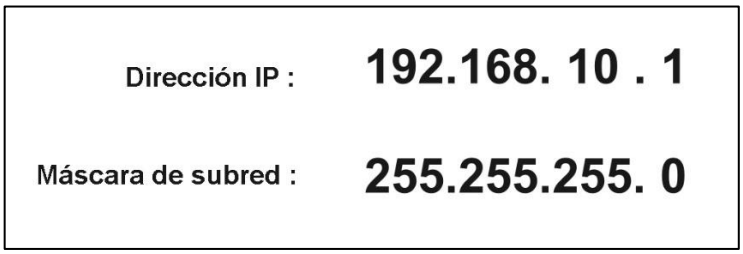
1.4. Direccionamiento IPv4

El direccionamiento IP versión 4 es una representación binaria que identifica a un equipo en una red, el cual está integrada por dos partes la dirección IP y la máscara de subred ambos parámetros deben de configurarse en los equipos que se deben de identificar, el primer parámetro identifica de forma única el dispositivo o interfaz y la máscara de red indica que dentro que porción de red estará configurada.

Una dirección IPv4 está formada mediante un número binario de 32 bits o cuatro octetos de 8 bits donde cada valor decimal puede representarse entre 0 a 255 como se muestra a continuación.

Figura 7.

Representación de una dirección IP



Nota. Ejemplo de dirección IP. Elaboración propia, realizado con Edraw Max.

1.5. Enrutamiento IP

Es el proceso en el cual los datos se transfieren de una red a otra, a través de la ruta más óptima entre una alta disposición de rutas hacia un destino. Todas las rutas hacia los diferentes destinos se registran en la base de datos del equipo router llamadas routing information base (RIB) o tablas de enrutamiento, esta

base de datos es un conjunto de reglas que sirven para determinar el camino que deben de seguir los paquetes al momento de cruzar varias redes de por medio.

Las tablas de enrutamiento se llenan con varios campos como pueden ser la red del destino, mascara de subred, dirección IP del siguiente salto, interfaz de salida y la métrica, este último criterio es de los más importantes a la hora de tener un destino con múltiples caminos de llegada, este consiste en otorgar una prioridad en base a varios parámetros de selección como lo pueden ser número de saltos, costo, ancho de banda y latencia.

En las tablas de enrutamiento las rutas se almacenan de tres formas diferentes, las primeras son las redes conectadas directamente, estas se agregan de forma automática cuando se agrega un direccionamiento IP dentro de alguna interfaz del equipo, también se encuentra el enrutamiento estático, donde el administrador de red añade o quita una o varias rutas de forma manual y por último el enrutamiento dinámico en este caso los equipos actualizan su base de datos de forma automática, lo hacen mediante protocolos para intercambiar información sobre las topologías de red, además permiten que los dispositivos puedan informar a los demás equipos cuando se detecte cambios dentro de la red para actualizar la base de datos de todos los equipos .

1.6. Enrutamiento estático

Es el tipo enrutamiento donde las rutas se configuran de forma manual otorgando un mayor control sobre los paquetes que se transmiten a través de la red, es ampliamente utilizadas en redes pequeñas y para establecer la conectividad con los proveedores de servicios, además ofrece ventajas y desventajas al entorno donde se requiera utilizar, por ejemplo las rutas configuradas de forma estática no se anuncian a través de la red por lo que

aumenta la seguridad de la información y es más eficiente al momento de gestionar los recursos, porque no requiere un elevado procesamiento del CPU y de memoria al momento de calcular por donde enviar los paquetes.

Este tipo de enrutamiento son útiles en redes con solo una ruta hacia una red externa proporcionando un Gateway predeterminado para tráficos salientes de una red LAN, en donde se necesite reducir el número de rutas anunciadas mediante una sola ruta resumen, además se vuelve conveniente utilizar una ruta de respaldo en caso de que falle un enlace de la ruta principal, esto se realiza configurando una ruta estática apuntando hacia otro proveedor de servicios o algún enlace dedicado para realizar la función de redundancia.

En redes muy grandes no se recomienda implementar rutas estáticas ya que, si un enlace falla, una ruta estática no tiene el comportamiento necesario para buscar un nuevo camino para volver a enrutar el tráfico por lo que podría ser una desventaja en ciertos tipos de escenarios.

La sintaxis para configurar una ruta estática es la siguiente.

Figura 8.

Sintaxis de ruta estática

Nota. Ejemplo de sintaxis de ruta estática. Elaboración propia, realizado con Edraw Max.

- Dirección de red: Es la dirección IPv4 destino que se desea

- alcanzar, agregándola a la tabla de enrutamiento

- Máscara de subred: identifica la máscara de subred de la red remota

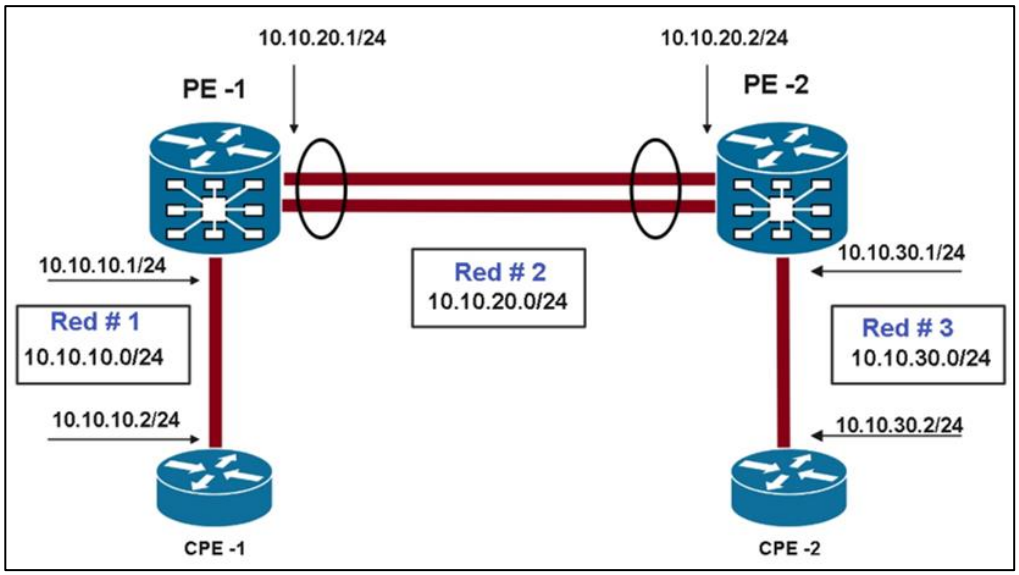
- Dirección IP: corresponde a la dirección IPv4 de siguiente salto, el cual estará dirigido el paquete.

- Interfaz de salida: identifica la interfaz donde se reenviarán los paquetes para lograr alcanzar la red destino.

Como ejemplo de la implementación de este tipo de enrutamiento se presenta la siguiente topología.

Figura 9.

Topología base para los ejemplos de enrutamiento



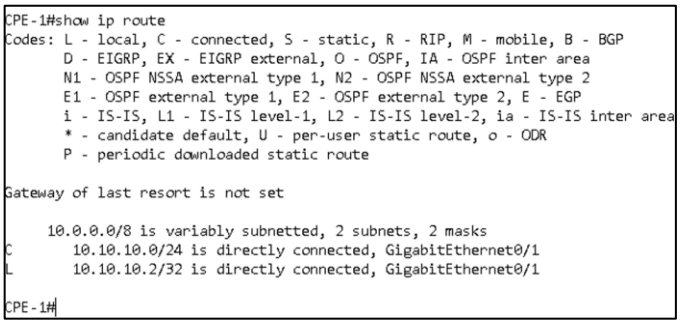
Nota. Ejemplo de enrutamiento. Elaboración propia, realizado con Edraw Max.

Donde ambos equipos CPE-1 y CPE-2 no son capaces de comunicarse entre sí, por lo que se necesita agregar las redes faltantes que desconocen en sus tablas de enrutamiento. Desde el punto de vista del CPE-1 solo conoce la red #1.

Para lograr la comunicación hacia el CPE-2 debe de conocer cómo llegar hacia la red # 2 y finalmente la red # 3, al igual que los demás equipos deben de conocer las redes faltantes en sus tablas de enrutamiento.

Figura 10.

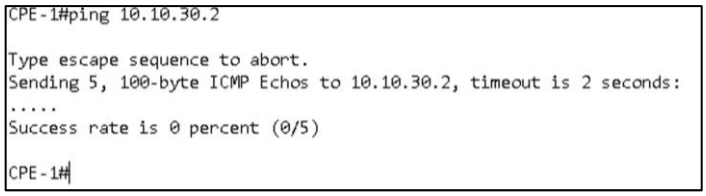
Enrutamiento de CPE-1



Nota. Ejemplo de enrutamiento. Elaboración propia, realizado con Cisco iOS.

Figura 11.

Prueba de comunicación entre CPE-1 y CPE-2



Nota. Ejemplo de prueba de comunicación. Elaboración propia, realizado con Cisco iOS.

Para establecer comunicación entre los dos equipos, es posible configurar manualmente rutas estáticas en todos los enrutadores que conforman la topología, indicando la red que se pretende alcanzar y la forma en que se enviaran los paquetes, pudiendo utilizarse una dirección IP de siguiente salto o una interfaz física para dar salida a la información.

Se configuran las rutas estáticas para cada equipo siguiendo la sintaxis mencionada anteriormente, quedando de la siguiente manera.

- Ruta estática sobre CPE-1
  - ip route 10.10.20.0 255.255.255.0 10.10.10.1
  - ip route 10.10.30.0 255.255.255.0 10.10.10.1

- Ruta estática sobre CPE-2
  - ip route 10.10.10.0 255.255.255.0 10.10.30.1
  - ip route 10.10.20.0 255.255.255.0 10.10.30.1

- Ruta estática sobre PE-1
  - ip route 10.10.30.0 255.255.255.0 10.10.20.2

- Ruta estática sobre PE-2
  - ip route 10.10.10.0 255.255.255.0 10.10.20.1

En la figura 12 se observa que las redes # 2 y 3 ya se conocen desde el CPE-1 por lo que se corre una prueba de conectividad y la comunicación se completa con éxito hacia CPE-2.

Figura 12.

Enrutamiento en CPE-1

```
CPE-1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.10.10.0/24 is directly connected, GigabitEthernet0/1
L       10.10.10.2/32 is directly connected, GigabitEthernet0/1
S       10.10.20.0/24 [1/0] via 10.10.10.1
S       10.10.30.0/24 [1/0] via 10.10.10.1

CPE-1#
```

Nota. Ejemplo de enrutamiento en CPE-1. Elaboración propia, realizado con Cisco iOS.

Figura 13.

Prueba de conectividad hacia CPE-1

```
CPE-1#ping 10.10.30.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.30.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/16/51 ms

CPE-1#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con Cisco iOS.

1.7. Enrutamiento dinámico

En una red grande con muchas subredes, la configuración y el mantenimiento de rutas estáticas conlleva a una sobrecarga administrativa y operativa, a tal grado que cuando se producen cambios en la red, como un enlace fuera de servicio o la implementación de un nuevo segmento de red pueden suponer tareas muy tediosas debido a la configuración de nuevas rutas estáticas hacia un nuevo destino. Todos estos problemas de administración pueden verse solucionados con la implementación de protocolos dinámicos de enrutamiento, ayudando a los administradores a reducir la carga de tareas de configuración y de mantenimiento.

Los protocolos de enrutamiento dinámico están integrados por algoritmos y procesos que permiten a los enrutadores intercambiar información entre sí para aprender y calcular las mejores rutas hacia cualquier destino.

Los protocolos de enrutamiento dinámico cuentan con las siguientes características.

- Estructura de datos: la información de las rutas aprendidas de cómo llegar a los diferentes destinos deben de ser almacenados en tablas o bases de datos y clasificarlos con la mejor métrica aprendida, esta información se guarda en la memoria RAM del equipo.

- Mensajes de protocolo de enrutamiento: todos los protocolos de este tipo utilizan varios tipos de mensajes para el descubrimiento de vecinos, intercambiar su base de datos con los demás equipos y facilitar las tareas para el descubrimiento de la red.

- Algoritmo: en el proceso del descubrimiento de vecinos para conocer como es la estructura total de la red, los protocolos de ruteo siguen una serie de pasos que les permite conocerse entre equipos y facilitar toda la información de ruteo para determinar los mejores caminos hacia múltiples destinos.

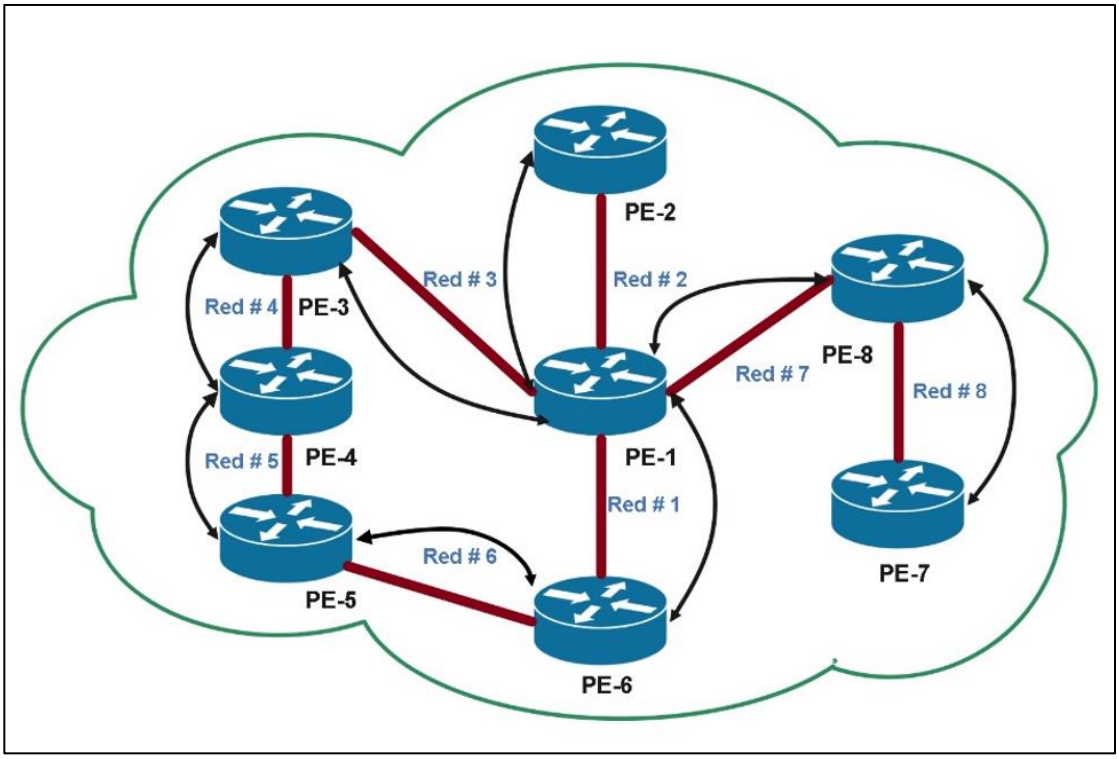
Los protocolos de enrutamiento dinámico determinan de forma automática la mejor ruta hacia cada red para luego instalarse en las tablas (RIB) o tablas de enrutamiento, solo las rutas con la métrica más baja se instalarán en esta base de datos tomando en cuenta la distancia administrativa de los demás protocolos de enrutamiento. Por ejemplo, una ruta estática que tiene una distancia administrativa de uno tendría una mayor prioridad sobre la misma red descubierta por un protocolo de enrutamiento dinámico.

Para ilustrar de mejor manera los beneficios sobre los protocolos de enrutamiento dinámico se presenta la siguiente topología de la figura 14 donde se puede observar que existen varios segmentos que conforma toda la red. Si la topología fuera configurada por enrutamiento estático el equipo PE-6 necesita de seis rutas estáticas para lograr enviar información a cualquier punto de la red y ocurre la misma situación para los demás equipos por lo que no es un proceso escalable para rutas estáticas.

En esta situación es necesaria la implementación de protocolos de enrutamiento dinámico que se encarguen de conocer de forma automática las redes provenientes de los demás equipos y enviar las redes que conocen de forma local y posterior al contar con esa información se procede a seleccionar un camino para llegar a un destino en particular.

Figura 14.

Ejemplo de enrutamiento dinámico



Nota. Enrutamiento dinámico. Elaboración propia, realizado con Edraw Max.

Los protocolos de enrutamiento dinámico se pueden clasificar en dos grupos los cuales pueden ser:

- Protocolos de Enrutamiento Vector Distancia
- Protocolos de Enrutamiento de Estado de Enlace

1.7.1. Protocolos de enrutamiento vector distancia

Este tipo de enrutamiento se caracteriza por intercambiar información entre todos los enrutadores para construir y mantener sus tablas de enrutamiento, cada enrutador mantiene información sobre el costo para llegar a cada red y utiliza esta información para determinar la mejor ruta para enviar paquetes a su destino, las métricas de este tipo de enrutamiento están conformadas por el conteo de numero de saltos, sin llegar a conocer el mapa completo de la topología. Los protocolos vector distancia en IPv4 más utilizados actualmente son los siguientes, RIPv1, RIPv2, EIGRP. Dentro de los proveedores de servicios este tipo de protocolos de enrutamiento no se utilizan por la cantidad de rutas que deben de aprender los equipos, lo que hace menos eficiente con la administración de los recursos, por lo que su aplicación abarca en entornos empresariales.

1.7.2. Protocolos de enrutamiento de estado de enlace

A diferencia del protocolo de vector distancia, este tipo de protocolo se caracteriza de tener una vista completa de la topología de red por lo que todos los enrutadores involucrados deben de estar bajo la misma configuración del proceso de enrutamiento, continuando con nuestra analogía si el protocolo vector distancia utiliza “letreros” para conocer la ruta hacia los destinos. Los protocolos de estado de enlace utilizan un mapa de red que proporciona una vista más detallada para la selección de un mejor camino, este proceso se realiza creando una relación entre equipos vecinos que envían mensajes de saludo, después de sincronizar se intercambian la información dentro de las tablas de ruteo mediante actualizaciones a través de la adyacencia compartida.

Por otro lado, los equipos habilitados con el protocolo de enrutamiento dinámico no envían actualizaciones periódicas de su información a los demás equipos en la red, enviando solamente cuando hay un cambio en la topología, por ejemplo, cuando se produce alguna caída de alguna interfaz, los demás equipos envían actualizaciones informando que la red proveniente sobre la interfaz caída es inalcanzable por lo que deben de calcular otro camino para llegar.

La implementación de estos protocolos suele configurarse en redes extensas permitiendo la escalabilidad, lo que significa que pueden añadir más equipos a la topología existente sin impactar de forma negativa el rendimiento de la red. También es importante mencionar la velocidad de convergencia que define que tan rápido comparten información de las rutas y alcanzan el conocimiento de la topología, el uso de recursos como lo es la memoria RAM y la utilización del CPU deben de ser lo suficientes para almacenar miles de prefijos de red sin generar problemas de latencia al momento de los procesos de reenvío de información. Estas características permiten que sea utilizado por los proveedores de servicios y en las redes corporativas siendo los más utilizados los protocolos IS-IS y OSPF.

1.8. Protocolo OSPF

El protocolo OSPF se encuentra dentro de los protocolos de enrutamiento dinámico desarrollado para las redes IP que utiliza el algoritmo Dijkstra para calcular la ruta más corta entre dos nodos, este protocolo utiliza el costo sobre las interfaces para calcular su métrica hacia las rutas destino, también tiene en cuenta otros parámetros como el ancho de banda y la congestión de los enlaces. Se identifica con el número de protocolo 89 y trabaja en la capa de red del modelo OSI, soporta diferentes tipos de autenticación y envía todas sus actualizaciones

hacia los demás equipos mediante las direcciones multicast 224.0.0.5 y 224.0.0.6 (Cisco, 2023).

Todos los protocolos dinámicos comparten características similares, usan mensajes de saludo para intercambiar información de las rutas instaladas en su base de datos, además es un protocolo eficaz de convergencia rápida por lo que funciona bien en tamaños de redes grandes gracias a su función de sistema jerárquico que admite áreas para una mejor administración.

En las redes OSPF cada equipo habilitado con este protocolo envía su información de sus vecinos hacia todos los demás enrutadores presentes en la red, por lo que cada uno de los equipos que ejecutan OSPF conocerá la ubicación de los demás dentro de la topología presente, lo que permite a cada enrutador calcular la ruta más corta libre de bucles hacia un destino en particular (Cisco, 2023).

1.8.1. Estructura de datos

El protocolo OSPF maneja diferentes tipos de base de datos para almacenar su información dividiéndose entre 3 tipos de tablas, las cuales se mencionan a continuación, adicional se utilizará la topología de la Figura 9 para representar las tablas mencionadas.

- Tabla de enrutamiento: guarda el listado completo de las rutas generadas para diferentes destinos brindando la información sobre el prefijo aprendido, sobre que interfaz de salida enviar el paquete y el tiempo que lleva aprendida la ruta.

Figura 15.

Enrutamiento para el protocolo OSPF

```
CPE-1#show ip route ospf
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O       10.10.20.0 [110/2] via 10.10.10.1, 00:05:55, GigabitEthernet0/1
O       10.10.30.0 [110/3] via 10.10.10.1, 00:04:43, GigabitEthernet0/1

CPE-1#
```

Nota. Ejemplo de enrutamiento para protocolo. Elaboración propia, realizado con Cisco iOS.

- Tabla de vecinos: se almacena la información de los enrutadores habilitados con OSPF donde se estableció comunicación bidireccional formando una adyacencia compartida sobre alguna de sus interfaces, previamente cumpliendo con los criterios establecidos para formar una vecindad.

Figura 16.

Vecinos en OSPF

```
CPE-1#show ip ospf neighbor


Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/BDR        00:00:32    10.10.10.1      GigabitEthernet0/1
CPE-1#
```

Nota. Ejemplo de vecinos. Elaboración propia, realizado con Cisco iOS.

- Tabla de topología (LSDB): es la base de datos donde se puede encontrar toda la información sobre los otros enrutadores dentro de la red por lo que representa un mapa completo sobre la topología configurada.

Figura 17.

Topología en OSPF

```
CPE-1#show ip ospf database
            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         659         0x80000002 0x0028e3 1
2.2.2.2         2.2.2.2         622         0x80000004 0x00821e 2
3.3.3.3         3.3.3.3         555         0x80000004 0x00df8f 2
4.4.4.4         4.4.4.4         555         0x80000002 0x00fcce 1

                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
10.10.10.2      1.1.1.1         659         0x80000001 0x005655
10.10.20.1      2.2.2.2         622         0x80000001 0x006f1e
10.10.30.2      4.4.4.4         555         0x80000001 0x008c6f
CPE-1#
```

Nota. Ejemplo de topología. Elaboración propia, realizado con Cisco iOS.

1.8.2. Funcionamiento basado en áreas

A medida que una red se encuentre en crecimiento constante el mantenimiento de todas las bases de datos constituye para cada equipo un alto consumo de recursos de procesamiento y ancho de banda entre los enlaces, debido a la cantidad de dispositivos que deben de compartir sus actualizaciones, por lo que se convierte en un problema de saturación limitando la capacidad de la red. Para evitar todos estos inconvenientes, se logra modificar el protocolo dividiéndose en distintos tipos de áreas cuya función es limitar el tamaño de las tablas de enrutamiento y servir como contención para diversos tipos de actualizaciones, facilitando el mantenimiento de la red (Cisco, 2023).

Actualmente las redes OSPF se pueden definir en distintos tipos de áreas cada una es utilizado bajo ciertas topologías que requiera propagar información de forma específica, pero las más utilizadas son las siguientes.

- Área de Backbone: es conocida como el área 0 es el área que interconecta todas las demás áreas sirviendo de referencia en todas las implementaciones configuradas con OSPF. Todas las áreas involucradas deben de conectarse al área 0 de forma obligatoria para prevenir bucles de enrutamiento (Cisco, 2023).

- Área Estándar: este tipo de área debe conectarse al área de Backbone y se identifica con otro número de área, todos los enrutadores configurados en un área en especifica deben de conocer a los demás equipos y contiene la misma base de datos de topología sin embargo cada equipo mantiene su propia tabla de enrutamiento (Cisco, 2023).

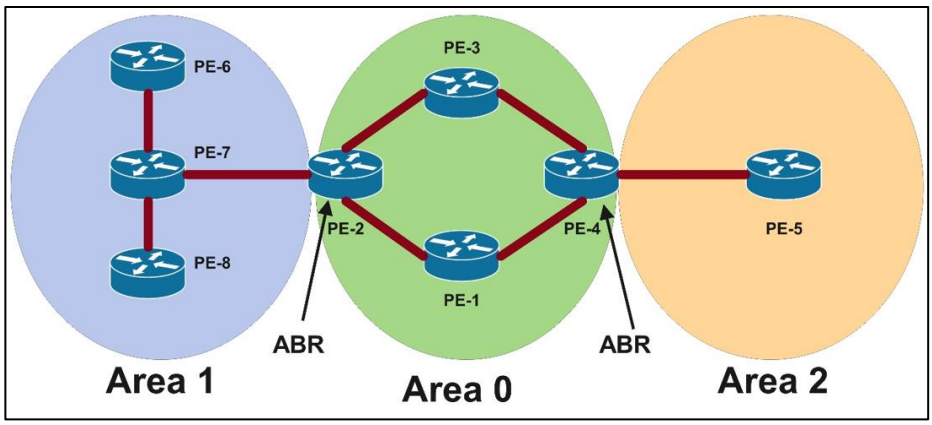
En el protocolo OSPF, la información presente en las tablas de enrutamiento de un enrutador dependerá del tipo de área donde se encuentre, incluso un mismo enrutador puede pertenecer a dos áreas por lo que ese equipo toma un rol más importante en la red.

En la siguiente topología se pude observar cómo los equipos PE-2 y PE-4 se encuentran entre dos áreas diferentes por lo que cumplen la función de un router (ABR).

La función principal de los ABR consiste en evitar la propagación de las actualizaciones periódicas fuera del área que corresponde y limitar la cantidad de entradas a la tabla de topología.

Figura 18.

Topología OSPF ejecutando tres áreas



Nota. Ejemplo de topología OSPF. Elaboración propia, realizado con Edraw Max.

1.8.3. Tipos de paquetes

El protocolo OSPF utiliza diferentes paquetes para lograr propagar su propia información

- *Hello*: son los primeros mensajes que utiliza el protocolo para establecer una adyacencia con sus equipos vecinos, utilizado para el descubrimiento y mantenimiento de la red OSPF.

- *Database Description*: este tipo de paquete es intercambiado cuando ya exista comunicación bidireccional entre equipos con OSPF.

- *Link State Request* (LSR): es utilizado para consultar información específica que no se conozca o que se encuentre desactualizado hacia un vecino.

- *Link State Update* (LSU): envía la información solicitada por el LSR. Sirve como contenedor a los diferentes tipos de actualizaciones.

- *Link state acknowledgement* (LSAck): es el paquete de acuse de recibo para saber si los LSAs llegaron al destino de forma confiable, reconociendo la información recibida.

1.8.4. Adyacencia OSPF

Este tipo de enlace se forma cuando dos enrutadores vecinos se encuentran dentro de la red OSPF y reconocen al otro equipo como parte de la topología para luego sincronizarse e intercambiar su información de enrutamiento. El proceso de formación de vecindades se logra atravesando diferentes estados los cuales se presentan a continuación.

- Estado *Down*: estado en el cual no existe comunicación con el dispositivo

- Estado *Init*: se reciben los paquetes *hello* o de saludo, es el primero paso para lograr la adyacencia.

- Estado *Two-Way*: es la fase de sincronización de la base de datos, intercambian información específica dependiendo el tipo de enlace que comparta con el equipo vecino.

- Estado *ExStart*: comienza el intercambio de paquetes de la base de datos, estableciendo comunicación bidireccional.

- Estado *Loading*: se intercambian paquetes LSR y LSU para obtener información adicional de la ruta.

- Estado *Full*: en este estado los equipos convergen de forma que se encuentran listos para propagar su información a los demás equipos.

1.8.5. Requerimientos de configuración

Para levantar adyacencias OSPF se necesita contar con la siguiente configuración en los equipos.

- Identificador de proceso es un número que se utiliza para identificar el proceso de enrutamiento OSPF, puede ser cualquier valor entre 1 y 65,535 el cual se elige por el administrador, sirve para distinguir entre varios procesos iniciados en el mismo equipo en el caso de que existan varios arrancados simultáneamente (Cisco, 2023).

- Identificador del enrutador (router ID) utilizado para identificar cada enrutador de la red OSPF (Cisco, 2023).

- La dirección de red que se agregue al proceso OSPF, se estará compartiendo a los demás equipos en forma de actualizaciones, esto indica que estará enviando las redes que conoce de forma local hacia los demás equipos con el propósito de crear la base de topología (Cisco, 2023).

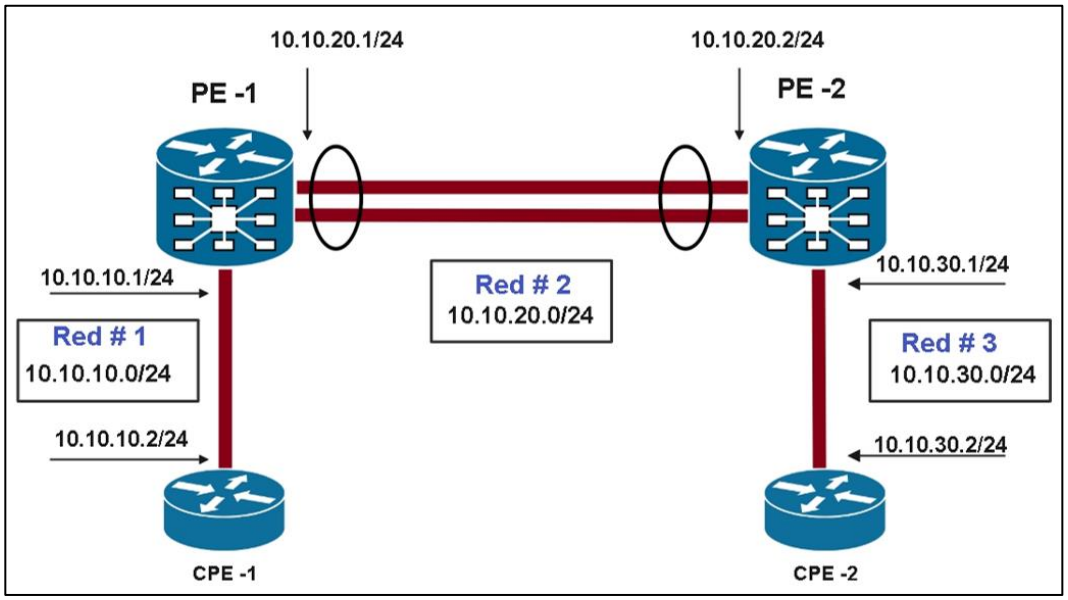
- Máscara de Wildcard, es una máscara de bits que indican que partes de una dirección IP son relevantes para la ejecución de una determinada tarea (Cisco, 2023).

- El número de área configurado indicará a que grupo de enrutadores pertenece, por lo que tendrán la misma información de estado de enlace.

Como ejemplo de configuración del protocolo de enrutamiento OSPF se utilizará nuevamente la topología utilizada anteriormente, con el fin de ilustrar la configuración de este protocolo.

Figura 19.

Topología base para los ejemplos de enrutamiento



Nota. Ejemplo de topología. Elaboración propia, realizado con Edraw Max.

Como en el ejemplo anterior se necesita que ambos equipos CPE-1 y CPE-2 logren comunicarse entre sí, por lo que se necesita que CPE-1 tenga en su tabla de enrutamiento las redes #2 y 3.

Para agregar la configuración de OSPF sobre el CPE-1 es necesario iniciar un numero de proceso, en este caso se utilizará el proceso número 1. Una vez iniciado el proceso de CPE-1 se asigna el identificador único del equipo sobre la red que se estará utilizando 1.1.1.1 seguido de las redes que se encuentran

configuradas de forma local en sus interfaces, para coincidir la porción de red que se estará distribuyendo se coloca la máscara de wildcard en 0.0.0.255 lo que tendrá una coincidencia exacta sobre los 3 primero octetos seguido del número de área el cual pertenecerá el equipo toda esta información se estará enviando a los demás equipos en forma de actualizaciones para poder ser agregadas a sus tablas de enrutamiento.

Se realiza la configuración sobre los demás equipos de la red quedando de la siguiente manera.

Figura 20.

Configuración de OSPF

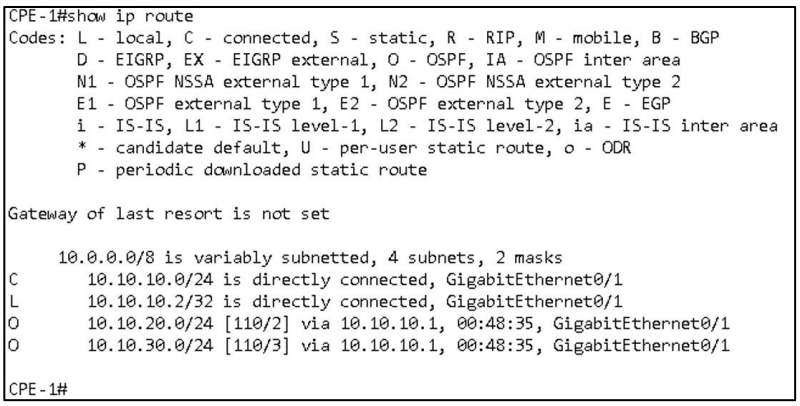
```
CPE-1(config)# router ospf 1
CPE-1(config)# router-id 1.1.1.1
CPE-1(config)# network 10.10.10.0 0.0.0.255 area 0


CPE-2(config)# router ospf 1
CPE-2(config)# router-id 2.2.2.2
CPE-2(config)# network 10.10.30.0 0.0.0.255 area 0


PE-1(config)# router ospf 1
PE-1(config)# router-id 2.2.2.2
PE-1(config)# network 10.10.10.0 0.0.0.255 area 0
PE-1(config)# network 10.10.20.0 0.0.0.255 area 0


PE-2(config)# router ospf 1
PE-2(config)# router-id 3.3.3.3
PE-2(config)# network 10.10.20.0 0.0.0.255 area 0
PE-2(config)# network 10.10.30.0 0.0.0.255 area 0
```

Nota. Ejemplo de configuración de OSPF. Elaboración propia, realizado con Cisco iOS.

Después de realizar las configuraciones de OSPF para cada uno de los equipos se puede observar que en CPE-1 ya conoce las redes 2 y 3 en su tabla de enrutamiento.

Figura 21.

Enrutamiento de CPE-1



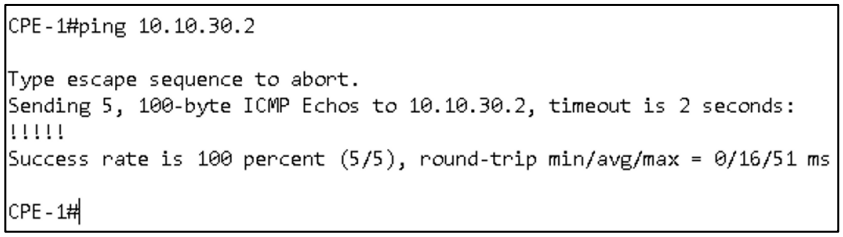
```
CPE-1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.10.10.0/24 is directly connected, GigabitEthernet0/1
L       10.10.10.2/32 is directly connected, GigabitEthernet0/1
O       10.10.20.0/24 [110/2] via 10.10.10.1, 00:48:35, GigabitEthernet0/1
O       10.10.30.0/24 [110/3] via 10.10.10.1, 00:48:35, GigabitEthernet0/1

CPE-1#
```

Nota. Ejemplo de enrutamiento CPE-1. Elaboración propia, realizado con Cisco iOS.

Se aplica una prueba de conectividad hacia CPE-2 donde se puede alcanzar de forma exitosa.

Figura 22.

Prueba de conectividad hacia CPE-2



```
CPE-1#ping 10.10.30.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.30.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/16/51 ms

CPE-1#
```

Nota. Ejemplo de prueba de conectividad hacia CPE-2. Elaboración propia, realizado con Cisco iOS.

1.9. Protocolo BGP

El protocolo de Gateway fronterizo es un protocolo de enrutamiento dinámico exterior que es utilizado para el intercambio de información de enrutamiento entre sistemas autónomos, por ejemplo, proveedores de servicios y empresas, el cálculo de la métrica es realizado por medio de un vector de ruta, es decir que no utiliza el costo de las interfaces, ancho de banda, numero de saltos, entre otros. Para determinar el mejor camino hacia los prefijos, el vector de ruta se basa en atributos que poseen los prefijos y basados en estos atributos puede decidir cuál es el mejor camino (Cisco, 2019).

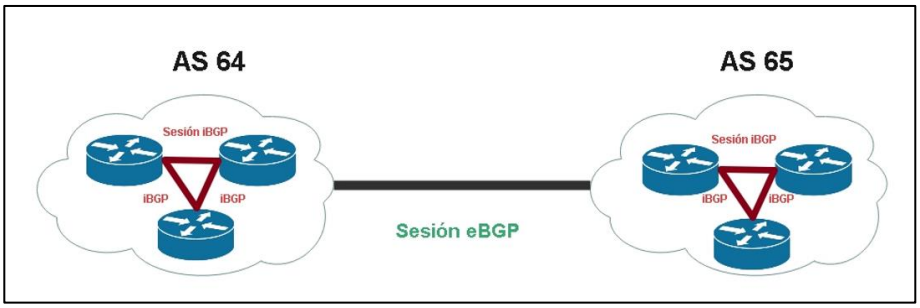
El protocolo BGP utiliza el número de puerto TCP 179 para enviar su información de protocolo, utiliza dos distancias administrativas 20 para sesiones externas y 200 para sesiones internas. Como bien se sabe, una distancia administrativa mide la confiabilidad de un protocolo dinámico, las sesiones externas de BGP son más confiables porque obligatoriamente deben de ser enlaces punto a punto, en cambio con las sesiones internas se encuentran dentro de su propio sistema autónomo y se desconoce la cantidad de saltos para llegar al equipo vecino, por tal razón las sesiones externas tienen una alta confiabilidad en comparación con los demás protocolos esto se ilustra mejor con la figura 23. (Cisco, 2019).

BGP es un protocolo definidos en los siguientes estándares RFC 4271, 4893 y 2858, la ventaja de este protocolo sobre los demás reside en que puede dar transporte a ciertos tipos de tráfico como lo es IPv4, IPv6, tráfico multicast, MPLS, vpnv4, l2vpn, y ciertos tipos de etiquetas, cada uno de este tipo de tráfico es llamado familias de direccionamiento o address family lo cual se le debe de especificar a BGP el tipo de familia que desee transportar, por lo que BGP es ampliamente utilizado por los proveedores de servicios (Cisco, 2019).

Un enrutador configurado con BGP no puede anunciar prefijos que no se encuentren en su tabla de enrutamiento y la única forma que los prefijos se puedan instalar dentro de estas tablas es por medio de un protocolo de sesiones internas siendo IS-IS y OSPF los más utilizado por los proveedores.

Figura 23.

Sistemas autónomos conectados mediante BGP



Nota. Ejemplo de sistemas autónomos. Elaboración propia, realizado con Edraw Max.

1.9.1. Terminología BGP

- BGP externo: son las adyacencias hacia otros enrutadores con otros sistemas autónomos.

- BGP interno: es el tráfico que se enruta dentro de un único sistema autónomo.

- Sincronización: es una técnica que indica que una ruta debe ser conocida por un IGP antes de ser publicadas a otros pares BGP se debe tener en cuenta esta configuración ya que viene deshabilitado por defecto.

- Tabla de vecinos: lista todos los vecinos o pares BGP dentro de la red.

- Tabla BGP: contiene todas las redes aprendidas por cada vecino y los atributos BGP para cada prefijo.

- BGP RIB: almacena información de todos los pares antes de modificar o agregar atributos o filtros.

- IP RIB: es la base de datos del enrutador que almacena todas las rutas por lo que posteriormente se transfiere a la tabla de enrutamiento.

1.9.2. Sistema autónomo (AS)

Es una organización que administra un conjunto de equipos de red que están bajo la misma política de enrutamiento, el cual se identifica mediante un número que es usado para enviar y recibir información de forma externa del AS, las cuales hay de distintos tipos las más comunes son las siguientes (Cisco, 2019).

- Stub AS: se conecta solo a un sistema autónomo (ISP) generalmente es la conexión hacia internet.

- Multihomed AS: se conecta a dos o más AS como redundancia hacia internet.

- Transit AS: provee conexión a través del mismo AS hacia otras redes.

1.9.3. Tipos de mensajes en BGP

El proceso de intercambio de información de los equipos configurados con BGP son los siguientes.

- *Open*: se intercambia información básica de cada par BGP, contiene el tipo de mensaje, versión del protocolo, sistema autónomo, temporizadores, router id y familias de direccionamiento que el equipo soporta.

- *Update*: informa sobre nuevos prefijos y los atributos que llevan asociados.

- *Notification*: notifica sobre los problemas relacionados al establecimiento de adyacencias o porque la vecindad BGP no se logra establecer.

- *Keepalive*: su función es mantener las conexiones entre pares BGP activas.

1.9.4. Tipos de estado de un vecino

El proceso de formación de adyacencias es sumamente importante dentro de la red, debido que a partir de este mecanismo se logra determinar cuando los enrutadores se encuentran listos para propagar los prefijos hacia los demás equipos, al igual que el protocolo OSPF los enrutadores pasan por diferentes estados hasta lograr la adyacencia completa los cuales se mencionan a continuación (Cisco, 2019).

- Idle: comienza la búsqueda de un nuevo vecino he inicia la conexión TCP a través del puerto 179, se mantiene en este estado cuando la contraparte no tiene configurado BGP.

- *Connect*: la sesión TCP fue completada por el mecanismo three-way handshake.

- *Open sent*: comparación de mensajes open.

- *Open confirm*: el vecino confirma el inicio de una sesión BGP.

- Active: se mantiene cuando detecta algún problema en los prefijos provenientes.

- *Established*: la vecindad se completa y el equipo comienza a recibir prefijos.

1.9.5. Atributos en BGP

Los atributos en BGP tienen la capacidad de desviar el tráfico basándose en criterios determinados por los administradores de la red, estas características pueden ser utilizadas para distinguir y seleccionar el mejor camino, básicamente los atributos son la métrica de BGP (Cisco, 2019).

Los atributos más utilizados por BGP son los siguientes.

- AS-path: lista los sistemas autónomos que deben ser atravesados para llegar a un determinado prefijo.

- Next-Hop: dirección IP del enrutador que está publicando el prefijo.

- Weight: atributo de significado local, utilizado solamente en enrutadores del fabricante Cisco.

- Origin: determina si el prefijo fue publicado de forma local, por sesiones BGP externas o fue redistribuido.

- Local Preference: valor numérico de significado local, se prefiere el valor mayor.

- Atomic Aggregate: determina si un prefijo fue sumarizado.

- Community: etiqueta ciertas rutas que contienen características comunes.

- Originator ID: indica el Id del enrutador iBGP que publica el prefijo en escenario de route-reflector como prevención de bucles.

- Cluster ID: indica el ID del enrutador que realiza la función de route-reflector.

- MED: informa a los enrutadores que se encuentran fuera del AS qué camino tomar para entrar en el AS, es conocido como la métrica externa de la ruta.

Estos atributos se pueden clasificar de dos grupos.

- Well-known: atributos cuya utilización es obligatoria.

- Optional: atributos opcionales.

Además, cada una de las dos divisiones se dividen a su vez dentro de dos grupos lo que permite una mejor clasificación.

En el grupo de los atributos Well-know están:

- Mandatory: estos atributos son requeridos y deben ser reconocidos por todas las implementaciones de BGP.

- Discretionary: no son requeridos, pero en el caso de estar presentes todos los enrutadores que ejecuten BGP tiene que reconocerlos y actuar de acuerdo con la información que contienen.

El grupo de Optional contiene la siguiente clasificación:

- Transitive: este puede ser comprendido o no por el router local y siempre deben ser distribuidos por los vecinos.

- Nontransitive: no son reenviados hacia los demás vecinos si no se reconoce de forma local.

La clasificación de los atributos se resume dentro de la siguiente tabla.

Tabla 1.

Atributos utilizados en BGP

Well-know		**Optional**	
Mandatory	Discretionary	Transitive	Nontransitive
AS-path	Local preference	Aggregator	Originator ID
Next-hop	Atomic aggregate	Community	Cluster ID
Origin			Med

Nota. Los atributos BGP sirven para elegir la mejor ruta para llegar a un destino. Elaboración propia, realizado con Excel.

1.9.6. Proceso de selección de ruta

Los enrutadores configurados con BGP suelen recibir varias rutas al mismo destino por lo que cada prefijo es sometido a un proceso de selección de forma jerárquica dependiendo de los atributos que tengan asignados. La siguiente lista proporciona las reglas que se utilizan para determinar la mejor ruta (Cisco, 2019).

- Prefiere la ruta con el Weight más alto, teniendo en cuenta que este parámetro solo es válido para el fabricante de Cisco.

- Prefiere las rutas con el mayor Local-preference tiene un valor establecido en 100 por defecto.

- Prefiere la ruta que se originó localmente.

- Prefiere las rutas con el menor AS-PATH, menor cantidad de saltos de sistema autónomos.

- Prefiere las rutas con el menor Origin, esto indica si la ruta fue redistribuida o anunciada con el comando network.

- Prefiere las rutas con el menor MED, es un valor numérico que se interpreta como la métrica del protocolo interno, pero en este caso la métrica se manipula de forma manual porque no es acumulativa.

- Prefiere las rutas de vecinos eBGP sobre los iBGP.

- Prefiere las rutas con la métrica IGP más baja hacia el BGP Next-hop.

- Prefiere la ruta eBGP más antigua instalada en la tabla de enrutamiento.

- Prefiere la ruta que tenga el vecino BGP con el menor router-id.

- Prefiere la ruta que tenga el vecino con la dirección IP más baja.

1.9.7. Requerimientos de configuración

Para poder implementar este protocolo en un entorno real se necesita configurar de forma obligada los siguientes pasos.

- Para su activación se necesita un sistema autónomo público o privado de 2 o 4 bytes.

- La instancia de BGP debe de llevar un identificador de enrutador o comúnmente conocido como router-id para identificar el equipo de la red BGP.

- Se necesita un protocolo interno IGP para las sesiones internas, para el transporte de prefijos como las interfaces de loopback de vecindades BGP y los prefijos internos que deben ser anunciados hacia otros ISP.

- Para la prevención de bucles, la red BGP utiliza la configuración de route-reflector para que cada prefijo sea enviado por este equipo hacia los demás vecinos BGP.

Para el siguiente ejemplo se utilizará la topología de la Figura 23 el cual se configura una vecindad BGP entre dos sistemas autónomos los cuales son el AS 64 y AS 65.

Para el sistema autónomo 64 se comparte la siguiente configuración para lograr la adyacencia BGP.

Figura 24.

Configuración BGP para PE-1

```
PE-1(config)# router bgp 64
PE-1(config)# bgp router-id 10.10.10.10
PE-1(config)# neighbor 200.192.168.1 remote-as 65
```

Nota. Ejemplo de configuración BGP. Elaboración propia, realizado con Cisco iOS.

De igual forma se aplican los mismos comandos para el AS 65.

Figura 25.

Configuración BGP para PE-2

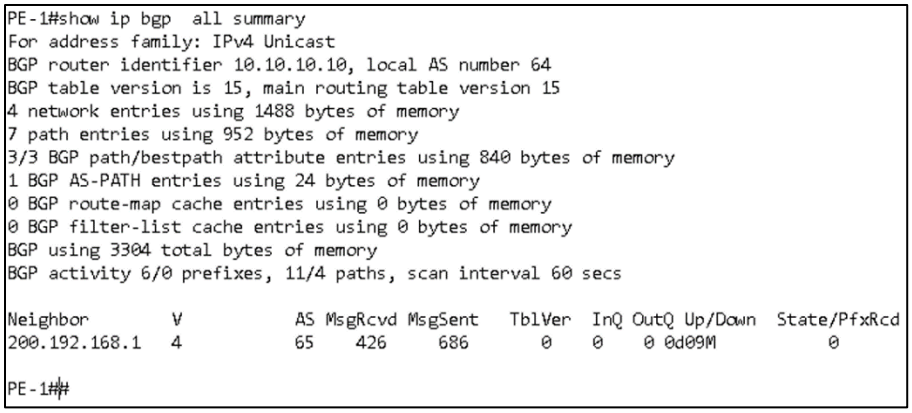
```
PE-2(config)# router bgp 65
PE-2(config)# bgp router-id 20.20.20.20
PE-2(config)# neighbor 200.192.168.2 remote-as 64
```

Nota. Ejemplo de configuración BGP. Elaboración propia, realizado con Cisco iOS.

Ahora si se verifica la tabla de vecinos de cualquiera de los equipos se observa que la adyacencia ya fue creada entre los dos AS y están listos para compartir prefijos entre sí.

Figura 26.

Vecinos de BGP



```
PE-1#show ip bgp  all summary
For address family: IPv4 Unicast
BGP router identifier 10.10.10.10, local AS number 64
BGP table version is 15, main routing table version 15
4 network entries using 1488 bytes of memory
7 path entries using 952 bytes of memory
3/3 BGP path/bestpath attribute entries using 840 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3304 total bytes of memory
BGP activity 6/0 prefixes, 11/4 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
200.192.168.1   4           65     426     686       0    0    0 0d09M           0

PE-1#
```

Nota. Ejemplo de vecinos, que también se les conoce como pares. Elaboración propia, realizado con Cisco iOS.

1.10. Sistema operativo Cisco Systems

En una red de proveedor de servicios los administradores de red deberán ser capaces de aplicar configuraciones tanto en los equipos del Core, equipos de distribución y finales. Dentro de este estudio se estará orientando la configuración del fabricante Cisco System que actualmente cuenta con tres sistemas operativos cada una enfocada a la cantidad de procesamiento y aplicaciones que puedan operar, las cuales se describen a continuación.

- Cisco IOS: (Internetwork Operating System). Es el sistema operativo utilizado por la mayoría de los enrutadores, su función principal es permitir la interacción con los dispositivos a través de una interfaz de línea de comando.

Este sistema operativo tiene la característica de ser monolítico quiere decir que bajo un mismo proceso se ejecutan todos los protocolos que se encuentren configurados, por lo que si ocurre alguna falla en algún protocolo, incrementa el proceso y detiene todo el sistema operativo quedando fuera he interrumpiendo la comunicación hasta que sea reiniciado.

En un entorno de redes corporativas se puede observar equipos con este sistema operativo instalado en los equipos finales también conocidos como CPE (*Customer Provided Equipment),* ya que es capaz de cubrir tareas de enrutamiento, conmutación y telecomunicaciones.

- Cisco IOS XE: sistema operativo basado en Linux de estructura modular lo que permite trabajar los protocolos de forma separada, la carga de trabajo es distribuida entre varios CPU realizando los procesos de forma eficiente, por lo que si en algún momento cualquier protocolo configurado dentro del equipo falla, solo estará fallando el proceso ejecutado por el protocolo sin detener todo el sistema operativo con la capacidad de anular el proceso , actualmente este sistema operativo se puede encontrar en los equipos de la serie ASR en comparación con el sistema iOS es mucho más escalable.

- Cisco IOS XR: es un sistema operativo orientado a la interconexión de redes ampliamente utilizado por los proveedores de servicios para integrar la parte del Core y la distribución de la red, este iOS tiene como objetivo proporcionar alta disponibilidad gracias al soporte de redundancia por hardware y métodos de contención de fallas como espacios de memoria protegidos para procesos individuales, lo que permite reiniciar un proceso cuando se requiera sin impactar de forma negativa a otro proceso diferente, además mejora la escalabilidad para grandes configuraciones

sobre hardware y mantiene un modelo de distribución de software basado en paquetes lo que conlleva a instalar y eliminar características opcionales como el enrutamiento de multicast y MPLS mientras el equipo se encuentre en servicio.

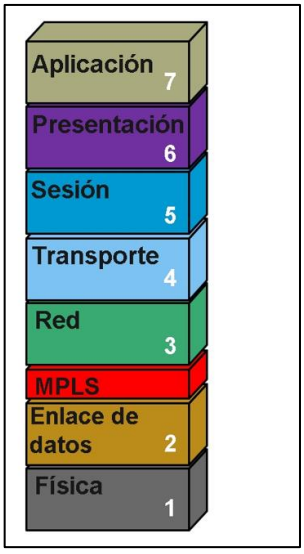
2. ARQUITECTURA MPLS

La conmutación de etiquetas multiprotocolo es una forma de transmisión de datos utilizada ampliamente por los proveedores de servicios para dar solución a los problemas de escalabilidad de la capa de red lo que mejora la flexibilidad en la entrega de los servicios de enrutamiento, las redes MPLS basa su funcionamiento en la asignación de etiquetas a los paquetes de datos.

La tecnología MPLS crea una etiqueta que se añade al paquete creando una capa intermedia entre la capa de enlace de datos y red por lo que situando a MPLS dentro del modelo OSI vendría siendo la capa 2.5.

Figura 27.

Operación de MPLS en el modelo OSI



Nota. Ejemplo de operación MPLS. Elaboración propia, realizado con Edraw Max.

El motivo para agregar etiquetas a los paquetes entrantes a la red MPLS es enviar toda la información hacia los destinos, basada en decisiones de reenvió de etiquetas dejando a un lado el enrutamiento tradicional por protocolos de enrutamiento dinámico, debido a que los proveedores de servicios deben de procesar miles de prefijos en las tablas de enrutamiento y tomar las decisiones basadas en IP conllevaría a problemas de latencia hacia los destinos, ya que se tendría que revisar cada una de las rutas de las miles instaladas para tomar una decisión de envío, por lo que MPLS puede solucionar el problema agrupando los prefijos a unas cuantas etiquetas para decidir hacía que enrutador de borde se debe dirigir la salida del tráfico, por lo que el enrutamiento se vuelve mucho más rápido (Penaloza, 2019).

2.1. La etiqueta MPLS

La etiqueta MPLS se encuentra definido por el estándar RFC3032 el cual menciona sobre la estructura del encabezado que agrega a los paquetes que ingresan de la red ISP, este encabezado se encuentra después de los encabezados de la capa de enlace de datos (Penaloza, 2019)

El encabezado MPLS consta de 32 bits las cuales se dividen en los siguientes campos.

- The label value: es el valor de la etiqueta de 20 bits conteniendo toda la información de a donde se deben de enviar los paquetes permitiendo tomar decisiones más fluidas y escalables dentro de la red.

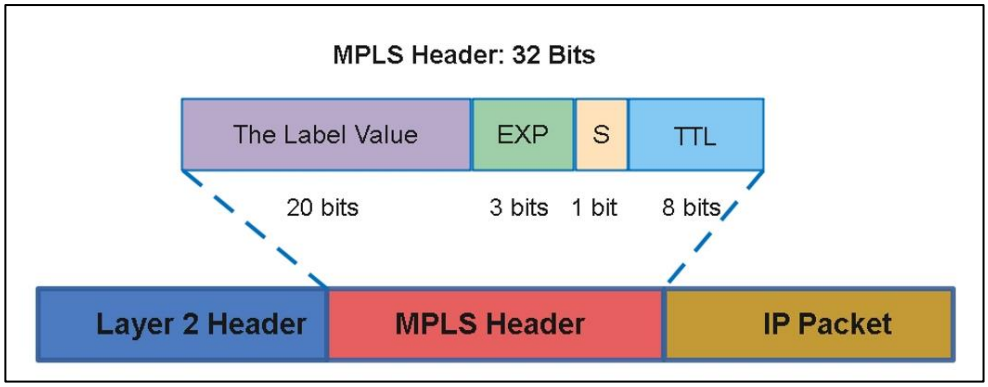
- Experimental (Exp): se reserva para etiquetas sobre los paquetes, políticas de Qos.

- Bottom of Stack (S): es la etiqueta que se usa para diferenciar de un servicio de otro, pueden ser diferentes etiquetas para ingeniería de tráfico o rutas de respaldo, si el valor del bit corresponde al valor 1 significa que es la última etiqueta por leer de lo contrario el equipo sabe que hay más etiquetas a evaluar (Penaloza, 2019).

- Time to live (TTL): es usado para limitar la duración de los paquetes de la red, por defecto es utilizado el valor de 255 con decrementos en cada salto.

Figura 28.

Encabezado MPLS



Nota. Ejemplo de encabezado MPLS. Elaboración propia, realizado con Edraw Max.

2.2. Proceso de envío de paquetes

La arquitectura MPLS fue creada para agilizar el proceso de enrutamiento y tomar mejores decisiones, inicialmente se necesita de las características de los protocolos dinámicos que ayudará en la construcción de las tablas de enrutamiento y esa información se toma como base para realizar el etiquetado de todos los destinos que se encuentran instaladas en el equipo.

Los enrutadores manejan diferentes áreas de funcionamiento las cuales son el plano de administración, plano de datos y plano de control. El plano de administración controla las sesiones de gestión sobre el equipo como pueden ser SSH, telnet, SNMP, http, entre otros.

El plano de control realizará toda la verificación y aplicará los diferentes algoritmos que determinan como se ejecutan el envío de paquetes, si se esta manejando el protocolo OSPF en este nivel es donde se maneja el intercambio de rutas entre lo enrutadores configurados con OSPF para construir la ip routing table (RIB), que contiene todas las rutas que se encuentran instaladas en el equipo, posteriormente al tener ya la tabla RIB serán seleccionadas las mejores rutas o las rutas que serán utilizados para realizar la transmisión de paquetes y se genera otra tabla llamada FIB *forwarding information* base, es decir cuando un paquete llega al router esta tabla es la que se utilizara para encontrar la ruta que coincide y ver a que enrutador mandar ese paquete(De Ghein, 2007).

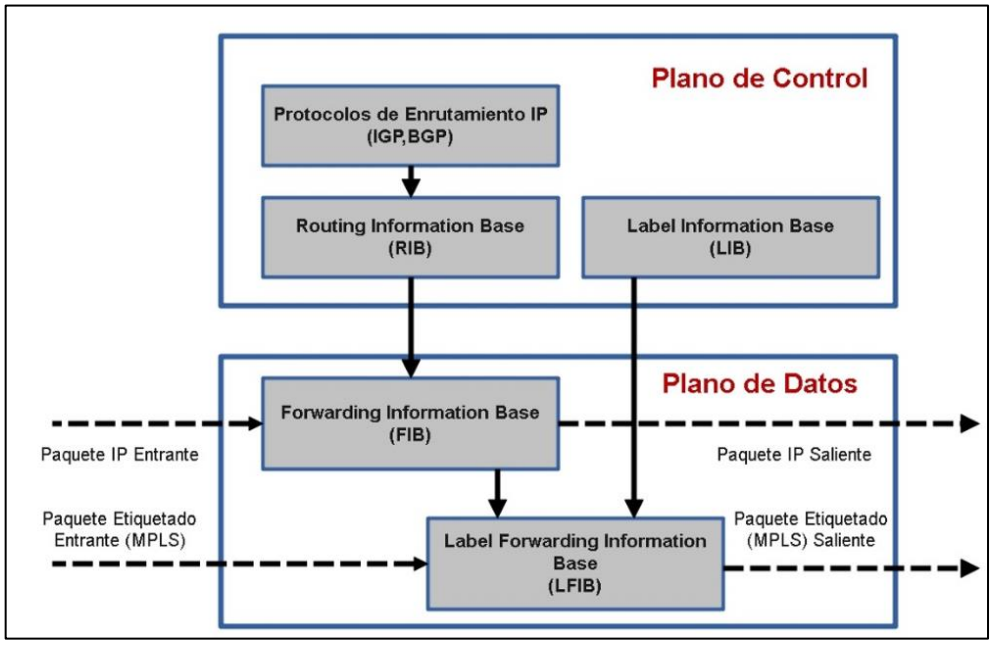
Cuando se utiliza MPLS cada enrutador será parte del dominio MPLS, en este dominio se encuentra el protocolo, que estará corriendo entre todas las interfaces con MPLS configurado dicho protocolo se llama LDP (*label distribucion protocol*), lo que hará es el intercambio de todas las rutas entre las interfaces. Los enrutadores que tengan activado MPLS una vez recibida toda la información etiquetada se construirá la tabla LIB (*label information base*), que es muy similar a la tabla RIB solo que en esta tabla se utilizan etiquetas, posteriormente se construye la tabla LFIB análogamente a la tabla FIB estas dos tablas de MPLS se encuentran gestionadas por plano de datos (De Ghein, 2007).

En un entorno real pueden ingresar a los equipos dos tipos de paquetes, etiquetados y no etiquetados, si los paquetes entrantes tienen o no etiqueta es como el *router* decidirá si va a utilizar la tabla FIB o la LFIB, si no tiene etiqueta

estará utilizando la FIB y si trae etiqueta esta se enviará y será analizada por la LFIB.

Figura 29.

Proceso de envío de paquetes en MPLS



Nota. Ejemplo de envío de paquetes. Elaboración propia, realizado con Edraw Max.

2.3. Terminología MPLS

En MPLS cada equipo que integra la red realiza un rol de funcionamiento dependiendo de la ubicación en la que se encuentren, de esta forma se les nombra específicamente por su rol (De Ghein, 2007). Entre ellos están:

- *Customer Provider Edge Router* (CPE *Router*): es el equipo ubicado dentro de las instalaciones del cliente, es el equipo final de la red ISP, utilizado para dar servicios de interconexión de transporte de datos.

- ELSR (*Edge Label Switching Router*): es el equipo de ingreso y salida de la red MPLS del ISP, es el punto de interconexión de todos los equipos CPE, es decir todos los equipos finales de múltiples clientes ubicadas en la misma zona geográfica se conectarán al mismo *router* ELSR.

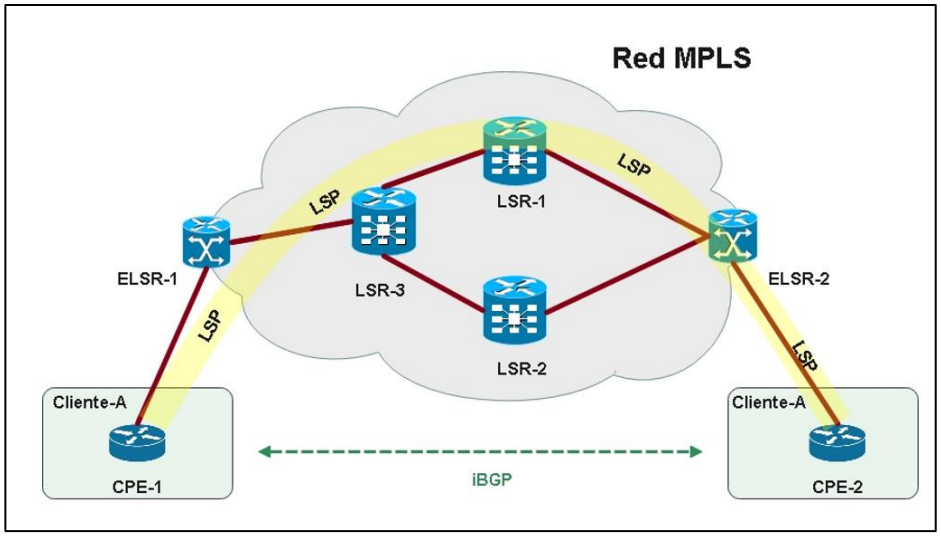
- LSR (*Label Switching Router*): son los equipos que componen el core de la red del ISP y conectan todos los equipos PE de la red.

- *Forwarding Equivalence Class* FEC: es el trato que se le da a un grupo de paquetes que tienen características similares y pueden ser reenviados con la misma etiqueta MPLS.

- LSP *Label Switched Path*: es la ruta que toman todos los prefijos para llegar al destino, es un túnel establecido entre los extremos para compartir rutas por etiqueta teniendo en cuenta que cada LSP es unidireccional.

Figura 30.

Ejemplo sobre una red MPLS



Nota. Ejemplo de red MPLS. Elaboración propia, realizado con Edraw Max.

En las redes MPLS el equipo LSR se dividen en tres tipos que se presentan a continuación:

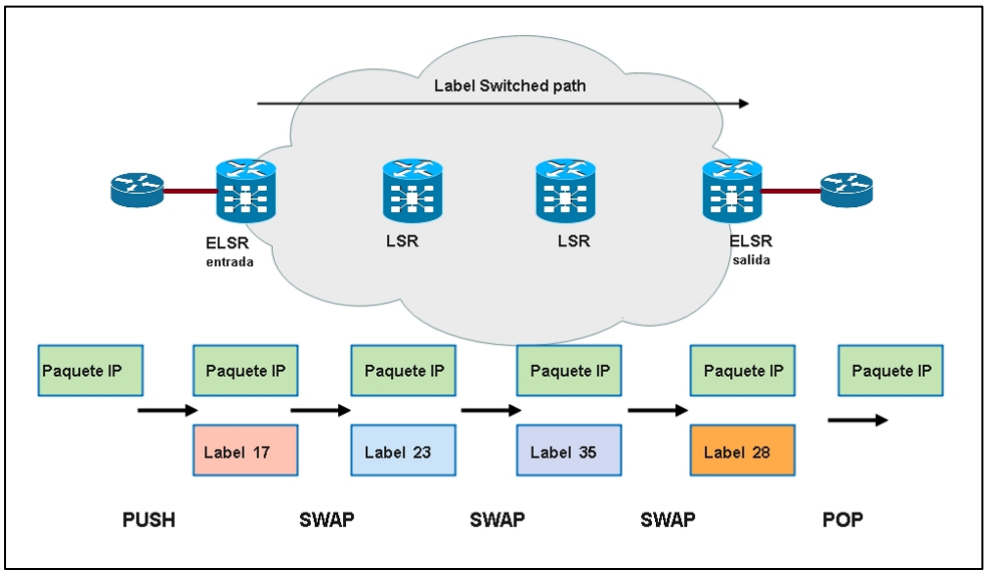
- LSR de ingreso: se utiliza cuando un paquete IP llega al equipo de borde y tiene la responsabilidad de agregar una etiqueta a los prefijos seleccionados para que puedan transitar dentro de la red MPLS, en otras palabras, se dice que realiza un *push* a los paquetes entrantes.

- LSR intermediario: es el enrutador que no es el origen ni el destino, su rol dentro de la red es el intercambio de etiquetas, Cuando el tráfico etiquetado es recibido será reenviado asignando otra etiqueta para llegar al siguiente salto, entonces se dice hace un swap al tráfico entrante.

- LSR de egreso: es el ultimo equipo donde se debe de remover la etiqueta y dejar el paquete des encapsulado de la red MPLS para luego trasladarse hacia la red destino.

Figura 31.

Tipos de LSR y conmutación por etiquetas



Nota. Ejemplo de tipos de LSR. Elaboración propia, realizado con Edraw Max.

2.4. Protocolo de distribución de etiqueta LDP

Las redes de los proveedores de servicio que implementen MPLS requieren de un protocolo que soliciten, publiquen y distribuyan de forma automática las etiquetas, cada enrutador genera localmente la etiqueta para sus prefijos y luego anuncia los valores de etiqueta a sus vecinos que se encuentran dentro del dominio MPLS con el objetivo de establecer una ruta de conmutación

de etiquetas. Esto permite que los LSR puedan distribuir a lo largo de rutas normalmente enrutadas para admitir el reenvío MPLS, con el reenvío de IP.

Cuando un paquete llega a un enrutador, el enrutador mira la dirección de destino en el encabezado de IP, realiza una búsqueda de ruta y reenvía el paquete al siguiente salto, en cambio con el reenvío MPLS cuando un paquete llega a un enrutador, el equipo mira la etiqueta entrante, busca la etiqueta y luego la envía al siguiente salto utilizando la tabla LFIB (De Ghein, 2007).

En las redes de los proveedores de servicios se necesita utilizar un protocolo dinámico IGP para conocer la topología completa del ISP por lo que LDP utiliza esta base de datos para conocer las rutas previamente establecidas por el protocolo dinámico y crear rutas etiquetadas llamadas LSP basadas en la métrica del protocolo IGP.

LDP es un estándar publicado sobre el RFC 30-31, basado en el protocolo de distribución de etiquetas TDP patentado por el fabricante Cisco Systems hoy en día LDP es el protocolo utilizado para la distribución de etiquetas. El funcionamiento de LDP se basa en la creación de una adyacencia hacia el equipo vecino, se establecen sesiones LDP enviando paquetes de saludo de multidifusión UDP con la dirección de multicast 224.0.0.2 utilizando el puerto de origen y destino 646 para descubrir nuevos vecinos, para el resto de mensajes se utiliza TCP , cada enrutador contiene una identificación única llamada LSR ID esto es similar como la mayoría de los protocolos seleccionan un identificador sobre el equipo, cuando se encuentre establecida la sesión LDP en MPLS pueden crearse de dos tipos, sesiones LDP directamente conectadas y sesiones LDP no directamente conectadas (De Ghein, 2007).

2.4.1. Sesiones LDP directamente conectadas

Este tipo de sesión se forma cuando un LSR se encuentra a un salto de su vecino, por lo que se dice que está directamente conectado, para establecer la adyacencia el LSR envía mensajes de *hello* mediante UDP a la dirección de multidifusión, un LSR vecino puede responder al mensaje *hello* del enlace, lo que permite que los dos enrutadores establezcan una sesión LDP.

Para iniciar una sesión LDP entre dos enrutadores, los *router* determinaran cuál de los dos tendrá un rol activo o pasivo, el *router* que toma el rol activo establece la sesión LDP he inicia la negociación de los parámetros LDP. Para determinar el rol activo cada equipo compara la dirección IP más alta dentro de sus interfaces y el mayor será el encargado de establecer la sesión. Después que la conexión LDP sea establecida, los LSR negocian los parámetros de sesión, incluyendo el método de distribución que usaran. actualmente existen dos métodos que son:

- *Downstream Unsolicited*: cada equipo LSR anuncia asignaciones de etiqueta a sus vecinos sin que ellos se lo pidan, recibiendo las etiquetas de los equipos adyacentes para construir el LSP o el camino basado en etiquetas para llegar a un destino (De Ghein, 2007).

- *Downstream on Demand: los* LSR configurados de este modo anuncia las etiquetas a sus equipos vecinos cuando le son solicitados lo que significa que los LSR van a crear las etiquetas y serán enviadas cuando un enrutador las necesite para crear la ruta por medio de etiquetas (De Ghein, 2007).

2.4.2. Sesiones LDP no conectadas directamente

Se producen cuando los LSR se encuentran a más de un salto de su vecino. para crear la sesión LDP, el LSR envía un mensaje de saludo de tipo unicast mediante UDP dirigido específicamente a ese LSR.

2.5. Etiquetas Especiales en MPLS

En MPLS se encuentran etiquetas reservadas para definir una función especial, las cuales se encuentran en el intervalo de 0 a 15 se detallan a continuación.

- Etiqueta 0 - *Explicit Null:* se envía al último LSR para informar que se necesita mantener el campo EXP intacto del paquete encapsulado de MPLS, de lo contrario se aplicará la técnica *Penultimate Hop Popping*, lo que significa que un LSR anterior removerá la etiqueta MPLS perdiendo los valores de Qos que estuvieran configurados.

- Etiqueta 1- *Router Alert Label*: se asigna cuando un paquete viaja por la red MPLS sin utilizar etiquetado especial por lo que será tratado como un paquete IP normal.

- Etiqueta 2 - *Explicit Null* IPv6: es aplicado para los paquetes IPv6 y su función es el mismo de los paquetes con etiqueta 0.

- Etiqueta 3- *Implicit Null*, (*Penultimate* hop *popping* (PHP)): indica al siguiente LSR que el paquete saldrá de la red MPLS y todas las etiquetas tendrán que ser removidas para que pueda ser tratado como un paquete IP, este proceso se realiza en el penúltimo equipo.

- Etiquetas 4-12: actualmente este rango de etiquetas se encuentra reservadas por la IETF aún no se les da un propósito en especial.

- Etiqueta 13-14: utilizados para detección de fallas y monitoreo de rendimiento.

- Etiqueta 15: se encuentra es estado de reserva.

- Unknown Label: cuando se recibe un paquete etiquetado que no conoce de forma local ni de forma remota, el LSR puede enviarlo o descartar el paquete.

3. ARQUITECTURA MPLS VPN DE CAPA 3

Es un tipo de conexión privada de modelo peer-to-peer que proporciona instancias de enrutamiento únicas dentro de un enrutador, aislando el tráfico de los clientes en una tabla de enrutamiento exclusiva con el objetivo en dar conectividad a un conjunto de sitios que se encuentran en diferentes zonas geográficas, la ventaja de este servicio es crear un circuito virtual único donde el cliente pueda observar solamente sus prefijos que conforma su tráfico IP.

Las VPN de capa 3 configurada dentro de MPLS utilizan el protocolo de enrutamiento BGP para crear una vecindad *overlay* entre los enrutadores de borde para dar transporte al tráfico que se relaciona con la VPN, lo que permite la creación de redes WAN de manera escalable y seguras (Cisco, 2019).

Las conexiones VPN permiten utilizar la infraestructura compartida de un proveedor de servicios para implementar sus redes privadas, existen dos modelos en las cuales se pueden establecer.

- Modelo Overlay: es el modelo donde el proveedor de servicios proporciona enlaces virtuales punto a punto entre los sitios de los clientes.

- Modelo Peer to Peer: se da cuando el proveedor de servicios participa en los procesos de enrutamiento de los clientes.

En la arquitectura MPLS VPN se implementan las características de ambos modelos Overlay y Peer to Peer además proporciona un entorno de conmutación de datos y eficiente basada en la etiqueta MPLS (Cisco, 2019).

Los principales componentes de la arquitectura MPLS VPN son las siguientes.

- Red del cliente: es dominio de red administrado por el propio cliente.

- Enrutador CE: son los equipos dentro de la red del cliente que proporciona conexión hacia la red del ISP.

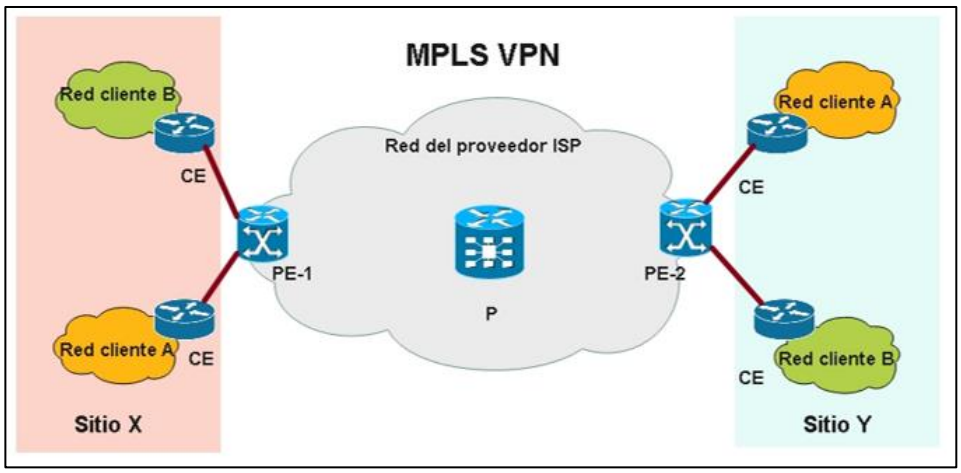
- Red del proveedor: es la red que consta de equipos de alta capacidad de procesamiento para realizar el enrutamiento de tráfico entre los sitios.

- Enrutador PE: es el equipo encargado de aislar el tráfico individual de cada cliente, asigna una tabla virtual de enrutamiento independiente hasta reenviar el tráfico entrante hacia otro equipo PE y lograr la interconexión entre sitios.

- Enrutador P: estos equipos conforman el núcleo de la red del proveedor ISP, proporcionan transporte de datos a través de la red troncal y no tienen clientes conectados a ellos.

Figura 32.

Topología MPLS VPN



Nota. Ejemplo de topología MPLS VPN. Elaboración propia, realizado con Edraw Max.

En una implementación de MPLS VPN desde la perspectiva de un enrutador CE, envía solo actualizaciones IPv4 hacia el enrutador PE, el CE no necesita ninguna configuración específica para que pueda ser parte del dominio VPN MPLS, el único requisito es tener un protocolo de enrutamiento estático o dinámico para intercambiar información de enrutamiento IPv4 con el PE conectado, en cambio el PE realizara múltiples funciones entre las cuales será poder aislar el tráfico en tablas de enrutamiento independientes para luego enviarlo a través del dominio MPLS hasta llegar a otro enrutador PE de borde y llegar finalmente hacia el equipo CE dentro de la red del cliente, todo este proceso es transparente para dicho cliente (Cisco 2019).

Los enrutadores P son responsables únicamente de la conmutación de las etiquetas, no llevan rutas VPN y no participan en el proceso VPN MPLS, los encargados de intercambiar rutas IPv4 provenientes de los CE son los equipos de borde PE, por lo que es necesario habilitar el protocolo de puerta de enlace

fronteriza multiprotocolo MP-BGP que se configura entre los equipos PE para el transporte de las rutas de los clientes, no es más que implementar BGP entre PEs sobre una red MPLS.

3.1. Tablas VRF

En las implementaciones de VPN MPLS los enrutadores PE aíslan el enrutamiento de un enlace de los demás clientes que estén conectados en el mismo equipo de borde, cada cliente se le asigna una tabla de enrutamiento independiente llamadas tablas VRF (*virtual routing forwarding table*), para separar los prefijos propios del ISP y de los diferentes clientes que contraten este servicio.

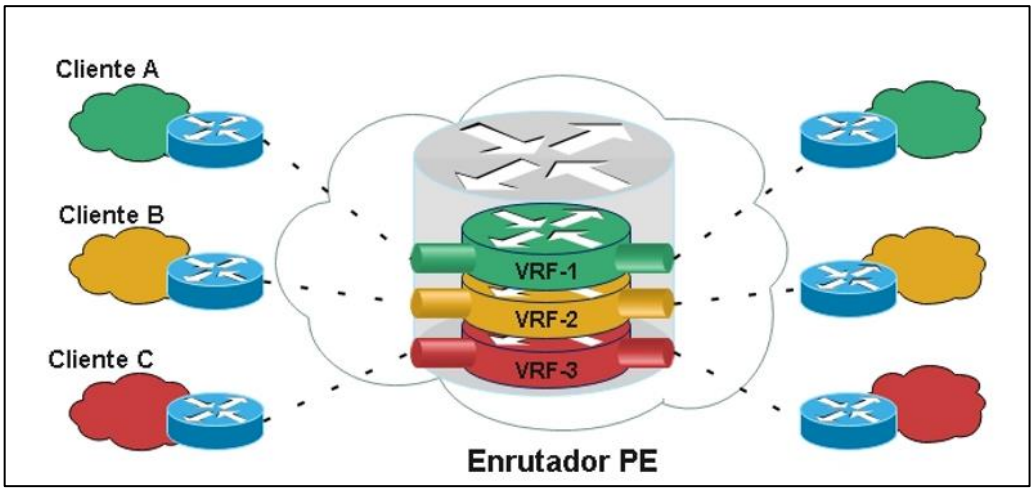
El enrutamiento de la red troncal se realiza mediante un proceso de enrutamiento sobre tablas globales, en esta tabla global se conocerán todos los prefijos que utiliza el ISP para la interconexión de la red, por lo tanto, solo los equipos PE serán los encargados de separar todas las tablas de enrutamiento debido que los equipos P no logran conocer las rutas VPN de los clientes y se limitarán solo el proceso de conmutación de etiquetas (Cisco, 2012).

Una instancia VRF pueden ser utilizados para un solo sitio VPN o para muchos sitios conectados al mismo enrutador de borde PE, solo si estos sitios comparten los mismos requisitos de conectividad. La estructura de datos de las tablas VRF abarca una tabla de enrutamiento IP que es idéntica a las tablas FIB dentro de Cisco IOS lo que soporta el mismo conjunto de mecanismos ya conocidos para el reenvío de paquetes, además ejecutan diferentes especificaciones como pueden ser políticas de calidad de servicio, ingeniería de tráfico, entre otros por cada instancia VRF (Cisco, 2012).

Para explicar este concepto se apoya en la Figura 33, donde el equipo PE se divide de forma virtual para cada cliente, de esta forma se estará intercambiando información dentro de sus propios prefijos sin llegar a mezclase con los de otros clientes, creando instancias VRF que otorga enlaces privados de forma virtual.

Figura 33.

VRF



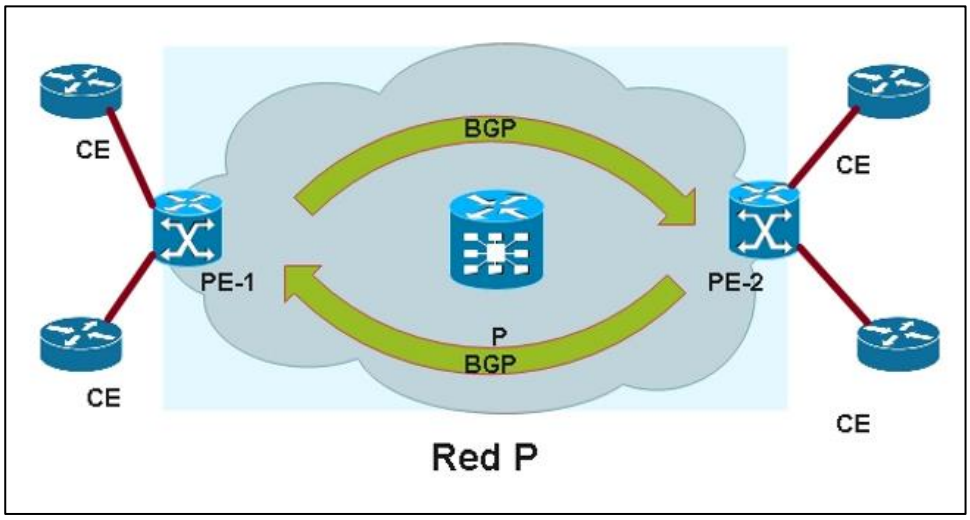
Nota. Ejemplo de tablas VRF. Elaboración propia, realizado con Edraw Max.

Si bien las tablas de enrutamiento virtual brindan aislamiento entre los clientes, los datos de estas tablas de enrutamiento aún deben intercambiarse entre enrutadores PE para permitir la transferencia de datos entre sitios. Por lo tanto, se necesita un protocolo de enrutamiento que transporte todas las rutas de los clientes a través de la red P manteniendo los espacios de direcciones de los clientes individuales.

La mejor solución al problema de la propagación de la ruta de los prefijos de los clientes es ejecutar un único protocolo de enrutamiento por instancia entre los enrutadores PE que intercambiarán todas las rutas del cliente sin la participación de los enrutadores P, como bien se mencionó anteriormente se necesita utilizar el protocolo BGP para transportar rutas de los clientes directamente entre los enrutadores PE.

Figura 34.

Topología P



Nota. Ejemplo de topología P. Elaboración propia, realizado con Edraw Max.

3.2. Distinguidores de rutas RD

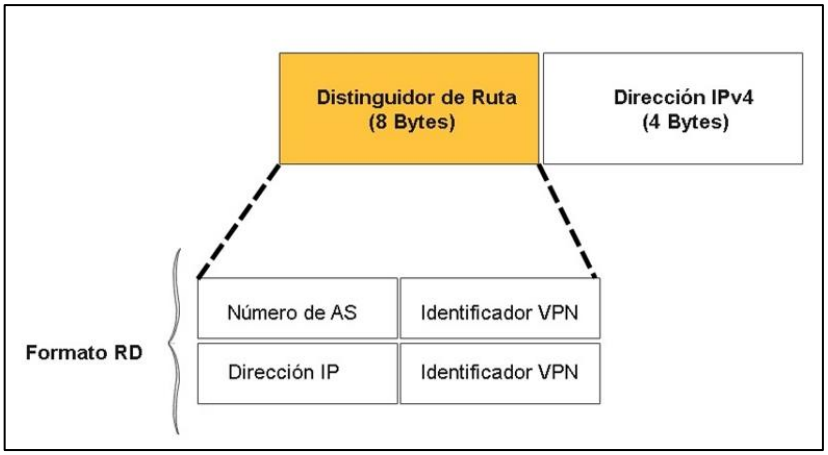
Cuando los prefijos son transportados a través de la red P los protocolos de transporte deberán ser capaces de distinguir los prefijos idénticos pero de cliente distinto, por ejemplo el cliente A estará utilizando el prefijo 172.32.10.0/24 además el cliente B decide utilizar el mismo prefijo para otro sitio, para realizar esta distinción se apoyan en los distinguidores de rutas no es más que agregar

una nueva cabecera de 64 bits a la dirección IPv4 tradicional de 32 bits, al momento de agregarla a una VRF esta dirección se convierte en una dirección de 96 bits que se transportan entre los enrutadores PE dentro del dominio MPLS, por lo tanto, se configura un RD único por VRF en el equipo PE, la dirección resultante, se denomina dirección VPNv4 (Cisco, 2012).

Las direcciones VPNv4 se intercambian solo entre enrutadores PE nunca es utilizada entre enrutadores CPE, entre los enrutadores PE el protocolo BGP es principalmente utilizado debido a que admite el intercambio de prefijos IPv4 tradicionales y prefijos VPNv4 por lo tanto una sesión BGP entre PE's se denomina sesión de protocolo de puerta de enlace de borde multiprotocolo MP-BGP (*Multiprotocol border gateway protocol*). Los RD pueden tener dos formatos, el primero es utilizar el sistema autónomo propio del proveedor de servicio seguido por el identificador VPN o utilizar una dirección IP seguido del identificador VPN (Cisco, 2012).

Figura 35.

Formato de encabezado RD



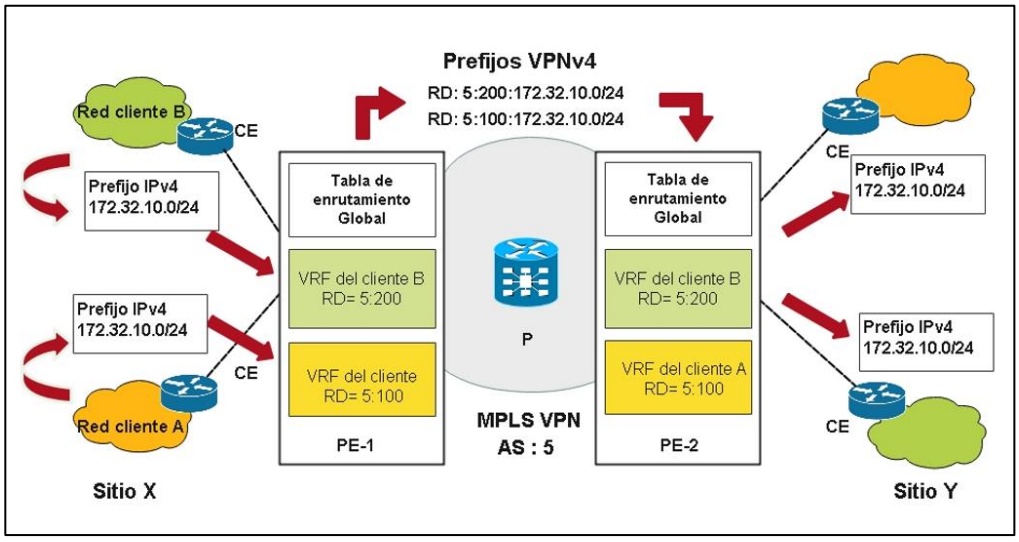
Nota. Ejemplo de encabezado RD. Elaboración propia, realizado con Edraw Max.

Se cuenta con el ejemplo de la Figura 36 donde la red del cliente A desea propagar el prefijo 172.32.10.0/24 al igual que el cliente B, el enrutador PE-1 recibe ambos prefijos de diferentes clientes, incluso se puede dar que existan varios clientes deseando compartir el mismo prefijo, el prefijo se hace único anteponiendo los distinguidores de ruta a los prefijos IPv4. Para el ejemplo se toman los RD con el formato del sistema autónomo que corresponde al número 5 y el identificador del prefijo con un valor asignado por el administrador quedando de la siguiente manera RD: 5:100 y 5:200 antes de ser propagada como una dirección VPNv4 por el equipo PE.

El protocolo utilizado para intercambiar estas rutas VPNv4 entre los equipos PE es BGP, debido que es capaz de transportar estos prefijos de 96 bits además de otras familias de direccionamiento.

Figura 36.

Implementación de RD



Nota. Ejemplo de implementación RD. Elaboración propia, realizado con Edraw Max.

El proceso de envío de rutas a través de una red MPLS VPN se menciona a continuación con el siguiente orden.

- Cuando un prefijo es enviado a través de un equipo CPE, se envía una actualización de enrutamiento IPv4 al enrutador PE.

- El enrutador PE antepone un RD de 64 bits dentro de la actualización de enrutamiento IPv4, lo que da resultado en un prefijo VPNv4 de 96 bits.

- Los prefijos VPNv4 se propagan a través de la red mediante una sesión MP-BGP hacia otros enrutadores PE.

- El último enrutador PE antes de salir de la red MPLS remueven el RD del prefijo VPNv4, dejando solamente el prefijo IPv4 como fue enviado al inicio del proceso, el valor del RD se utiliza para coincidir la tabla de enrutamiento virtual adecuada.

- Se finaliza la comunicación entre sitios reenviando a otros equipos CPE una actualización de enrutamiento IPv4.

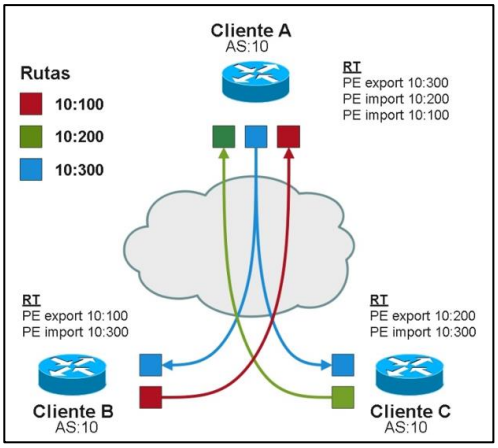
3.3. Destinos de Ruta RT

Los *router target* o destinos de ruta, se introducen dentro de la arquitectura de MPLS VPN con el objetivo de permitir la participación de un sitio en más de una VPN, además cuentan con valores adicionales que identifican las rutas aprendidas de algún sitio en particular hacia una VPN. Estos Identificadores de rutas se implementan mediante el uso de comunidades BGP extendidas de 16 bits que agrega valores de una lista de atributos y los asocia con los paquetes VPNv4 (Cisco, 2012).

Al momento de enviar un prefijo los RT se adjuntan a las rutas de los clientes en el momento que el equipo PE los convierte de una ruta IPv4 a una ruta VPNv4 los RT adjuntos a la ruta se denomina exportación de RT y se configuran para cada tabla de enrutamiento virtual en el enrutador PE, los RT de exportación identifican un conjunto de VPN a las que pertenecen los sitios asociados con la tabla de enrutamiento virtual. Cuando las rutas VPNv4 se propagan hacia los demás enrutadores PE, esos enrutadores deben de seleccionar las rutas las cuales se añadirán a las tablas de enrutamiento virtual local, este proceso de denomina importación de RT debido que cada tabla de enrutamiento virtual en un enrutador PE, puede tener una cantidad de RT de importación configurados que identifican el conjunto de VPN desde el cual la tabla de enrutamiento virtual acepta las rutas aprendidas, en otras palabras un solo prefijo se puede asociar a más de un destino cuando se propaga a través de la red MPLS VPN, el RT permite asociarse a sitios que pueden ser miembros a más de una VPN (Cisco, 2012).

Figura 37.

Ejemplo de RT



Nota. La imagen muestra las rutas RT. Elaboración propia, realizado con Edraw Max.

Para el siguiente ejemplo con la misma topología que se presentó para explicar los RD de la Figura 36 ahora se agregan los RT para explicar cada proceso en el cual se involucra el reenvío de prefijos.

- Como primer paso se muestra el reenvío de los paquetes desde la perspectiva de los equipos CPE, el cual se cuenta con el prefijo 172.32.10.0/24 que se origina desde la red del cliente y es recibido por CPE 1-A el cual es parte de la VRF del cliente A en el PE-1, luego se cuenta con el segundo prefijo 192.168.10.0/24 recibida por CPE 1-B el cual es parte de la VRF del cliente B en PE-1.

- Para CPE 1-A, el PE-1 se asocia un valor de RD de 5:100 y un valor de exportación de 5:100, de igual forma para CPE 1-B se asocia un valor de RD de 5:200 y un RT de 5:200, tal como fue configurado para cada VRF.

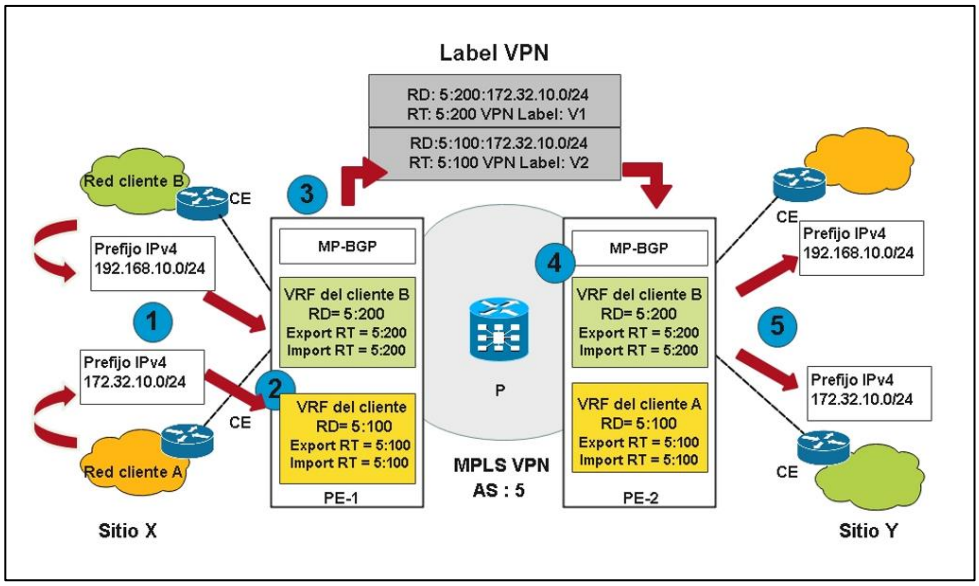
- Las rutas que fueron recibidas por el equipo PE-1 provenientes de los CPE se encuentran listas para ser redistribuidas por el proceso MP-BGP, donde los prefijos 172.32.10.0/24 y 192.168.10.0/24 se agregan una nueva cabecera con los valores de RD y RT correspondiente para luego enviar los prefijos VPNv4 como actualizaciones hacia los demás PE por medio de la etiqueta VPN.

- La actualización de MP-BGP la recibe el enrutador PE-2 y las rutas se almacenan en las tablas correspondientes para cada cliente según la etiqueta VPN asociada con el valor de RT de importación, cabe aclarar que en cada VRF de los clientes se necesita contar con RT de importación y exportación para que exista comunicación de forma bidireccional.

- Las rutas finales que son recibidas por PE-2 se redistribuyen en los procesos de enrutamiento entre PE y CPE hasta llegar a la red destino del cliente.

Figura 38.

Implementación de RT



Nota. La imagen muestra la implementación RT. Elaboración propia, realizado con Edraw Max.

3.4. Requerimientos de configuración

La característica principal de las VPN MPLS de capa 3 es otorgar tablas de enrutamiento aisladas de los demás clientes sobre una misma ruta VPN, estas tablas VRF independientes admite exactamente el mismo conjunto de mecanismos que las tablas de enrutamiento estándar como lo pueden ser filtros, selección de rutas entre protocolos entre otros.

La tabla de reenvío FIB se crea a partir de la tabla de enrutamiento de cada VRF, estas tablas se utilizan para reenviar todos los paquetes que se reciben a través de las interfaces asociadas a la VRF, toda interfaz que admita CEF se puede asociar a una VRF ya sea una interfaz física, subinterfaz o una interfaz lógica, no hay límite para la cantidad de interfaces que se pueden asociar a una VRF, sin embargo, cada interfaz solamente se puede asignar a un solo VRF (Cisco, 2012).

La configuración de una tabla VRF y el inicio de la implementación de un servicio MPLS VPN de capa 3 para un cliente consta de los siguientes pasos de forma obligatoria.

- Crear una nueva instancia VRF

- Asignar un RD para cada VRF creada en un enrutador PE, el mismo RD puede usarse en múltiples enrutadores PE, según los requisitos de conectividad del cliente.

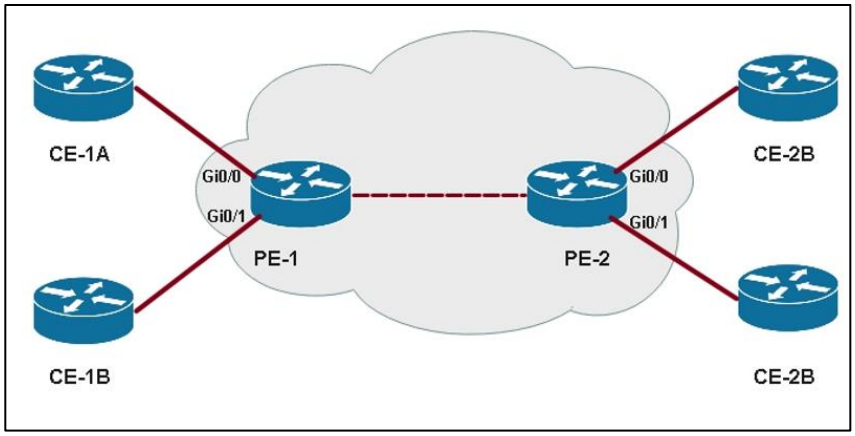
- Especificar los RT de importación y exportación para cada VRF.

- Asignar las interfaces hacia la VRF correspondiente.

Se ilustran el uso de los comandos de configuración de la siguiente topología básica entre dos enrutadores PE en una red con dos clientes VPN.

Figura 39.

Implementación de MPLS VPN de capa 3



Nota. La figura muestra la implementación MPLS VPN capa 3. Elaboración propia, realizado con Edraw Max.

Figura 40.

Comandos de configuración sobre equipo PE

PE-1

```
ip vrf CLIENTE-A
  rd  100:10
  route-targe export 100:10
  route-targe import 100:10
!
ip vrf CLIENTE-B
  rd  100:20
  route-targe export 100:20
  route-targe import 100:20
!
interface GigabitEthernet 0/0
  ip vrf forwarding CLIENTE-A
  ip address 10.10.10.1 255.255.255.252
!
interface GigabitEthernet 0/1
  ip vrf forwarding CLIENTE-B
  ip address 10.10.10.1 255.255.255.252
!
```

PE-2

```
ip vrf CLIENTE-A
  rd  100:10
  route-targe export 100:10
  route-targe import 100:10
!
ip vrf CLIENTE-B
  rd  100:20
  route-targe export 100:20
  route-targe import 100:20
!
interface GigabitEthernet 0/0
  ip vrf forwarding CLIENTE-A
  ip address 10.10.10.2 255.255.255.252
!
interface GigabitEthernet 0/1
  ip vrf forwarding CLIENTE-B
  ip address 10.10.10.2 255.255.255.252
!
```

Nota. Ejemplo de comandos de configuración. Elaboración propia, realizado con Cisco iOS.

4. ARQUITECTURA VPN DE CAPA 2

Es un tipo de conexión privada que proporciona conexión entre múltiples puntos sobre una red corporativa, es una forma de crear túneles punto a punto o multipunto donde se necesita mantener la VLAN y los protocolos de capa 2 de cada enlace de forma intacta, la clasificación de los paquetes entre el equipo de borde del ISP y los equipos del cliente se lleva a cabo sobre la capa 2 y el transporte en el núcleo del proveedor ISP se realiza sobre MPLS.

Esta técnica permite al cliente una extensión de su red LAN al estar en zonas geográficas distantes, es decir desde la perspectiva del cliente varios sitios pueden compartir el mismo segmento IP de capa 3 simulando estar directamente conectados sobre el mismo dominio de difusión, con la ventaja que se pueden omitir toda regla de ruteo para alcanzar el destino remoto ya que toda la transmisión se realiza mediante la capa 2 (Frausto, 2019).

4.1. Introducción a las VPN de capa 2

Las VPN de capa 2 comprende tres elementos principales:

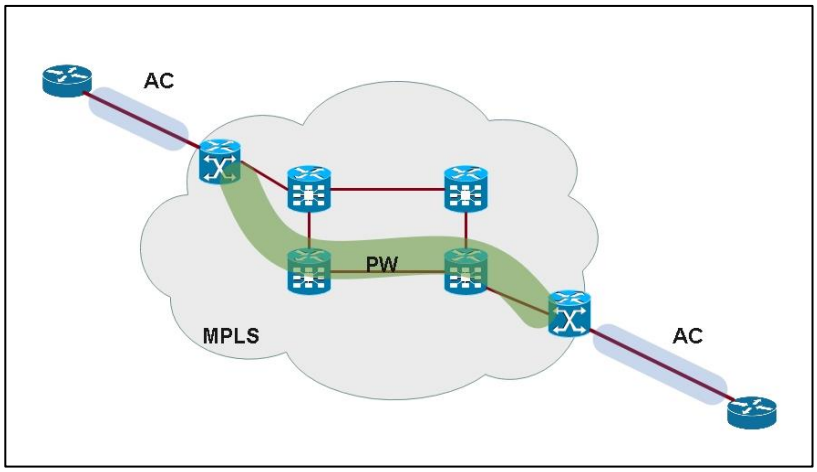
- Attachment circuit (AC): es un circuito físico o virtual directamente conectado al PE del ISP que provee conexión de un sitio en particular.

- Pseudowire (PW): conexión punto a punto a través de una red MPLS que simula el funcionamiento de un "cable transparente" que conecta dos Attachment circuit hacia dos enrutadores PE.

- Transport Infrastructor: red de transporte que pueda admitir cualquier topología de capa 2 (punto a punto, punto a multipunto o multipunto a multipunto) mediante MPLS o L2TP basada en IP.

Figura 41.

Arquitectura VPN de capa 2



Nota. En la imagen se observa el ejemplo de arquitectura VPN. Elaboración propia, realizado con Edraw Max.

Las tramas que ingresan sobre la interfaz de entrada del enrutador PE son tramas de capa 2, al momento de entrar a la red MPLS se encapsulan y se envían a través de la red troncal utilizando dos etiquetas. La primera etiqueta se utiliza para propagar la trama del PE de entrada hacia el PE de salida, la segunda etiqueta es utilizado por el PE de salida para reenviarlo hacia la interfaz correcta de salida, la etiqueta superior se llama etiqueta de túnel, este nombre indica que su uso es para tunelizar el paquete a través de la red troncal MPLS hasta el enrutador PE de salida, la segunda etiqueta se llama etiqueta VC, su nombre indica que su uso es para identificar circuitos individuales dentro de un túnel (Cisco, 2012).

Las VPN de capa 2 se pueden agrupar en tres modelos principales:

- Local Switching: se utiliza en enlaces conectados directamente lo que permite el reenvío de las tramas entre dos interfaces del mismo equipo.

- MPLS Core: utiliza la infraestructura de las redes MPLS como transporte para levantar túneles de diferentes topologías como pueden ser VPWS Y VPLS.

- IP Core: es una solución a nivel empresarial que utiliza el protocolo L2TP para establecer túneles de capa 2 sobre una infraestructura IP cuando las empresas no cuentan con una red MPLS.

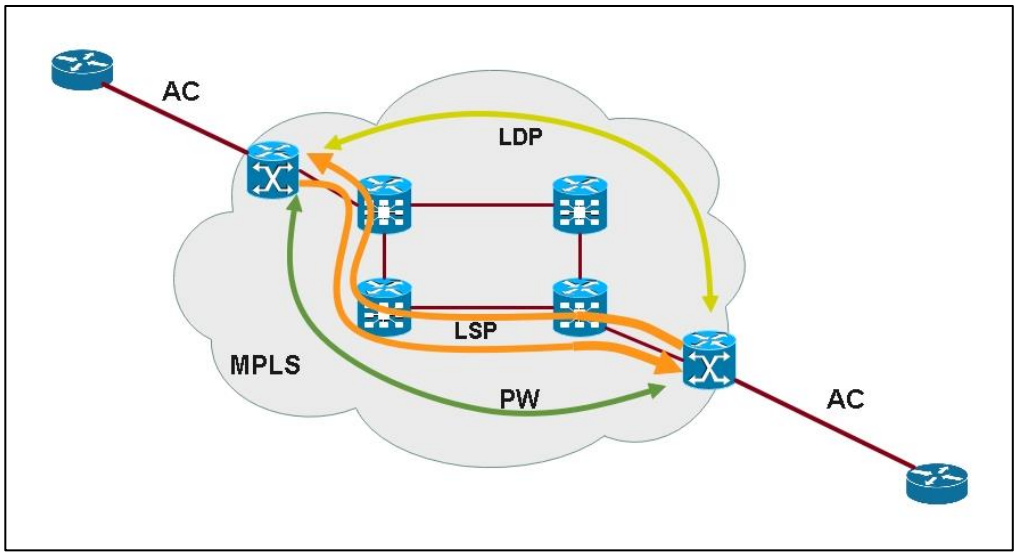
Las VPN de capa 2 permite tanto a los proveedores de servicios como a las empresas construir una única infraestructura sobre MPLS para el uso de transporte de los servicios tradicionales, por lo que pueden aprovechar el dominio de la capa 2 extendido para optimizar el envío de datos lo que permite una gran cantidad de extensiones de alta disponibilidad, además de proporcionar una separación lógica entre los dominios del cliente y el proveedor, que incluye segmentación, políticas de QoS entre otros (Cisco, 2012).

En MPLS, se requiere un protocolo entre los PEs que puedan intercambiar la información de la etiqueta VC, por lo que se utiliza el protocolo de distribución de etiquetas LDP para este propósito. Se establece una sesión LDP multisalto dirigida entre los enrutadores PE. El enrutador PE de salida envía un mensaje LDP en el que indica el valor de etiqueta que se utilizará para un FEC en particular, luego el enrutador PE de ingreso usa ese valor de etiqueta como la segunda etiqueta de la pila para enviar las tramas por el FEC indicado (Cisco, 2012).

La figura 42 muestra una sesión LDP multisalto dirigida entre los enrutadores PE de entrada y salida que se usa para intercambiar la etiqueta de VC. Cualquier par de enrutadores PE de entrada y salida necesitará una sesión LDP de este tipo.

Figura 42.

Arquitectura VPN de capa 2 sobre MPLS



Nota. La imagen muestra ejemplo de arquitectura VPN sobre MPLS. Elaboración propia, realizado con Edraw Max.

4.2. Redes Metro-Ethernet

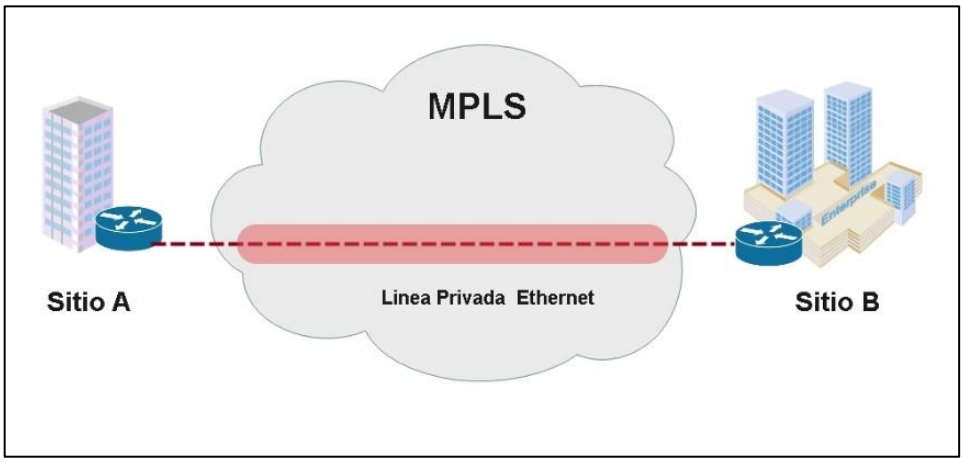
Es una red de transporte de capa 2 que ofrece servicios de conectividad punto a punto o multipunto permitiendo consolidar una amplia gama de servicios donde se incluye video, internet, servicios de datos y telefonía que están enfocadas a clientes corporativos y servicios residenciales cuyo transporte está basado en interfaces ethernet (Cisco, 2012).

El metro ethernet fórum define tres tipos de servicios principales y define los atributos de conectividad con el usuario final.

- E-line: es el servicio que conecta dos puertos ethernet del cliente a través de una interfaz WAN, donde solo puede comunicarse entre sí, estableciendo un enlace punto a punto.

Figura 43.

Topología E-line



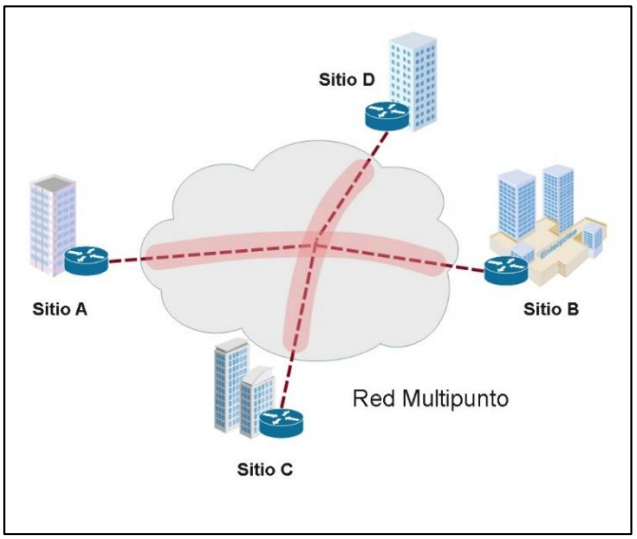
Nota. La figura muestra ejemplo de topología E-line. Elaboración propia, realizado con Edraw Max.

- E-LAN: servicio multipunto que conecta un conjunto de puntos finales de los clientes, lo que da una apariencia de una red LAN extendida conectando varios sitios bajo un mismo segmento de red, este tipo de servicio de LAN es transparente, se denomina VPN de capa 2 multipunto, o más conocido por el estándar como Servicios de LAN privada virtual VPLS.

Se utiliza cuando se necesita conservar la dirección de Mac-addres y todos los parámetros de capa 2 de un sitio remoto sin que se elimine de la red MPLS, lo que estará simulando un equipo switch conectado a varios sitios sobre sus interfaces hacia un conmutador central, las aplicaciones que se benefician de esta topología son los servicios de multidifusión y servicios de voz sobre IP (Cisco, 2012).

Figura 44.

Topología E-LAN

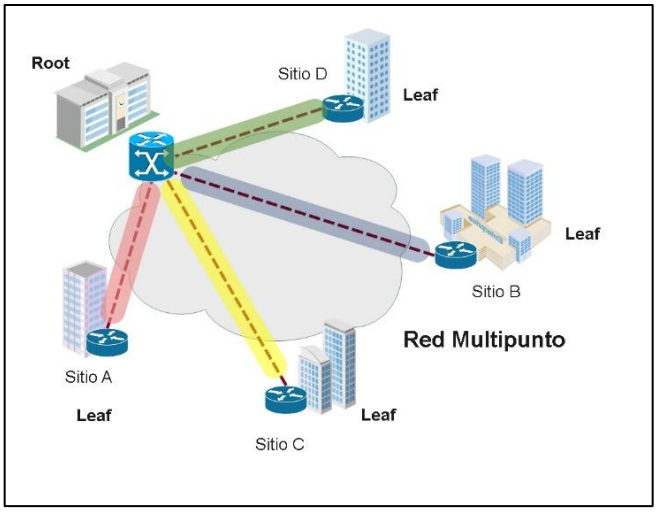


Nota. Ejemplo de topología E-LAN. Elaboración propia, realizado con Edraw Max.

- E-Tree: es un servicio multipunto que conecta un conjunto de sitios finales hacia un concentrador denominado root, cada sitio individual se designa como root o leaf, un sitio root puede comunicarse con cualquier leaf, pero un sitio leaf solo puede comunicarse con un solo sitio root, lo que proporciona una separación lógica entre servicios hacia una sola instancia con el proveedor de servicios.

Figura 45.

Topología E-Tree



Nota. Ejemplo de topología E-Tree. Elaboración propia, realizado con Edraw Max.

Todos estos servicios proporcionan características estándar tales como ancho de banda, filtrado y multiplexación, lo que permite a los clientes comparar ofertas de servicios y facilitar acuerdos de nivel de servicio (SLA).

4.3. Servicio de transporte VPWS

Las VPN de capa 2 (VPWS) emplean servicios de capa 2 sobre MPLS para construir una topología de conexiones punto a punto que conectan los sitios de los clientes finales en una VPN, el enlace virtual que une ambos extremos es transparente a los mensajes SPT, BPDU, VTP y otros mensajes de control sin el aprendizaje de direcciones MAC.

Para crear el circuito del túnel puede ser por puerto ethernet, subinterfaz o 802.1Q VLAN, debido a cada circuito LDP emplea un tipo de etiqueta VC

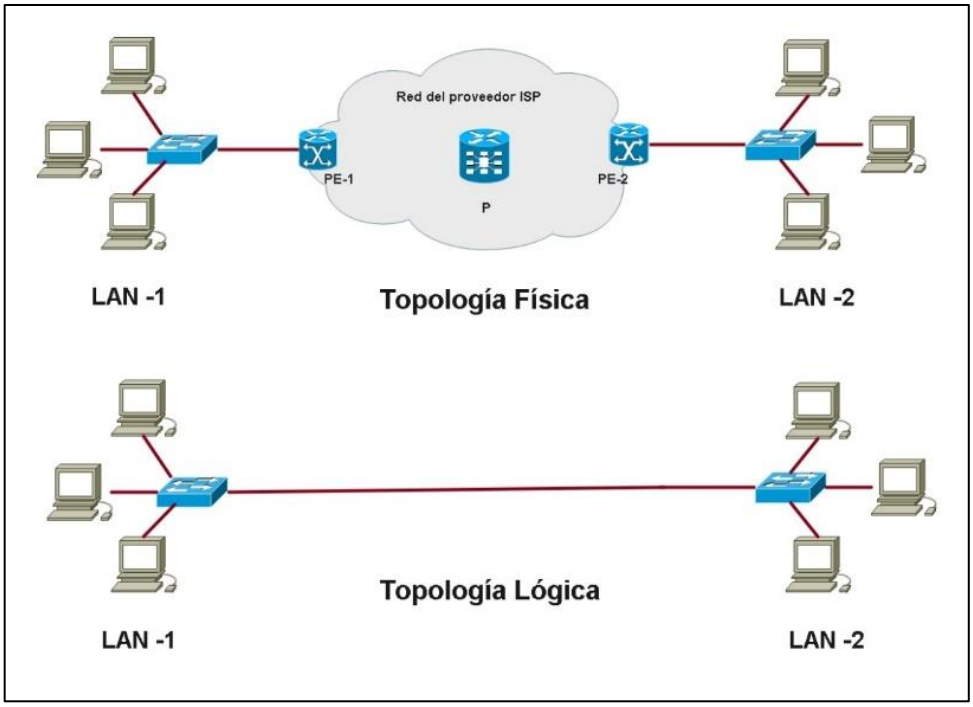
diferente a través de las sesiones LDP de destino. El VC tipo 5 (VC Label 0x0005), se usa para el modo de puerto Ethernet y el de tipo 4 (VC Label 0x0004) es utilizado para el modo ethernet VLAN (Cisco, 2012).

En el modo de puerto Ethernet, ambos extremos del Pseudowire están conectados a puertos ethernet, desde el puerto se crea el túnel a través de un PW mediante la conmutación local que envía los paquetes de un circuito de conexión a otro conectado al mismo PE, en este modo, el PW siempre es una conexión virtual de tipo 5, en el PE de ingreso el ISP pasa los paquetes al punto de inicio del PW y agrega las etiquetas MPLS a los paquetes y los envía a través de PW, en este modo un encabezado de VLAN puede o no estar presente en la trama, en cualquier caso, el enrutador PE lo transporta de forma transparente, esto permite que un enlace troncal de Ethernet se transmita a través de un solo PW (Cisco, 2012).

En el modo VLAN cada enlace se puede configurar como una conexión VPN de Capa 2 separada, utilizando una conexión virtual tipo 4. En el PE de entrada se reciben los paquetes originados por el cliente y el proveedor de servicios quita la etiqueta VLAN del extremo para colocar los paquetes en el PW para luego transmitirse hacia el PE de salida con solo las cabeceras de etiquetado MPLS ya finalizada la transmisión el PE realiza el proceso de insertar de nuevo la etiqueta VLAN en la pila de protocolos antes de enviar el paquete final hacia la red del cliente (Cisco, 2012).

Figura 46.

Túnel de transporte VPWS



Nota. La imagen detalla el ejemplo de túnel de transporte VPWS. Elaboración propia, realizado con Edraw Max.

El funcionamiento de los túneles VPWS se pueden dividir de dos modos:

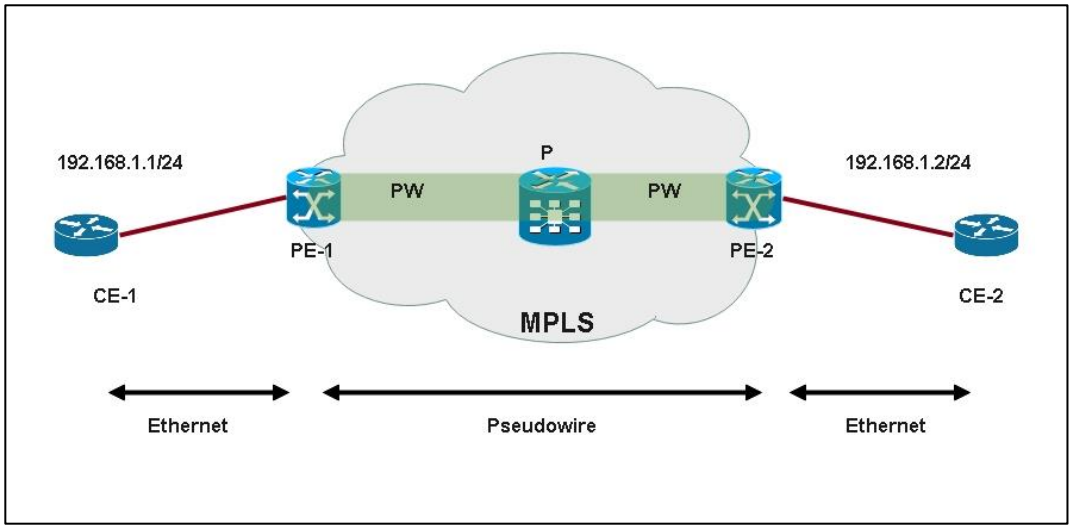
- Bridged interworking: las tramas ethernet se extraen del attachment circuit y es enviado a través del PW si no son tramas ethernet son descartados, en el caso de una VLAN, la etiqueta de VLAN se elimina dejando una trama sin etiquetar. Esta funcionalidad se implementa configurando el comando interworking ethernet en el modo de configuración del pseudowire (Cisco, 2012).

- Routed interworking: los paquetes IP se extraen del attachment circuit y se envían a través del PW, las tramas recibidas se descartan si no contienen los paquetes IPv4. Esta funcionalidad se implementa configurando el comando interworking ip en el modo de configuración del pseudowire (Cisco, 2012).

En el siguiente ejemplo se explica los pasos a seguir para establecer una conexión de túnel de capa 2 y se detallan los comandos para establecer la conexión mediante el modo de funcionamiento Ethernet, previamente configurada la red MPLS.

Figura 47.

Ejemplo sobre el túnel VPWS



Nota. La imagen muestra como es el túnel VPWS. Elaboración propia, realizado con Edraw Max.

Figura 48.

Configuración sobre los equipos CE

```
!                                          !
hostname CE-1                              hostname CE-2
!                                          !
interface Gi0/0                            interface Gi0/0
ip address 192.168.1.1 225.255.255.0       ip address 192.168.1.2 225.255.255.0
no shut                                    no shut
!                                          !
```

Nota. Ejemplo de configuración equipos CE. Elaboración propia, realizado con Cisco iOS.

Figura 49.

Configuración sobre los equipos PE

```
!                                          !
hostname PE-1                              hostname PE-2
!                                          !
pseudowire-class Ether-1                   pseudowire-class Ether-1
 encapsulation mpls                         encapsulation mpls
 interworking ethernet                      interworking ethernet
!                                          !
interface Gi0/1                            interface Gi0/1
 no ip address                              no ip address
 xconnect 100.10.10.1  100 encapsulation    xconnect 100.10.10.2  100 encapsulation
 mpls pw-class Ether-1                      mpls pw-class Ether-1
!                                          !
```

Nota. Ejemplo de configuración equipos PE. Elaboración propia, realizado con Cisco iOS.

- El primer paso es definir el Pseudowire en los enrutadores PE, en este paso se define el PW denominado Ether-1 en ambos enrutadores PE-1 y PE-2, se necesita que los parámetros de configuración sean los mismos en ambos equipos PE para permitir el establecimiento del PW, en la Figura 49 se muestra que se utilizará la encapsulación sobre MPLS y el modo de funcionamiento Bridged interworking.

- En el segundo paso, se utiliza la instrucción xconnect para definir el circuito virtual que da el transporte de las tramas de capa 2 de CE-1 a CE-2 y viceversa, asociándola con el PW definida en el paso 1.

4.4. Servicio de transporte AToM

AToM permite el transporte de tramas de capa 2 punto a punto a través de una infraestructura MPLS, esto permitirá a los proveedores de servicios conectar redes de capa 2 de clientes de forma transparente. Los túneles se construyen a partir de una ruta LSP de etiqueta conmutada entre los dos enrutadores PE participantes, en este proceso se asocia una etiqueta con el LSP ya existente que se denomina etiqueta del túnel, esta etiqueta identifica la pertenencia de las tramas de cada cliente que utiliza este servicio (Cisco, 2012).

Se necesita tener en cuenta los siguientes aspectos para que los túneles de capa 2 sean implementados de forma correcta.

- El comando xconnect se debe configurar en el equipo PE de ingreso, puede ser por un puerto ethernet o alguna subinterfaz, de forma alterna se puede utilizar el mecanismo de autodescubrimiento para detectar al vecino.

- El PE-1 inicia una sesión LDP dirigida a PE-2.

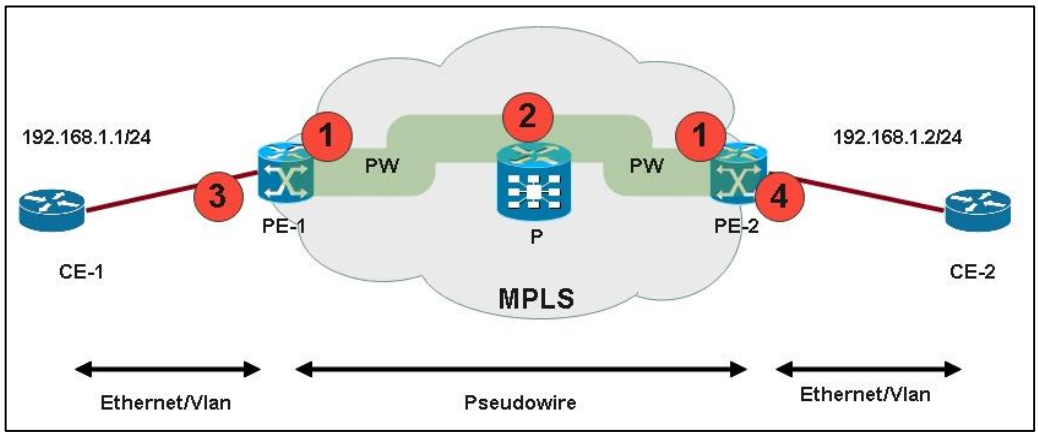
- El PE de ingreso asigna una etiqueta de circuito virtual VC y se asocia a un ID de VC, se debe de configurar el mismo valor de ID de VC en ambos extremos.

- En el PE de salida recibe la etiqueta VC y se asigna al VC ID configurado localmente.

- El otro extremo PE-2 repite el proceso de los pasos 1 a 4, que luego finaliza en PE-1 al recibir la etiqueta VC y asignarlo al ID configurado de forma local.

Figura 50.

Proceso de implementación de AToM



Fuente: Ejemplo de proceso implementación AToM. Elaboración propia, realizado con Edraw Max.

El proceso de reenvío requiere un Protocolo IGP en la red troncal MPLS ya que todos los enrutadores de la red tienen información de enrutamiento sobre cómo enviarse paquetes IP entre sí. LDP también se utiliza entre vecinos conectados directamente. Se asignan etiquetas locales a cada ruta derivada de IGP. Luego los valores de la etiqueta se propagan al vecino a través de la sesión LDP (Cisco, 2012).

El IGP junto con las sesiones LDP entre vecinos conectados directamente, establece rutas de conmutación de etiquetas (LSP), desde cualquier enrutador dentro de la red troncal a cualquier otro enrutado de la red. En la figura 50, se establece un LSP unidireccional entre los enrutadores PE de entrada y salida. La etiqueta del túnel se utiliza para propagar los paquetes a lo largo del LSP al enrutador PE de salida correcto (Cisco, 2012).

La figura 51 también muestra la sesión multihop LDP dirigida, que se utiliza para intercambiar la etiqueta VC entre los enrutadores PE de entrada y salida. Cualquier par de enrutadores PE de entrada-salida necesitará una sesión LDP de este tipo. En este ejemplo, el enrutador PE de salida asigna el valor de etiqueta 40. La etiqueta VC se anuncia al enrutador PE de entrada mediante la sesión LDP dirigida entre ellos (Cisco, 2012).

El enrutador PE de entrada ahora forma una pila de etiquetas. La etiqueta superior que corresponde a la del túnel tiene el valor 28 y se utiliza para guiar los paquetes al enrutador PE de salida. La segunda etiqueta, la etiqueta VC, tiene el valor 40 y es utilizada por el enrutador PE de salida para propagar los paquetes en la interfaz correcta. El enrutador PE de ingreso recibe una trama sobre la interfaz entrante. El paquete se asigna al túnel AToM a través de la red troncal. Por lo tanto, la trama se encapsula en MPLS utilizando la pila de etiquetas, con la etiqueta 28 como la etiqueta superior y la etiqueta 40 como la segunda etiqueta. Luego, el paquete se reenvía a lo largo del LSP (Cisco, 2012).

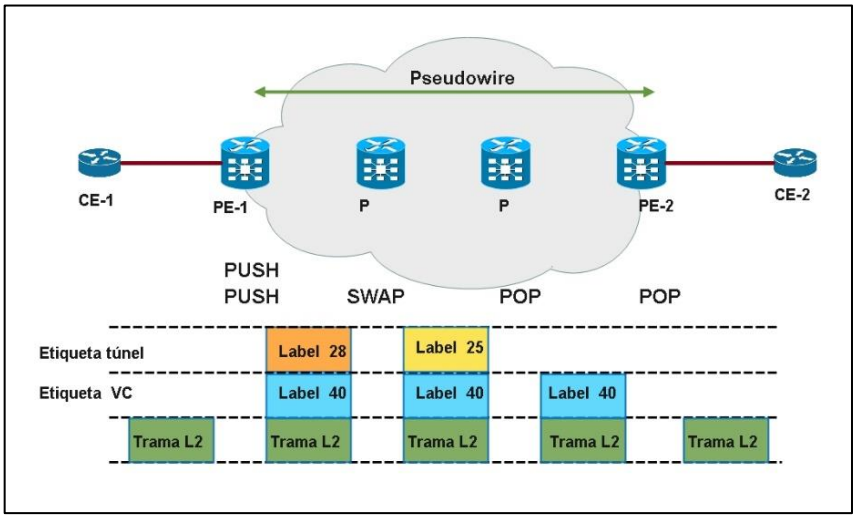
La etiqueta superior se utiliza para el intercambio de etiquetas y además cambia de valor en cada salto este intercambio de etiquetas da como resultado el valor de etiqueta 25. El enrutador P justo antes del enrutador de salida, el valor 25 de la etiqueta se remueve realizando un pop sobre la etiqueta, por lo tanto, esa etiqueta realiza el penúltimo salto (PHP). Se quita la etiqueta superior

y el paquete se propaga al enrutador PE de salida con el valor de etiqueta 40, la etiqueta VC, que ahora es la única etiqueta que se encuentra activa (Cisco, 2012).

Cuando el enrutador PE de salida recibe el paquete con el valor de etiqueta 40, ese valor de etiqueta le indica al enrutador PE que realice el proceso de des encapsulación del paquete y lo envíe al puerto de salida asociado.

Figura 51.

Proceso de reenvío de etiquetas AToM

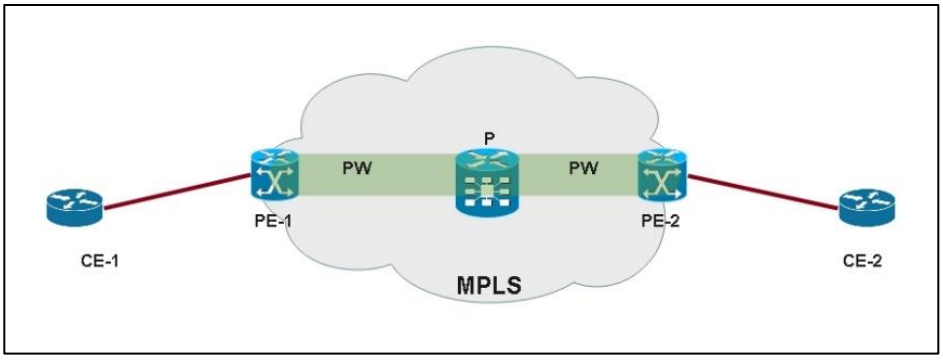


Nota. Ejemplo de proceso de reenvío AToM. Elaboración propia, realizado con Edraw Max.

Con el siguiente ejemplo se implementa AToM sobre una red MPLS, antes de configurarlo se habilita el transporte de tramas de capa 2 en ambos PE de los extremos, después asegurarse que el MTU sobre las interfaces sea la misma en ambas interfaces de los puntos finales.

Figura 52.

Ejemplo de configuración de AToM



Nota. La figura muestra la configuración AToM. Elaboración propia, realizado con Edraw Max.

Figura 53.

Configuración de AToM sobre equipos PE

```
!
hostname PE-1
!
interface Loopback 0
 ipv4 address 100.10.10.10 255.255.255.255
!
interface Giga 0/0/0/0.40 L2transport
 encapsulation dot1q 40
!
l2vpn
xconnect group eompls-group
 p2p eompls-p2p
  interface Giga0/0/0/0.40
  neighbor 200.20.20.20 pw-id 343
!
```

```
!
hostname PE-2
!
interface Loopback 0
 ipv4 address 200.20.20.20 255.255.255.255
!
interface Giga 0/0/0/0.40 L2transport
 encapsulation dot1q 40
!
l2vpn
xconnect group eompls-group
 p2p eompls-p2p
  interface Giga0/0/0/0.40
  neighbor 100.10.10.10 pw-id 343
!
```

Nota. La figura muestra como es una configuración AToM en equipos PE. Elaboración propia, realizado con Edraw Max.

- Como primer paso se asigna una dirección de loopback de longitud 32, esto se requiere para identificar el equipo dentro de la red y enlazar la etiqueta de VC.

- Se agrega el comando l2transport para habilitar el transporte de capa 2.

- Se levanta el PW utilizando los comandos xconnect group y p2p en el modo de configuración de VPN de capa 2.

- Se especifica el vecino con cual levantar el PW con el PE remoto en conjunto con el ID de VC.

4.5. Servicio de transporte VPLS

El servicio de VPLS es una VPN de capa 2 multipunto privada basada en Ethernet, lo que permite conectar sitios LAN de zonas geográficas distantes a través de una red troncal MPLS mediante Pseudowires, todas las redes LAN emulados estarán completamente separados de otros segmentos LAN. Cuando un cliente con diferentes sitios se conecta a una red troncal MPLS donde se implemente VPLS, parece como si todos los sitios estuvieran interconectados a través de un conmutador Ethernet virtual (Cisco, 2012).

Para cada VPLS los PE están completamente integrados con un PW, un PE que recibe una trama de otro PE puede identificar a que VPLS pertenece la trama, basándose en una etiqueta PW, en lo que respecta a cada cliente, una trama de Ethernet que se envía en la red del proveedor de servicios se entrega a los sitios correctos en función de la dirección MAC de destino. Es tarea de cada enrutador PE inspeccionar la dirección MAC de destino de cada trama que llega de un sitio de forma local y reenviarla al sitio de destino apropiado este sitio de

destino puede estar conectado al mismo PE en un puerto diferente o un PE remoto, si el sitio de destino está conectado al mismo PE, el PE conmuta localmente la trama al puerto correcto, si el sitio destino está conectado a un PE remoto el PE de ingreso debe reenviar la trama al PW apropiado hacia el PE remoto, esto significa que el PE de entrada necesita saber a qué PE de salida enviar la trama (Cisco, 2012).

Hay dos formas de lograr el envío de las tramas hacia el destino correcto uno es tener una señalización de plano de control para transportar información sobre direcciones MAC entre los PE; otra es tener un esquema basado en el aprendizaje de direcciones MAC, VPLS adopta el último enfoque al hacer que cada PE tome la responsabilidad de aprender qué PE remoto está asociado con una dirección MAC determinada, de esta manera, un PE de entrada simplemente necesita identificar qué tramas deben enviarse a los PE de salida, y los PE de salida se encargan de identificar a qué puertos locales reenviar el paquete. Al inspeccionar la dirección MAC de origen de la trama que llega a un puerto, ya sea un puerto local o un PW de un PE remoto, se crea una entrada correspondiente en la tabla de reenvío, el PE aprende a dónde enviar las tramas futuras con esa dirección MAC de destino (Cisco, 2012).

Los principales beneficios de VPLS son los siguientes:

- Es virtual porque varias instancias de este servicio comparten la misma infraestructura física.

- Es privado porque cada instancia del servicio es independiente de las demás.

- Es un servicio LAN porque emula la conectividad multipunto de capa 2 entre suscriptores.

Esta tecnología admite conexiones multipunto de capa 2 al agrupar una colección de Pseudowire concentrados en un enrutador PE sobre una interfaz de reenvío virtual (VFI), El VFI representa una extensión virtual del circuito físico conectado al PE, el VFI se asemeja a un conmutador que es capaz de aprender las direcciones MAC y reenvía el tráfico según su tabla de direcciones MAC. Un VPLS puede conectar varios PE en una sola VLAN, y por lo tanto está sujeto a restricciones de escalabilidad.

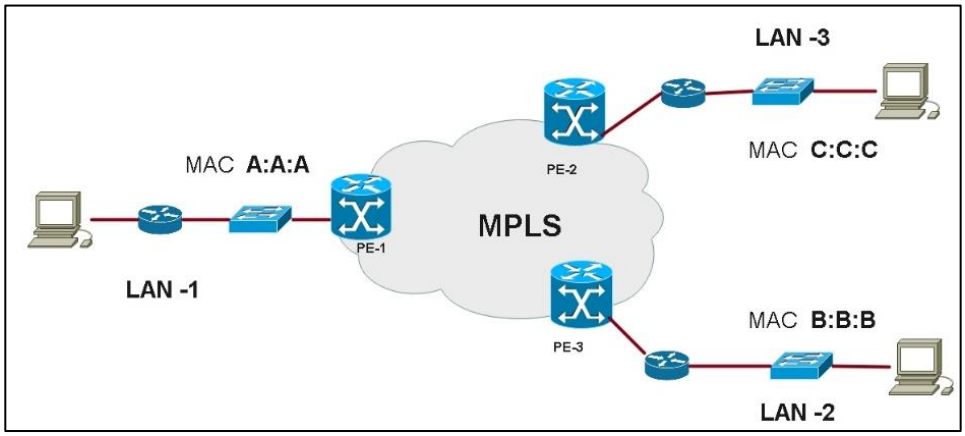
VPWS crea PW que emulan circuitos de capa 2. Una red de servicio de LAN privada virtual (VPLS), es similar a VPWS, pero proporciona reenvío de tráfico de punto a multipunto en contraste con el reenvío de tráfico de punto a punto de VPWS.

Para configurar un VPLS, primero se crea una instancia de reenvío virtual (VFI), en cada enrutador PE participante, el conjunto de todos los VFI formado por las interconexiones de los circuitos virtuales se denomina instancia VPLS es lo que forma el puente lógico sobre una red de conmutación de paquetes, por lo tanto todos los equipos PE usan la VFI para establecer un LSP de malla completa para todos los demás PEs sobre la misma instancia, esta red de malla completa permite que todos los equipos bajo la misma configuración de VPLS mantenga un solo dominio de transmisión, por lo tanto, cuando el PE reciba un paquete de difusión este paquete será enviado a los demás circuitos adjuntos y a todos los equipos CE que participen a la misma instancia VPLS, desde el punto de vista de los equipos CE ven la instancia VPLS como una LAN emulada (Cisco, 2012).

El ejemplo de la Figura 54 se utiliza para explicar el funcionamiento de VPLS.

Figura 54.

Ejemplo de funcionamiento de VPLS



Nota. La figura muestra como es el funcionamiento VPLS. Elaboración propia, realizado con Edraw Max.

El siguiente diseño consta de tres clientes que se encuentran separadas en diferentes zonas geográficas, se cuentan con las direcciones MAC asociadas para cada sucursal como se muestra en la figura 54, la sucursal del sitio C envía una trama con la dirección de MAC de origen C:C:C hacia el destino de sucursal B, ahora el PE-2 no conoce la ubicación de cómo llegar hacia el destino de sucursal B, entonces PE-2 manda un mensaje de broadcast hacia los demás equipos dando a conocer las direcciones MAC que conoce hacia todos los puertos excepto en el puerto de donde se origina la trama, esto significa que el paquete se envía a todos los PW que se encuentren habilitados hacia PE-3 y PE-1.

En este punto PE-1 y PE-3 saben que el paquete pertenece a una instancia VPLS, ya que la trama ingresa por un PW ya establecido, seguido PE-1 y PE-3 realizan búsquedas de direcciones MAC de destino en sus tablas locales correspondientes a este cliente, si PE-3 no conoce la ubicación de la dirección MAC de B, se inunda la red con un mensaje de broadcast en sus puertos hasta el extremo del cliente CE-B, sin embargo, no inunda la trama a ningún otro enrutador de borde con el objetivo de prevenir bucles que puedan causar problemas de inestabilidad, de forma similar el PE-1 realiza la misma búsqueda en sus tablas locales hasta reenviar la trama en el puerto donde se encuentra conectado el equipo CE correcto.

La recepción de tramas con la dirección MAC C:C:C permite que cada PE aprenda la ubicación de la sucursal C. Por lo tanto, PE-3 y PE-1 crean una entrada en sus tablas de reenvío con una asociación entre la dirección MAC C:C:C y sus respectivas PW hacia PE-2. De esta forma, todos los PE aprenden las direcciones MAC y crean una asociación entre las direcciones MAC y las PW para destinos remotos en sus tablas de reenvío para cada instancia de VPLS en particular.

En el siguiente ejemplo se implementa la configuración de VPLS para la interconexión de tres sucursales, esto después que la infraestructura MPLS se encuentre activa.

- Como primer paso se configura una interfaz loopback de longitud 32 para identificar a cada equipo PE de la red y habilitar las sesiones LDP.

- Habilitar sobre las interfaces el transporte de tramas de capa 2 sobre ambos equipos PE.

- Asegurarse que el MTU sea del mismo valor en ambas interfaces en los extremos finales.

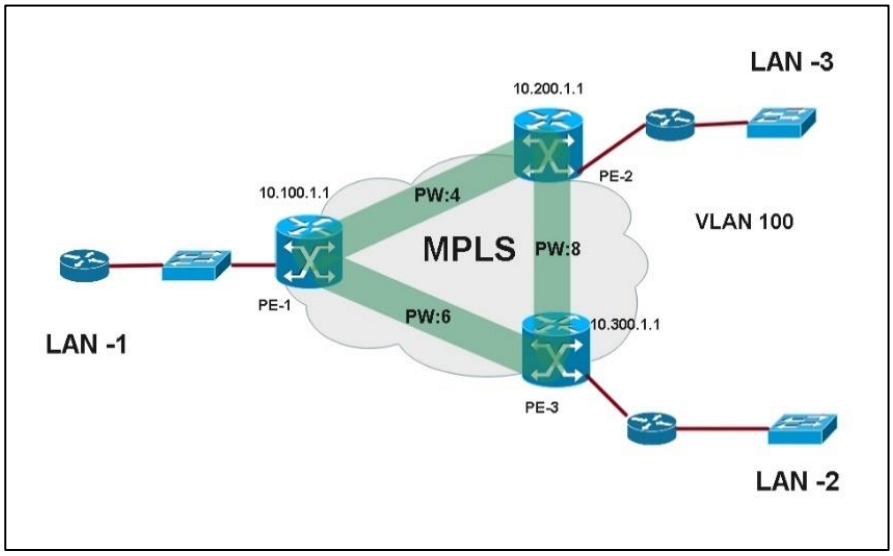
- Configurar el dominio VPLS.

- Asignar las interfaces que serán parte del dominio VPLS.

- Configurar los VFI con los PW ya definidos y nombrar los vecinos los cuales formaran parte de la topología VPLS.

Figura 55.

Ejemplo de implementación de VPLS



Nota. La figura muestra como es una implementación VPLS. Elaboración propia, realizado con Edraw Max.

Figura 56.

Configuración de VPLS sobre equipos PE

```
!
hostname PE-1
!
interface Loopback 0
 ipv4 address 10.100.1.1. 255.255.255.255
!
interface GigabitEthernet 0/0/0/0.100
l2 transport
 encapsulation dot1q 100
 !
l2vpn
 bridge group VPLS-GRUPO1
  brigde-domain VPLS- VPLS- domain1
  interface GigabitEthernet 0/0/0/0.100
 exit
!
 vfi  VPLS-1
  neighbor 10.200.1.1 pw-id 4
  neighbor 10.300.1.1 pw-id 6
!
```

```
!
hostname PE-2
!
interface Loopback 0
 ipv4 address 10.200.1.1. 255.255.255.255
!
interface GigabitEthernet 0/0/0/0.100
l2 transport
 encapsulation dot1q 100
 !
l2vpn
 bridge group VPLS-GRUPO1
  brigde-domain VPLS- VPLS- domain1
  interface GigabitEthernet 0/0/0/0.100
 exit
!
 vfi  VPLS-1
  neighbor 10.100.1.1 pw-id 4
  neighbor 10.300.1.1 pw-id 8
!
```

```
!
hostname PE-3
!
interface Loopback 0
 ipv4 address 10.300.1.1. 255.255.255.255
!
interface GigabitEthernet 0/0/0/0.100
l2 transport
 encapsulation dot1q 100
 !
l2vpn
 bridge group VPLS-GRUPO1
  brigde-domain VPLS- VPLS- domain1
  interface GigabitEthernet 0/0/0/0.100
 exit
!
 vfi  VPLS-1
  neighbor 10.100.1.1 pw-id 6
  neighbor 10.200.1.1 pw-id 8
!
```

Nota. Se muestra en la figura como debe realizarse una configuración VPLS en equipos PE. Elaboración propia, realizado con Cisco iOS.

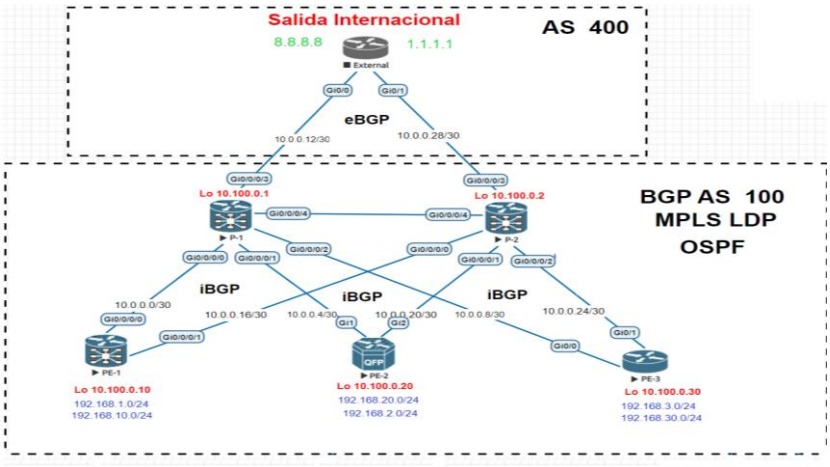
5. DISEÑO Y SIMULACIÓN DE LA RED MPLS

Las redes de los proveedores de servicios ISP del inglés *Internet Service Provider* es una infraestructura que está conformada por diversos equipos de alta capacidad que proporciona soluciones de transporte de datos, internet, telefonía entre otros tipos de servicios. La construcción de la red utilizada a continuación estará basada conforme a los diseños tradicionales de los proveedores de servicios, encontrando equipos *Provider* y *Provider* Edge, conectados mediante la topología *hub* and *spoke* por su capacidad de ser altamente escalable.

A continuación, se presenta el siguiente diseño que está conformado por dos equipos P y tres equipos PE, realizando la función de redundancia a nivel del Core para la prevención de fallas, adicional se comparte la conexión hacia otro proveedor que permite tener salida hacia el exterior de la red, proporcionando conectividad con salidas internacionales que se pueden simular hacia internet.

Figura 57.

Diseño de la red ISP



Nota. Ejemplo de diseño red ISP. Elaboración propia, realizado con EVE-NG.

Para este trabajo de investigación se estará utilizando EVE-NG como simulador de red virtual donde permite diseñar topologías de red complejas y crear simulaciones a través de las plataformas de varios fabricantes, para este diseño se estará utilizando el sistema operativo de Cisco System por ser un amplio proveedor de equipos de telecomunicaciones tanto como las versiones IOS, XE y XR.

5.1. Configuración del protocolo IGP

En la figura 57 se puede observar la infraestructura que tendrá el diseño de nuestra red ISP, se utiliza el sistema autónomo número 100, este AS estará diferenciando de otras organizaciones de proveedores que tendrán otro número diferente de AS, en escenarios reales esta asignación de AS es emitida por la

IANA quien tiene la responsabilidad de asignar los AS para cada red ISP presentes en la región.

Como primer paso para establecer nuestra red MPLS consiste en levantar entre los equipos un protocolo de enrutamiento dinámico que permita distribuir la información de ruteo dentro del propio sistema autónomo, con el objetivo de verificar que rutas se conocen en las distintas partes de la red, los protocolos más utilizados para este propósito son OSPF Y IS-IS son los que combinan el algoritmo de estado de enlace lo que significa que todos los equipos conocerán la información completa de la red, lo que permite estados de convergencia más rápido en comparación con otros protocolos dinámicos, en este trabajo se utiliza el protocolo OSPF por su facilidad de configuración y su diseño basado en áreas (Molenaar, 2020).

El diseño de red que se contempla para este trabajo es del tipo Hub and spoke cada equipo PE estará conectado hacia los dos equipos P que serán el Hub teniendo como objetivo consolidar, ordenar y direccionar los flujos de transporte de la red por lo que cada equipo PE estará concentrando la información de una zona geográfica para luego enviar la información hacia los P y este determinará hacía que zona dirigir la información hasta alcanzar su destino. Para los puntos de interconexión entre los nodos se aplica el segmento de red 10.0.0.0/30 este será el direccionamiento IPv4 que se configura sobre las interfaces físicas de nuestra red permitiendo la comunicación entre nodos, así mismo se crean las interfaces de loopback que ayuda al protocolo de enrutamiento en identificar a cada equipo de la topología completa. (Molenaar, 2020).

A continuación, se adjunta la configuración sobre el direccionamiento IPv4 de los equipos de la red ISP.

Figura 58.

Direccionamiento IPv4 sobre P-1

```
!
interface Loopback0
ipv4 address 10.100.0.1 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 10.0.0.1 255.255.255.252
no shutdown
!
interface GigabitEthernet0/0/0/1
ipv4 address 10.0.0.5 255.255.255.252
no shutdown
!
interface GigabitEthernet0/0/0/2
ipv4 address 10.0.0.9 255.255.255.252
no shutdown
!
```

Nota. Ejemplo de direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

Figura 59.

Direccionamiento IPv4 sobre P-2

```
!
interface Loopback0
ipv4 address 10.100.0.2 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 10.0.0.17 255.255.255.252
no shutdown
!
interface GigabitEthernet0/0/0/1
ipv4 address 10.0.0.21 255.255.255.252
no shutdown
!
interface GigabitEthernet0/0/0/2
ipv4 address 10.0.0.25 255.255.255.252
no shutdown
!
```

Nota. Ejemplo direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

Figura 60.

Direccionamiento IPv4 PE-1

```
!
interface Loopback0
ipv4 address 10.100.0.10 255.255.255.255
!
interface GigabitEthernet0/0/0/0
ipv4 address 10.0.0.2 255.255.255.252
no shutdown
!
interface GigabitEthernet0/0/0/1
ipv4 address 10.0.0.18 255.255.255.252
no shutdown
!
```

Nota. Ejemplo direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

Figura 61.

Direccionamiento IPv4 sobre PE-2

```
!
interface Loopback0
ip address 10.100.0.20 255.255.255.255
!
interface GigabitEthernet1
ip address 10.0.0.6 255.255.255.252
no shutdown
!
interface GigabitEthernet2
ip address 10.0.0.22 255.255.255.252
no shutdown
!
```

Nota. Ejemplo direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

Figura 62.

Direccionamiento IPv4 sobre PE-3

```
!
interface Loopback0
ip address 10.100.0.30 255.255.255.255
!
interface GigabitEthernet0/0
ip address 10.0.0.10 255.255.255.252
no shutdown
!
interface GigabitEthernet0/1
ip address 10.0.0.26 255.255.255.252
no shutdown
!
```

Nota. Ejemplo direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

Ya con las configuraciones de direccionamiento aplicadas de nuestra red ya está todo listo para levantar las sesiones de vecindad del protocolo IGP que dará conectividad a los equipos que conforman el AS, por lo tanto, se aplican las siguientes normas generales de configuración las cuales se presentan a continuación.

- Se estará utilizando el proceso 1 del protocolo OSPF para configurar la red, este proceso puede ser diferente en cada equipo, por orden general se utiliza el mismo para todos los enrutadores.

- Para identificar a cada equipo de la red se agrega la interfaz de loopback 0 configurado previamente.

- Se utilizará el área de Backbone o área 0 del protocolo OSPF para delimitar el alcance que tendrá nuestra red ISP.

- Se utiliza el segmento IPv4 configurado sobre las interfaces físicas para levantar la vecindad OSPF contra los equipos vecinos, también se debe configurar la máscara de wildcard correspondiente al segmento de red.

Teniendo en cuenta las normas generales de configuración se comparte en la siguiente tabla los comandos de OSPF de los sistemas operativos de Cisco.

Tabla 2.

Comandos de OSPF en Cisco XR

Comando	Propósito
RP/0/0/CPU0: *Router*> configure terminal	Entra en el modo de configuración Global.
RP/0/0/CPU0: *Router*# *router* ospf 1	Se definine el número de proceso OSPF.
RP/0/0/CPU0: *Router* (config-ospf)# *router*-id *a.b.c.d*	Configura un ID de enrutador fijo de 32 bits como identificador del dispositivo local que ejecuta OSPF.
RP/0/0/CPU0: *Router* (config-ospf)# area 0	Se define el área el cual está configurando nuestra red.
RP/0/0/CPU0: *Router* (config-ospf-ar)# interface G0/0/0/0	Se asignan las interfaces físicas que estarán ejecutando el protocolo OSPF para establecer la vecindad con los vecinos.

Nota. Ejemplo de comandos de OSPF. Elaboración propia, realizado con Cisco iOS.

Tabla 3.

Comandos de OSPF en Cisco IOS/XE

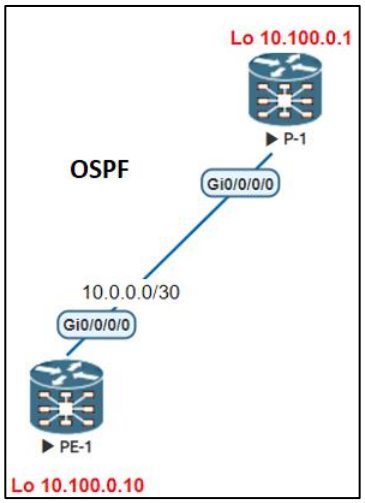
Comando	Propósito
Router> enable	Entra en el modo privilegiado del equipo.
Router# configure terminal	Entra en el modo de configuración Global.
Router# *router* ospf *1*	Se define el número de proceso OSPF.
Router(config-ospf)# *router*-id *a.b.c.d*	Configura un ID de enrutador fijo de 32 bits como identificador del dispositivo local que ejecuta OSPF.
Router (config-ospf)# network *a.b.c.d wild-card mask* area *area-id*	Dentro del mismo comando se establece el segmento el cual se estará levantado la vecindad, la máscara de wildcard y el área a la que pertenecerán los equipos.

Nota. Ejemplo de comandos de OSPF. Elaboración propia, realizado con Cisco iOS.

Ya conociendo los comandos básicos para levantar una vecindad sobre el protocolo OSPF se comparte el siguiente ejemplo sobre la topología de la figura 57 donde se levanta una sesión de vecindad entre el equipo P-1 contra PE-1 ambos sobre equipos XR.

Figura 63.

Ejemplo de configuración de OSPF



Nota. La figura muestra una configuración OSPF. Elaboración propia, realizado con EVE-NG.

Figura 64.

Proceso OSPF sobre P-1

```
!
router ospf 1
log adjacency changes
router-id 10.100.0.1
area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
```

Nota. Ejemplo del proceso OSPF. Elaboración propia, realizado con EVE-NG.

Figura 65.

Proceso OSPF sobre PE-1

```
!
router ospf 1
log adjacency changes
router-id 10.100.0.10
area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
```

Nota. Ejemplo del proceso OSPF. Elaboración propia, realizado con EVE-NG.

Para comprobar que la sesión se encuentre establecida se aplica el siguiente comando, el cual proporciona la siguiente información.

Figura 66.

Verificación de vecindad OSPF en P-1

```
RP/0/0/CPU0:P-1#show ip ospf neighbor
Wed Apr 20 03:53:31.632 UTC

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.10     1     FULL/BDR        00:00:37    10.0.0.2        GigabitEthernet0/0/0/0


Total neighbor count: 1
RP/0/0/CPU0:P-1#
```

Nota. Ejemplo de verificación OSPF. Elaboración propia, realizado con EVE-NG.

Figura 67.

Verificación de vecindad OSPF en PE-1

```
RP/0/0/CPU0:PE-1#show ospf neighbor
Wed Apr 20 03:54:31.148 UTC

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.1      1     FULL/DR         00:00:39    10.0.0.1        GigabitEthernet0/0/0/0
    Neighbor is up for 00:20:53

Total neighbor count: 1
RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de verificación OSPF. Elaboración propia, realizado con EVE-NG.

Los comandos de verificación para OSPF muestra la siguiente información con relación a su vecino configurado.

- Neighbor Id: indica el ID del enrutador vecino, en este caso es la dirección de loopback con cual estableció una vecindad.

- Pri: establece la prioridad para el enrutador vecino, este parámetro es utilizado para redes de múltiple acceso el cual no es el caso de este diseño.

- State: indica cual es el estado del vecino, la adyacencia se encuentra en el estado Full indicando que se estableció de forma correcta la sesión OSPF lo que significa que se encuentra listo para transportar los prefijos entre equipos.

- Dead time: indica la cantidad de tiempo restante que esperará el *router* para recibir un paquete OSPF de saludo del vecino, antes de declarar que el vecino está inactivo.

- Address: indica la dirección IP de la interfaz a la que este vecino está conectado directamente.

- Interface: indica sobre que interfaz física en la cual el vecino OSPF ha formado adyacencia.

Aplicando los mismos comandos de configuración hacia los demás equipos para crear las adyacencias entre vecinos teniendo en cuenta las direcciones IPv4 ya asignadas, se comparten las configuraciones correspondientes.

Figura 68.

Proceso OSPF sobre P-1

```
!
router ospf 1
router-id 10.100.0.1
area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
!
```

Nota. Ejemplo de proceso OSPF. Elaboración propia, realizado con EVE-NG.

Figura 69.

Proceso OSPF sobre P-2

```
!
router ospf 1
router-id 10.100.0.2
area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
!
```

Nota. Ejemplo de proceso OSPF. Elaboración propia, realizado con EVE-NG.

Figura 70.

Proceso OSPF sobre PE-2

```
!
router ospf 1
router-id 10.100.0.20
network 10.0.0.4 0.0.0.3 area 0
network 10.0.0.20 0.0.0.3 area 0
!
```

Nota. Ejemplo de proceso OSPF. Elaboración propia, realizado con EVE-NG.

Figura 71.

Proceso OSPF sobre PE-3

```
!
router ospf 1
router-id 10.100.0.30
network 10.0.0.8 0.0.0.3 area 0
network 10.0.0.24 0.0.0.3 area 0
network 10.100.0.30 0.0.0.0 area 0
!
```

Nota. Ejemplo de proceso OSPF. Elaboración propia, realizado con EVE-NG.

Al terminar las configuraciones sobre el protocolo OSPF sobre todos los enrutadores se verifica el estado de adyacencia de los equipos P hacia los PE, el estado del protocolo deberá encontrarse correctamente establecida para su correcto funcionamiento dentro de la red, para comprobarlo nuevamente se usa del comando de comprobación.

Figura 72.

Verificación de vecindad OSPF en P-1

```
RP/0/0/CPU0:P-1#show ip ospf neighbor
Wed Apr 20 03:53:31.632 UTC

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.10     1     FULL/BDR        00:00:37    10.0.0.2        GigabitEthernet0/0/0/0
    Neighbor is up for 00:19:53
10.100.0.20     1     FULL/BDR        00:00:37    10.0.0.6        GigabitEthernet0/0/0/1
    Neighbor is up for 00:11:41
10.100.0.30     1     FULL/BDR        00:00:36    10.0.0.10       GigabitEthernet0/0/0/2
    Neighbor is up for 00:06:52

Total neighbor count: 3
RP/0/0/CPU0:P-1#
```

Nota. Ejemplo de verificación OSPF. Elaboración propia, realizado con EVE-NG.

Figura 73.

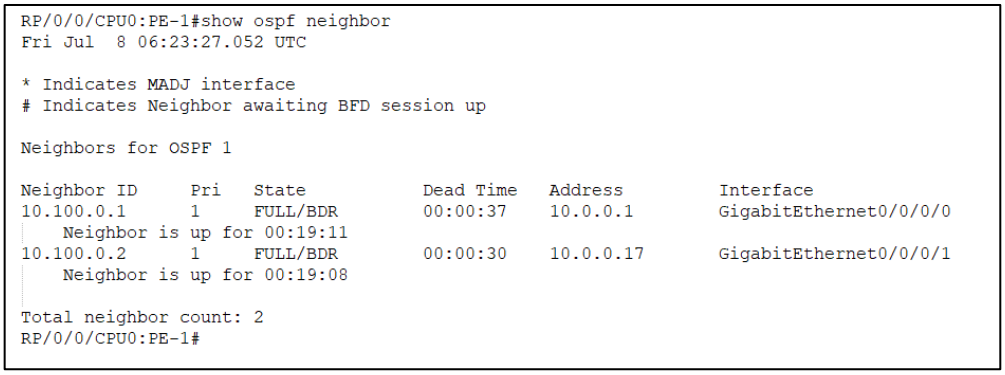
Verificación de vecindad OSPF en P-2

```
RP/0/0/CPU0:P-2#show ospf neighbor
Fri Jul  8 06:22:23.196 UTC

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.10     1     FULL/DR         00:00:31    10.0.0.18       GigabitEthernet0/0/0/0
    Neighbor is up for 00:17:59
10.100.0.20     1     FULL/BDR        00:00:39    10.0.0.22       GigabitEthernet0/0/0/1
    Neighbor is up for 00:00:19
10.100.0.30     1     FULL/DR         00:00:30    10.0.0.26       GigabitEthernet0/0/0/2
    Neighbor is up for 00:18:05

Total neighbor count: 3
RP/0/0/CPU0:P-2#
```

Nota. Ejemplo de verificación OSPF. Elaboración propia, realizado con EVE-NG.

Figura 74.

Verificación de vecindad OSPF en PE-1

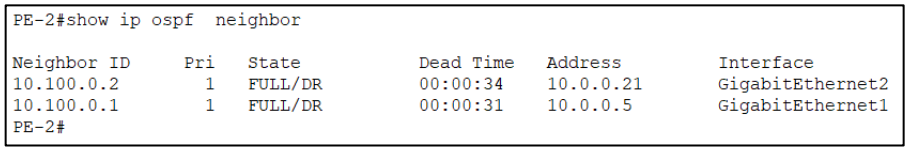


```
RP/0/0/CPU0:PE-1#show ospf neighbor
Fri Jul  8 06:23:27.052 UTC

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.1      1     FULL/BDR        00:00:37    10.0.0.1        GigabitEthernet0/0/0/0
    Neighbor is up for 00:19:11
10.100.0.2      1     FULL/BDR        00:00:30    10.0.0.17       GigabitEthernet0/0/0/1
    Neighbor is up for 00:19:08

Total neighbor count: 2
RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de verificación OSPF. Elaboración propia, realizado con EVE-NG.

Figura 75.

Verificación de vecindad OSPF en PE-2

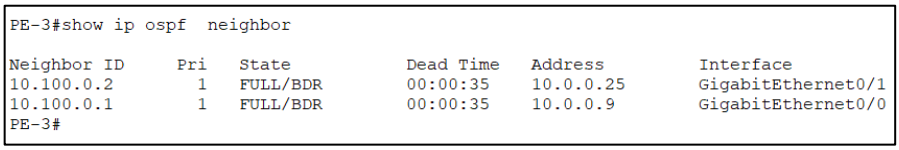


```
PE-2#show ip ospf  neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.2        1   FULL/DR         00:00:34    10.0.0.21       GigabitEthernet2
10.100.0.1        1   FULL/DR         00:00:31    10.0.0.5        GigabitEthernet1
PE-2#
```

Nota. Ejemplo de verificación OSPF. Elaboración propia, realizado con EVE-NG.

Figura 76.

Verificación de vecindad OSPF en PE-3



```
PE-3#show ip ospf  neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.2        1   FULL/BDR        00:00:35    10.0.0.25       GigabitEthernet0/1
10.100.0.1        1   FULL/BDR        00:00:35    10.0.0.9        GigabitEthernet0/0
PE-3#
```

Nota. Ejemplo de verificación OSPF. Elaboración propia, realizado con EVE-NG.

En este momento se finaliza la configuración del IGP del sistema autónomo ya que es importante para los procesos de ruteo de la red, en donde cada equipo PE comparte una vecindad sobre OSPF hacia los equipos P permitiendo identificar a cada enrutador sobre la red ya creada.

5.2. Configuración de sesiones iBGP

El protocolo BGP tiene dos tipos de sesiones los internos iBGP y los externos eBGP, estos tipos de sesiones se utilizan dependiendo si el vecino al cual quieren formar adyacencia se encuentra en el AS o forma parte de otro AS de forma externa.

En los proveedores de servicios se considera fundamental manejar adyacencias iBGP en el propio sistema autónomo, como proveedor es necesario mantener interconexión con demás AS que puedan compartir diferentes prefijos a nivel global, por lo tanto por temas de escalabilidad es necesario crear adyacencias BGP sobre OSPF debido que no es posible llevar toda la tabla de enrutamiento de diferentes AS o de Internet dentro del IGP lo que provocaría problemas de rendimiento sobre el AS ya que OSPF no fue diseñado para manejar miles de rutas en sus tablas de enrutamiento, entonces, la razón por la que tienen ambos, es porque en una red especifica, se tiene una complejidad suficientemente baja para manejarlo con el protocolo de estado de enlace, lo que permite obtener todas las ventajas (Molenaar, 2020).

Ya con las configuraciones sobre el IGP previamente configurado se está listos para crear adyacencias sobre el protocolo BGP por lo que se consideran las siguientes normas generales de configuración las cuales se presentan a continuación.

- El sistema autónomo que se estará utilizando para la simulación será el AS 100, este número identifica de otros AS que existan en la región, en los entornos reales la asignación de los números de AS es emitido por la organización de la IANA a través de los registros regionales de Internet (RIR), de acuerdo con las normas ya establecidos por la entidad (Molenaar, 2020).

- Se estará utilizando la dirección IP de loopback previamente configurado para establecer las vecindades hacia los demás equipos, como buena práctica se utilizan estas direcciones para ofrecer opciones de redundancia en casos que compartan diferentes enlaces hacia un mismo enrutador, ya que si se establecen sobre las direcciones IP de las interfaces el protocolo no tendrá el mismo efecto y no conocerá otros caminos en caso se produjera algún fallo en alguna de las interfaces de la red troncal.

- Es necesario especificar al protocolo BGP la familia de direccionamiento global que se estarán utilizando para levantar las adyacencias, en este caso se utiliza el protocolo de Internet versión 4 del tipo unicast, esto se define tomando en cuenta al tipo de información que estará transportando BGP.

Teniendo en cuenta las normas generales de configuración se comparte en la siguiente tabla los comandos de BGP de los sistemas operativos de Cisco.

Tabla 4.

Comandos de BGP en Cisco XR

Comando	Propósito
RP/0/0/CPU0: *Router*> configure terminal	Entra en el modo de configuración Global.
RP/0/0/CPU0: *Router*# *router* bgp xy	Se configura un proceso de enrutamiento donde se define el argumento de AS para especificar de 0 y 65534, que identifique el dispositivo con otros equipos BGP.
RP/0/0/CPU0: *Router* (config-bgp)# bgp *router*-id *a.b.c.d*	Configura un ID de enrutador fijo de 32 bits como identificador del dispositivo local que ejecuta BGP.
RP/0/0/CPU0: *Router* (config-bgp)# address-family ipv4 unicast	Especifica la familia de direcciones IPv4 y entra en el modo de configuración de la familia de direcciones, la palabra clave unicast especifica la familia de direcciones de unidifusión IPv4.
RP/0/0/CPU0: *Router* (config-bgp-af)# neighbor *a.b.c.d*	Se declara el equipo vecino con cual estará formando la adyacencia BGP para el intercambio de prefijos.
RP/0/0/CPU0: *Router* (config-bgp-nbr)#remote-as	Se agrega el número de AS el cual forma parte el vecino
RP/0/0/CPU0: *Router* (config-bgp-nbr)# address-family ipv4 unicast	Dentro de la configuración del vecino se declara la familia de direcciones de unidifusión IPv4.

Nota. Ejemplo de comandos BGP. Elaboración propia, realizado con Excel.

Tabla 5.

Comandos BGP en Cisco IOS/XE

Comando	Propósito
Router> enable	Entra en el modo privilegiado del equipo.
Router# configure terminal	Entra en el modo de configuración Global.
Router# *router* bgp *xy*	Configura un proceso de enrutamiento donde se define el argumento de AS para especificar de 0 y 65534, que identifique el dispositivo con otros equipos BGP.
Router(config-*router*)# *router*-id *a.b.c.d*	Configura un ID de enrutador fijo de 32 bits como identificador del dispositivo local que ejecuta BGP.
Router (config-*router*)# no bgp default ipv4-unicast	Deshabilita la familia de direcciones de unicast IPv4 para el proceso de enrutamiento BGP, con la finalidad de habilitar funciones de multiprotocolo que permite transportar diferentes familias de direccionamiento, es la opción más recomendada para entornos ISP.

Continuación de la tabla 5.

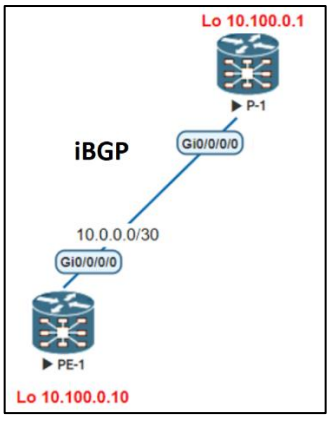
Comando	Propósito
Router (config-*router*)# neighbor *a.b.c.d* remote-as	Se declara el equipo vecino con cual estará formando la adyacencia BGP así mismo agregando el AS que forma parte.
Router (config-*router*)# neighbor *a.b.c.d* update-source Loopback0	Configura un *router* para seleccionar una interfaz específica para recibir actualizaciones de la tabla de enrutamiento.
Router (config-*router*)# address-family ipv4	Dentro de la configuración del vecino se declara la familia de direcciones de unicast IPv4 el cual el vecino BGP estará ejecutando.
Router (config-*router*-af)# neighbor *a.b.c..d* activate	Activa el intercambio de prefijos sobre la familia de direccionamiento seleccionado.

Nota. Ejemplo de comandos BGP. Elaboración propia, realizado con Excel.

Ya conociendo los comandos para levantar una vecindad sobre el protocolo BGP se comparte el siguiente ejemplo sobre la topología de la figura 57 donde se levanta una sesión de vecindad entre el equipo P-1 contra PE-1 ambos sobre equipos XR.

Figura 77.

Ejemplo de configuración de BGP



Nota. La figura muestra una configuración BGP. Elaboración propia, realizado con EVE-NG.

Figura 78.

Proceso BGP sobre P-1

```
!
router bgp 100
bgp router-id 10.100.0.1
address-family ipv4 unicast
!
neighbor 10.100.0.10
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 79.

Proceso BGP sobre PE-1

```
!
router bgp 100
bgp router-id 10.100.0.10
address-family ipv4 unicast
!
neighbor 10.100.0.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Para comprobar que la sesión se encuentre establecida se aplica el siguiente comando, el cual proporciona la siguiente información.

Figura 80.

Verificación de vecindad BGP en P-1

```
RP/0/0/CPU0:P-1#show bgp ipv4 unicast summary
Fri Jul  8 07:10:12.788 UTC
BGP router identifier 10.100.0.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 14
BGP main routing table version 14
BGP NSR Initial initsync version 6 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker              14         14         14         14          14           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd

10.100.0.10       0   100      42      47       14    0    0 00:36:08          0

RP/0/0/CPU0:P-1#
```

Nota. Ejemplo de verificación BGP. Elaboración propia, realizado con EVE-NG.

Los comandos de verificación para BGP muestran la siguiente información con relación a su vecino.

- Neighbor: dirección IP del vecino BGP configurado
- AS: sistema autónomo al que pertenece el vecino
- MsgRcvd: número de mensajes recibidos del vecino
- MsgSent: número de mensajes enviados al vecino
- TblVer: número de la versión de la tabla, que se incrementa cada vez que cambia la tabla de enrutamiento.

- InQ: número de mensajes recibidos en la cola de entrada.
- OutQ: número de mensajes listos en la cola de salida.
- Up/Down: tiempo transcurrido cuando el vecino ha estado activo o caído.
- St/PfxRcd: estado del vecino y número de rutas recibidas. Si no se indica ningún estado, el estado está arriba.

A continuación, se comparten las configuraciones completas de todos los enrutadores que conforman la red de back bone de nuestra red ISP.

Figura 81.

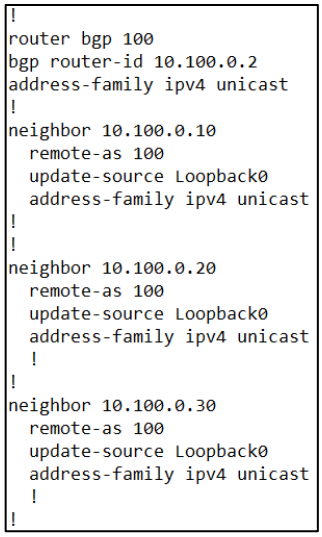
Proceso BGP sobre P-1

```
!
router bgp 100
bgp router-id 10.100.0.1
address-family ipv4 unicast
!
neighbor 10.100.0.10
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
!
neighbor 10.100.0.20
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
!
neighbor 10.100.0.30
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 82.

Proceso BGP sobre P-2

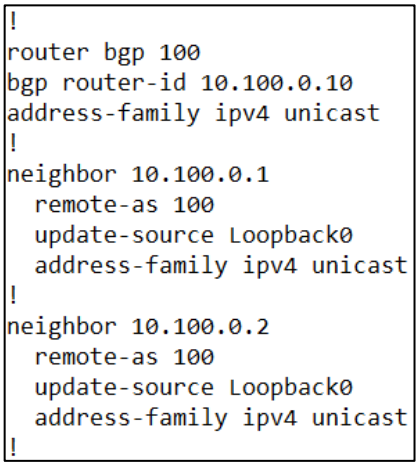


```
!
router bgp 100
bgp router-id 10.100.0.2
address-family ipv4 unicast
!
neighbor 10.100.0.10
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
!
!
neighbor 10.100.0.20
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
!
neighbor 10.100.0.30
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 83.

Proceso BGP sobre PE-1



```
!
router bgp 100
bgp router-id 10.100.0.10
address-family ipv4 unicast
!
neighbor 10.100.0.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
!
neighbor 10.100.0.2
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 84.

Proceso BGP sobre PE-2

```
!
router bgp 100
bgp router-id 10.100.0.20
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 10.100.0.1 remote-as 100
neighbor 10.100.0.1 update-source Loopback0
neighbor 10.100.0.2 remote-as 100
neighbor 10.100.0.2 update-source Loopback0
!
address-family ipv4
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.2 activate
exit-address-family
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 85.

Proceso BGP sobre PE-3

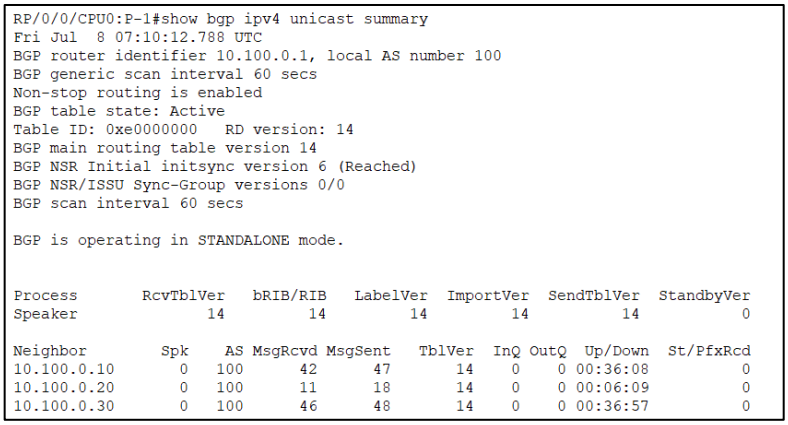
```
!
router bgp 100
bgp router-id 10.100.0.30
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 10.100.0.1 remote-as 100
neighbor 10.100.0.1 update-source Loopback0
neighbor 10.100.0.2 remote-as 100
neighbor 10.100.0.2 update-source Loopback0
!
address-family ipv4
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.2 activate
exit-address-family
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Luego de aplicar los comandos de configuración para levantar las sesiones iBGP se aplican los comandos de comprobación para verificar que la configuración sea la correcta.

Figura 86.

Verificación de BGP sobre P-1

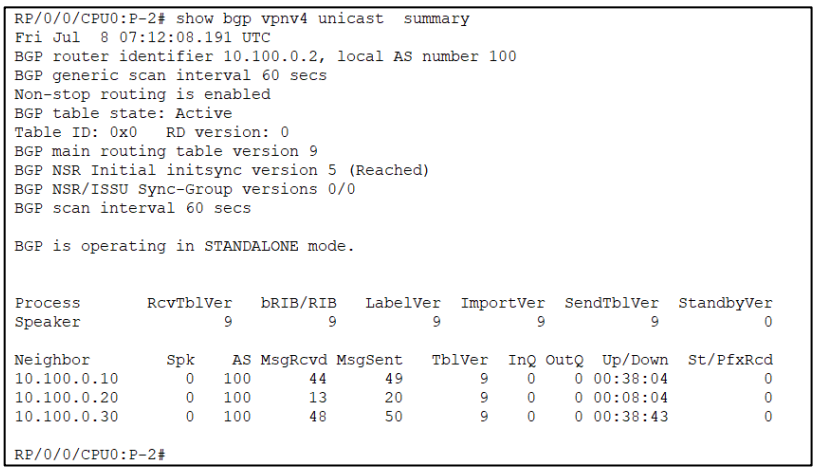


```
RP/0/0/CPU0:P-1#show bgp ipv4 unicast summary
Fri Jul  8 07:10:12.788 UTC
BGP router identifier 10.100.0.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 14
BGP main routing table version 14
BGP NSR Initial initsync version 6 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker              14         14         14         14          14           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.100.0.10       0   100      42      47       14    0    0 00:36:08          0
10.100.0.20       0   100      11      18       14    0    0 00:06:09          0
10.100.0.30       0   100      46      48       14    0    0 00:36:57          0
```

Nota. Ejemplo de verificación BGP. Elaboración propia, realizado con EVE-NG.

Figura 87.

Verificación de BGP sobre P-2



```
RP/0/0/CPU0:P-2# show bgp vpnv4 unicast  summary
Fri Jul  8 07:12:08.191 UTC
BGP router identifier 10.100.0.2, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 9
BGP NSR Initial initsync version 5 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               9          9          9          9           9           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.100.0.10       0   100      44      49        9    0    0 00:38:04          0
10.100.0.20       0   100      13      20        9    0    0 00:08:04          0
10.100.0.30       0   100      48      50        9    0    0 00:38:43          0

RP/0/0/CPU0:P-2#
```

Nota. Ejemplo de verificación BGP. Elaboración propia, realizado con EVE-NG.

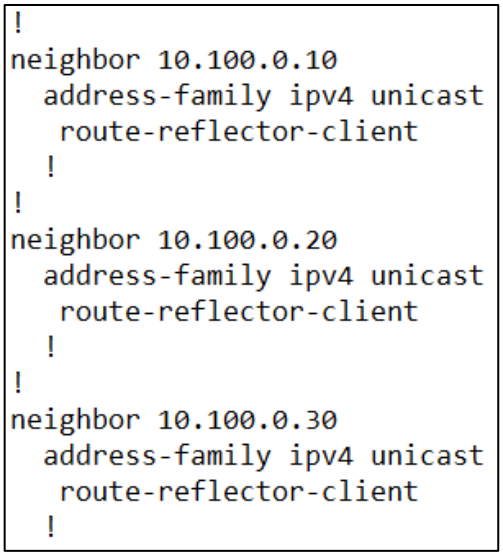
Algo muy importante de los proveedores de servicios es tener en cuenta la regla de horizonte dividido de BGP, esta regla cumple la función de prevención de bucles dentro de la red que pueda provocar problemas de rendimiento, esto indica que ningún vecino iBGP puede anunciar prefijos aprendidos hacia otros vecinos iBGP, para evitar este problema, se debe establecer una malla completa iBGP hacia todos los equipos que conforman el AS este método es factible cuando hay pocos enrutadores PE. Si dentro de la red existen varios el mantenimiento de las sesiones iBGP sobre la malla completa proporciona un aumento de los recursos de hardware de los equipos, degradando el rendimiento del enrutador, para resolver este problema es necesario romper la regla del horizonte dividido mediante dos mecanismos que son los reflectores de rutas y confederaciones, en este trabajo se estará utilizando el reflector de ruta como es habitual dentro de las redes ISP.

Un reflector de ruta es el enrutador encargado de enviar actualizaciones a vecinos a través del mismo AS, los vecinos iBGP se necesitan identificarse como clientes dentro de la configuración de BGP. Los reflectores de rutas básicamente se comportan como un espejo que reflejan las actualizaciones de sus clientes sin necesidad de una red totalmente mallada rompiendo de esa forma la regla del horizonte dividido, por lo tanto, se define en la topología existente los reflectores de ruta en el Core de la red (Molenaar, 2020).

En la topología de la figura 57 los reflectores de rutas serán los equipos P-1 y P-2, para agregar los equipos PE dentro de la configuración de clientes se aplica los siguientes comandos BGP en ambos equipos.

Figura 88.

Configuración de reflector de ruta sobre P-1 y P-2



```
!
neighbor 10.100.0.10
  address-family ipv4 unicast
   route-reflector-client
  !
!
neighbor 10.100.0.20
  address-family ipv4 unicast
   route-reflector-client
  !
!
neighbor 10.100.0.30
  address-family ipv4 unicast
   route-reflector-client
  !
```

Nota. Ejemplo de configuración de reflector. Elaboración propia, realizado con EVE-NG.

5.3. Configuración de sesiones eBGP

Las sesiones BGP que se encuentran en diferentes sistemas autónomos se establecen como sesiones externas, estas sesiones proporcionan conectividad con diferentes AS para el envío y recepción de prefijos a nivel global, en la figura 57 se observa que nuestra red ISP le corresponde el AS 100 mientras el proveedor B le corresponde el AS 400. Para establecer las sesiones eBGP se consideran las siguientes características (Molenaar, 2020).

- En el proceso de enrutamiento la distancia administrativa se le asigna un valor de 20 mientras con las sesiones internas la distancia será de 200.

- Los paquetes TTL o de tiempo de vida tienen un valor de uno de forma predeterminada, por lo que se recomienda que la sesiones se establezcan del tipo de red punto a punto utilizando la dirección IP física directamente conectada.

- Las sesiones eBGP se configuran de forma similar excepto el número de AS dentro de la instrucción remote-as, que es diferente al proceso BGP local.

- Para la prevención de bucles, eBGP utiliza el atributo AS-Path lo que permite descartar los prefijos que son recibidos de un vecino eBGP que contiene el AS local de quien las recibe.

Figura 89.

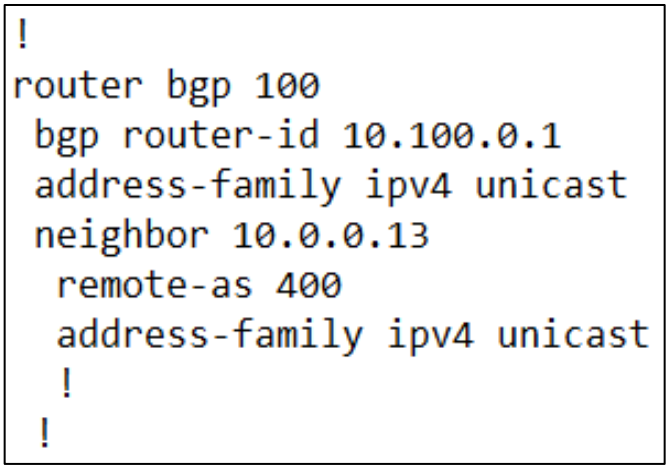
Proceso BGP sobre el equipo externo

```
!
router bgp 400
bgp router-id 20.200.10.10
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 10.0.0.14 remote-as 100
neighbor 10.0.0.30 remote-as 100
!
address-family ipv4
  network 1.1.1.1 mask 255.255.255.255
  network 8.8.8.8 mask 255.255.255.255
  neighbor 10.0.0.14 activate
  neighbor 10.0.0.30 activate
exit-address-family
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 90.

Proceso BGP sobre P-1 hacia externo

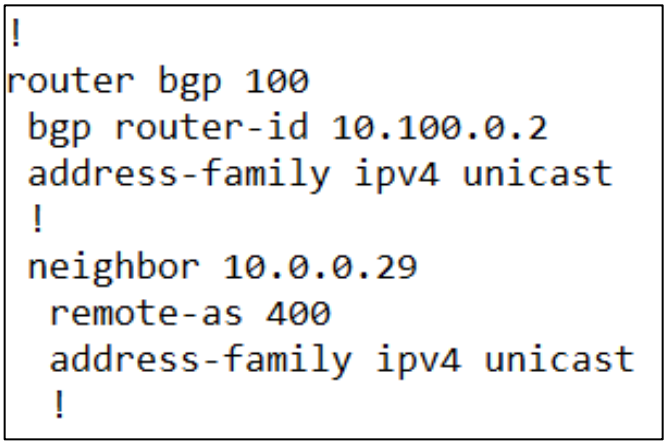


```
!
router bgp 100
 bgp router-id 10.100.0.1
 address-family ipv4 unicast
 neighbor 10.0.0.13
  remote-as 400
  address-family ipv4 unicast
  !
 !
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 91.

Proceso BGP sobre P-2 hacia externo



```
!
router bgp 100
 bgp router-id 10.100.0.2
 address-family ipv4 unicast
 !
 neighbor 10.0.0.29
  remote-as 400
  address-family ipv4 unicast
  !
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

5.4. Configuración de MPLS

Las redes de los proveedores de servicios se implementa la técnica de reenvío por etiquetas de MPLS con la finalidad de aumentar la velocidad en el envío y la toma de decisiones del equipo enrutador, dado que las búsquedas ya no se realizan sobre el direccionamiento IP ahora cada paquete se le asigna una etiqueta el cual sirve para enviarlos al siguiente equipo por el camino más corto hasta llegar a su destino (Molenaar, 2020).

Antes de contar con la red MPLS, previamente es necesario configurar el protocolo IGP debido que el protocolo LDP construye los LSP basándose en las mismas métricas, con esa información agrupa múltiples prefijos alojados en el mismo LSR de egreso según los LSP creados para enviarlos por un mismo FEC.

Con base a la topología de la figura 57 se estará configurando MPLS para lograr las adyacencias de los equipos PE hacia los P, la configuración se realiza dentro de los diferentes sistemas operativos que cuenta el fabricante como se presenta a continuación.

Tabla 6.

Comandos LDP en Cisco XR

Comando	Propósito
RP/0/0/CPU0:*Router*> configure terminal	Entra en el modo de configuración Global.
RP/0/0/CPU0:*Router*(config)# mpls label range	Seleccionar de un rango de etiquetas el cual se desea utilizar, este comando es opcional dentro de la configuración.
RP/0/0/CPU0:*Router*(config)# mpls ldp	Asigna el protocolo dedicado a la distribución de etiquetas, el más utilizado es LDP.
RP/0/0/CPU0:*Router*(config-ldp)# *router*-id *a.b.c.d*	Configura un ID de enrutador fijo de 32 bits como identificador del dispositivo local que ejecuta MPLS.

Continuación de la tabla 6.

Comando	Propósito
RP/0/0/CPU0:*Router*(config-ldp)# discovery hello interval 5	Temporizador para el descubrimiento de vecinos por defecto se encuentra configurado para 5 segundos.
RP/0/0/CPU0:*Router*(config-ldp)# discovery hello holdtime 15	Se configura el tiempo de espera que utiliza LDP para declarar a un vecino como inalcanzable por defecto se encuentra configurado en 15 segundos.
RP/0/0/CPU0:*Router*(config-ldp-if)# interface GigabitEthernet 0/0/0/0	Se asignan las interfaces que estarán siendo parte del dominio MPLS.

Nota. Ejemplo de comandos LDP. Elaboración propia, realizado con Excel.

Tabla 7.

Comandos LDP sobre en Cisco IOS/XE

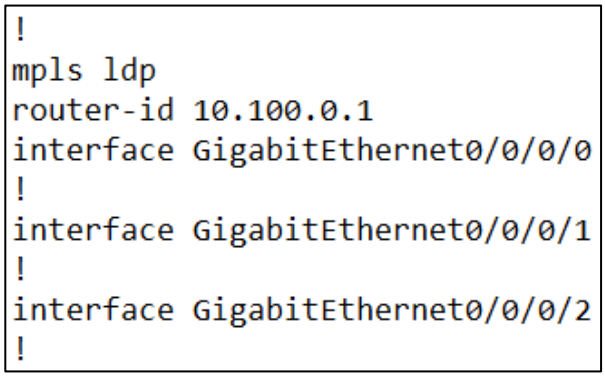
Comando	Propósito
Router> enable	Entra en el modo privilegiado del equipo.
Router# configure terminal	Entra en el modo de configuración Global.
Router(config)#ip cef	Se habilitó CEF que proporciona la capacidad de conmutar paquetes.
Router(config)#mpls label protocol ldp	Asigna el protocolo dedicado a la distribución de etiquetas, el más utilizado es LDP.
Router(config)#mpls ldp *router*-id loopback 0	Configura un ID de enrutador fijo de 32 bits como identificador del dispositivo local que ejecuta MPLS.
Router(config)#mpls label range	Seleccionar un rango de etiquetas el cual se desea utilizar, este comando es opcional dentro de la configuración.
Router(config)#mpls ldp discovery hello interval 5	Temporizador para el descubrimiento de vecinos por defecto se encuentra configurado para 5 segundos.
Router(config)#mpls ldp discovery hello holdtime 15	Se configura el tiempo de espera que utiliza LDP para declarar a un vecino como inalcanzable por defecto se encuentra configurado en 15 segundos.
Router(config-if)#interface gigabitEthernet 0/0	Se ingresan las interfaces que serán parte del dominio MPLS
Router(config-if) # mpls ip	Dentro de la configuración de la interfaz se activa LDP para la formación de adyacencias.

Nota. Ejemplo de comandos LDP. Elaboración propia, realizado con Excel.

Se comparten las siguientes configuraciones del protocolo LDP sobre todos los equipos de la red del ISP.

Figura 92.

Proceso LDP en P-1

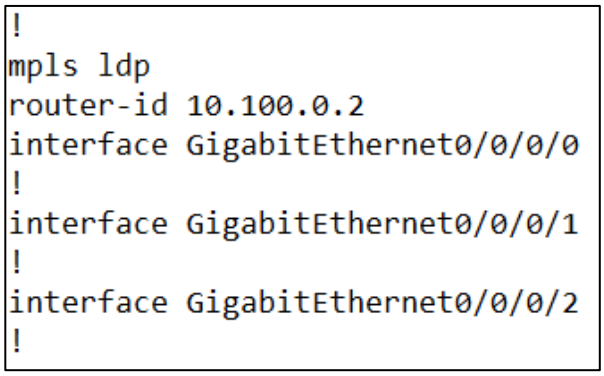


```
!
mpls ldp
router-id 10.100.0.1
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
```

Nota. Ejemplo de proceso LDP. Elaboración propia, realizado con EVE-NG.

Figura 93.

Proceso LDP en P-2

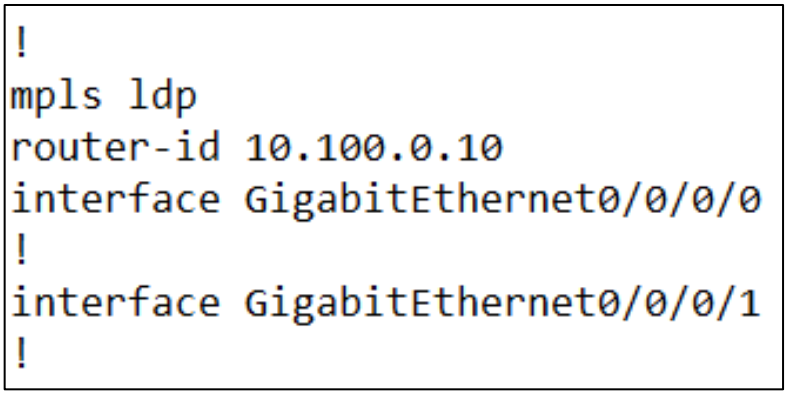


```
!
mpls ldp
router-id 10.100.0.2
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
```

Nota. Ejemplo de proceso LDP. Elaboración propia, realizado con EVE-NG.

Figura 94.

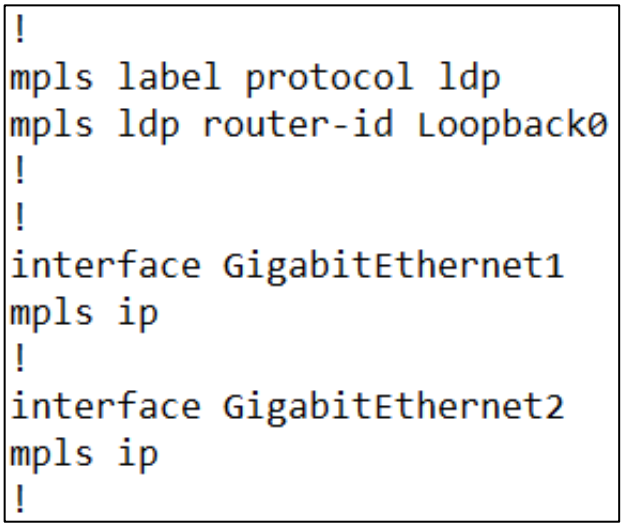
Proceso LDP en PE-1



Nota. Ejemplo de proceso LDP. Elaboración propia, realizado con EVE-NG.

Figura 95.

Proceso LDP en PE-2



Nota. Ejemplo de proceso LDP. Elaboración propia, realizado con EVE-NG.

Figura 96.

Proceso LDP en PE-3

```
!
mpls label protocol ldp
mpls ldp router-id Loopback0
!
!
interface GigabitEthernet 0/0
mpls ip
!
interface GigabitEthernet 0/1
mpls ip
!
```

Nota. Ejemplo de proceso LDP. Elaboración propia, realizado con EVE-NG.

Para verificar que las sesiones LDP fueron configuradas de forma correcta hay que apoyarse con los siguientes comandos de verificación, donde se puede obtener la siguiente información.

- Peer LDP *Identifier*: identifica el vecino con el cual creó una adyacencia LDP, la dirección IP corresponde a la dirección de loopback del vecino.

- TCP *Connection*: se establece una conexión TCP entre los enrutadores que desean compartir información a través de sus direcciones de loopback y los números de puertos asociados durante la conexión, adicional se observa que el tiempo de espera de la sesión de mantiene dentro de 180 segundos.

- *State*: muestra el estado de la conexión, los mensajes recibidos y enviados durante el establecimiento de la sesión y el modo de distribución de etiquetas.

- LDP *Discovery Sources*: lista las interfaces físicas que participan en el establecimiento de la sesión LDP.

- *Address bound to this peer*: almacena todos los prefijos IPv4 que contiene el enrutador remoto, para luego asignarlos dentro de una etiqueta.

Figura 97.

Verificación LDP en P-1 hacia los vecinos PE-1, PE-2 y PE-3

```
P/0/0/CPU0:P-1#show mpls ldp neighbor
Fri Jul  8 07:11:07.894 UTC

Peer LDP Identifier: 10.100.0.30:0
  TCP connection: 10.100.0.30:52173 - 10.100.0.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 62/64; Downstream-Unsolicited
  Up time: 00:37:53
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/2
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (5)
      10.0.0.10      10.0.0.26      10.100.0.30
    IPv6: (0)

Peer LDP Identifier: 10.100.0.10:0
  TCP connection: 10.100.0.10:18591 - 10.100.0.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 62/64; Downstream-Unsolicited
  Up time: 00:37:10
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/0
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.0.0.2       10.0.0.18      10.0.0.93      10.100.0.10
    IPv6: (0)
        Addresses bound to this peer:
    IPv4: (6)
      10.0.0.2       10.0.0.18      10.0.0.93      10.100.0.10
    IPv6: (0)
Peer LDP Identifier: 10.100.0.20:0
  TCP connection: 10.100.0.20:15707 - 10.100.0.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 27/28; Downstream-Unsolicited
  Up time: 00:07:05
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/1
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (5)
      10.0.0.6       10.0.0.22      10.100.0.20
    IPv6: (0)

RP/0/0/CPU0:P-1#
```

Nota. Ejemplo de verificación LDP hacia los vecinos. Elaboración propia, realizado con EVE-NG.

Figura 98.

Verificación LDP en P-2 hacia los vecinos PE-1, PE-2 y PE-3

```
P/0/0/CPU0:P-2#show mpls ldp neighbor
Fri Jul  8 07:12:22.140 UTC

Peer LDP Identifier: 10.100.0.30:0
  TCP connection: 10.100.0.30:55564 - 10.100.0.2:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 63/64; Downstream-Unsolicited
  Up time: 00:38:57
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/2
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (5)
      10.0.0.10      10.0.0.26      10.100.0.30
    IPv6: (0)

Peer LDP Identifier: 10.100.0.10:0
  TCP connection: 10.100.0.10:41754 - 10.100.0.2:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 63/66; Downstream-Unsolicited
  Up time: 00:38:25
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/0
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.0.0.2       10.0.0.18      10.0.0.93      10.100.0.10
    IPv6: (0)
        Peer LDP Identifier: 10.100.0.20:0
  TCP connection: 10.100.0.20:17942 - 10.100.0.2:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 28/30; Downstream-Unsolicited
  Up time: 00:08:32
  LDP Discovery Sources:
    IPv4: (1)
      GigabitEthernet0/0/0/1
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (5)
      10.0.0.6       10.0.0.22      10.100.0.20
    IPv6: (0)

RP/0/0/CPU0:P-2#
```

Nota. Ejemplo de verificación LDP hacia los vecinos. Elaboración propia, realizado con EVE-NG.

Con la siguiente tabla se pueden observar las etiquetas que son asignadas para cada prefijo IPv4 que se encuentra almacenadas en las tablas de enrutamiento, se observan las etiquetas de entrada y salida las cuales serán conmutadas por LDP hacia los demás vecinos con su respectiva interfaz de salida y dirección IPv4 del siguiente salto.

Figura 99.

Reenvío de etiquetas en P-1

```
RP/0/0/CPU0:P-1#show mpls ldp forwarding
Thu May 12 23:26:25.517 UTC

Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix          Label   Label(s)       Outgoing     Next Hop            Flags
                In      Out            Interface                        G S R
--------------- ------- -------------- ------------ ------------------- -----
10.0.0.16/30    24001   ImpNull        Gi0/0/0/0    10.0.0.2
10.0.0.20/30    24003   ImpNull        Gi0/0/0/1    10.0.0.6
10.0.0.24/30    24005   ImpNull        Gi0/0/0/2    10.0.0.10
10.100.0.2/32   24006   24007          Gi0/0/0/0    10.0.0.2
                        17             Gi0/0/0/1    10.0.0.6
                        16             Gi0/0/0/2    10.0.0.10
10.100.0.10/32  24000   ImpNull        Gi0/0/0/0    10.0.0.2
10.100.0.20/32  24002   ImpNull        Gi0/0/0/1    10.0.0.6
10.100.0.30/32  24004   ImpNull        Gi0/0/0/2    10.0.0.10

RP/0/0/CPU0:P-1#
```

Nota. Ejemplo de reenvío de etiquetas. Elaboración propia, realizado con EVE-NG.

Figura 100.

Reenvío de etiquetas en P-2

```
RP/0/0/CPU0:P-2#show mpls ldp forwarding
Thu May 12 23:27:30.572 UTC

Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup

Prefix           Label   Label(s)       Outgoing     Next Hop            Flags
                 In      Out            Interface                        G S R
---------------- ------- -------------- ------------ ------------------- -----
10.0.0.0/30      24006   ImpNull        Gi0/0/0/0    10.0.0.18
10.0.0.4/30      24001   ImpNull        Gi0/0/0/1    10.0.0.22
10.0.0.8/30      24003   ImpNull        Gi0/0/0/2    10.0.0.26
10.100.0.1/32    24004   24000          Gi0/0/0/0    10.0.0.18
                         19             Gi0/0/0/1    10.0.0.22
                         20             Gi0/0/0/2    10.0.0.26
10.100.0.10/32   24005   ImpNull        Gi0/0/0/0    10.0.0.18
10.100.0.20/32   24000   ImpNull        Gi0/0/0/1    10.0.0.22
10.100.0.30/32   24002   ImpNull        Gi0/0/0/2    10.0.0.26

RP/0/0/CPU0:P-2#
```

Nota. Ejemplo de reenvío de etiquetas. Elaboración propia, realizado con EVE-NG.

6. SIMULACIÓN DE SERVICIOS VPN DE CAPA 3 SOBRE MPLS (L3VPN)

Los servicios de VPN de capa 3 están enfocados a clientes corporativos con la finalidad de proveer conexión entre sucursales a nivel nacional e internacional, permitiendo a los clientes separación de tráfico y aislamiento de rutas en una infraestructura de red compartida.

Las VPN de capa 3 sobre MPLS utiliza los tipos de red ya conocidos entre los cuales se encuentran del tipo punto a punto, hub and spoke y malla completa, para conectar los sitios se utiliza el protocolo BGP que distribuye toda la información relacionada con la VPN, por tal motivo este servicio permite a los clientes crecer en magnitud, así como reducir los costos de operación para sus empresas.

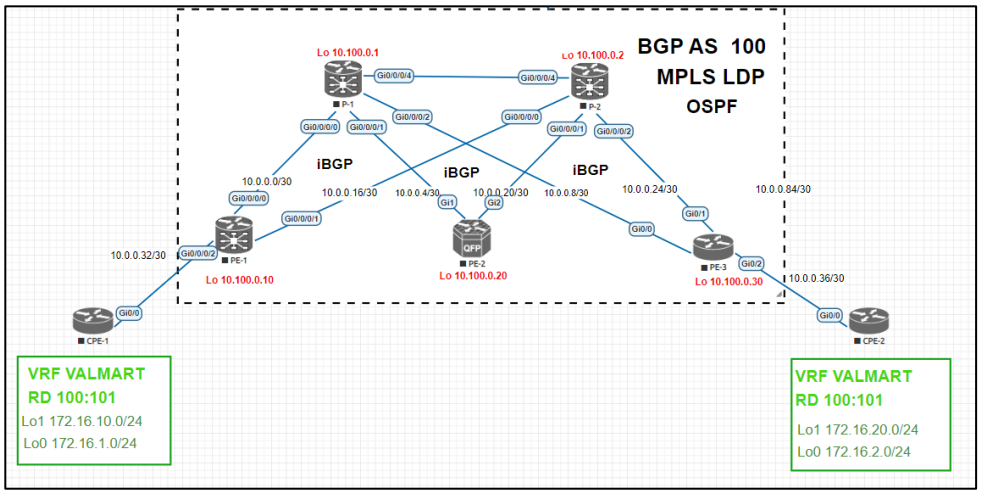
Siguiendo la topología planteada en el capítulo anterior, ya se cuenta con las sesiones de vecindad establecidas por los protocolos IGP, BGP y LDP sobre nuestra red ISP, por lo tanto, ya se pede realizar el diseño y las configuraciones correspondientes de los servicios de túnel de capa 3 sobre MPLS (L3VPN), por lo tanto, se estarán utilizando redes del tipo punto a punto para establecer la conexión entre los sitios remotos, por lo que se estarán verificando tres clientes con VRF distintas, para proporcionar conectividad mediante tres tipos de enrutamiento las cuales son rutas estáticas, OSPF Y BGP.

6.1. Configuración de CPEs con enrutamiento estático

En los entornos de MPLS los CPEs viene de las palabras en ingles *Customer Premises Equipment*, estos son enrutadores ubicados dentro de las instalaciones de los clientes que proporciona conectividad de la red LAN hacia la red MPLS, estos equipos son administrados por el proveedor.

Figura 101.

Topología de red del cliente Valmart



Nota. Ejemplo topología cliente Valmart. Elaboración propia, realizado con EVE-NG.

En el diseño de la figura 101 se observa los enrutadores CPE-1 y CPE-2 que estarán comunicándose entre ellos simulando una conexión entre redes LAN.

Para iniciar la configuración sobre el equipo CPE se detallan las siguientes normas generales.

- Configuración del segmento WAN utilizado para brindar conectividad hacia su PE correspondiente.

- Es necesario definir el protocolo de enrutamiento para el envío y recepción de rutas hacia los demás sitios remotos.

- Se añaden los segmentos IPv4 privados que representan los segmentos de la red interna, para luego realizar pruebas de conectividad hacia el otro equipo CPE.

Se inician las configuraciones sobre el enrutador CPE-1 ubicado en las instalaciones del cliente, estas configuraciones se comparten a continuación.

Figura 102.

Direccionamiento IPv4 sobre CPE-1

```
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface Loopback1
 ip address 172.16.10.1 255.255.255.0
!
interface GigabitEthernet0/0
 ip address 10.0.0.34 255.255.255.252
!
```

Nota. Ejemplo de direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

Para simular el direccionamiento interno del cliente se utilizan las interfaces de loopback del CPE.

Figura 103.

Direccionamiento IPv4 sobre CPE-2

```
!
interface Loopback0
 ip address 172.16.2.1 255.255.255.0
!
interface Loopback1
 ip address 172.16.20.1 255.255.255.0
!
interface GigabitEthernet0/0
 ip address 10.0.0.38 255.255.255.252
!
```

Nota. Ejemplo de direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

El protocolo de enrutamiento a utilizar para brindar la conectividad entre los CPEs será de rutas estáticas, las cuales se configuran a continuación.

Figura 104.

Rutas estáticas en CPE-1

```
!
ip route 172.16.2.0 255.255.255.0 10.0.0.33
ip route 172.16.20.0 255.255.255.0 10.0.0.33
!
```

Nota. Ejemplo de rutas estáticas. Elaboración propia, realizado con EVE-NG.

Dentro de cada CPE se añaden las rutas estáticas con el direccionamiento LAN del CPE ubicado en el otro extremo con IP de siguiente salto hacia su PE correspondiente.

Figura 105.

Rutas estáticas en CPE-2

```
!
ip route 172.16.1.0 255.255.255.0 10.0.0.37
ip route 172.16.10.0 255.255.255.0 10.0.0.37
!
```

Nota. Ejemplo de rutas estáticas. Elaboración propia, realizado con EVE-NG.

En este momento ya se tiene la configuración necesaria dentro de los equipos CPEs que permitirán la comunicación entre sí, pero aún falta la creación del túnel que permitirá la conexión completa de los equipos.

6.2. Configuración de VRF con enrutamiento estático

Actualmente los equipos CPEs están con las configuraciones mínimas para poder comunicarse entre sí, pero aun se necesita crear el túnel L3VPN entre el equipo PE-1 y PE-3 para que pueda completar la comunicación entre los sitios, para esto se toman en cuenta las siguientes consideraciones.

- Para lograr la conectividad de extremo a extremo ambos equipos PE deben de tener configurado las VRF correspondientes de los clientes que necesitan transportar.

- Configurar los RD que identificarán las VRF sobre la red ISP y es necesario definir los RT que importarán y exportarán sobre los PEs.

- Se crea sobre BGP el túnel donde se transporten las VRF seleccionadas, definiendo una sesión VPNv4 con los equipos PE-1 y PE-3, que permita intercambiar tráfico proveniente de una VRF.

Teniendo en cuenta lo mencionado anteriormente se añaden los comandos necesarios sobre PE-1 y PE-3 para habilitar la VRF del primer cliente, en este caso al seleccionar el cliente de Valmart que es una cadena de sucursales que necesita establecer comunicación con todos los sitios ubicados en distintas zonas geográficas, Por lo tanto, se crea la VRF con nombre de VALMART y dentro de la familia de direcciones IPv4 unicast se importa y exporta el RD correspondiente al cliente, en este caso se asigna con el RD 100:101, siguiendo el formato recomendado de configuración.

Figura 106.

Configuración de VRF en PE-1

```
!
vrf VALMART
 address-family ipv4 unicast
  import route-target
   100:101
  !
  export route-target
   100:101
  !
 !
!
```

Nota. Ejemplo de configuración VRF. Elaboración propia, realizado con EVE-NG.

Figura 107.

Configuración de VRF en PE-3

```
 !
vrf definition VALMART
 rd 100:101
 !
 address-family ipv4
  route-target export 100:101
  route-target import 100:101
 exit-address-family
!
```

Nota. Ejemplo de configuración VRF. Elaboración propia, realizado con EVE-NG.

Ahora se asignan las interfaces físicas de los PEs en donde existirá la conexión de la VRF del cliente hacia los CPEs, en este caso se observa en la topología de la figura 101 que las interfaces que corresponden son las Gi0/0/0/2 y Gi0/2, por lo que agrega la VRF correspondiente.

Figura 108.

Asignación de VRF en PE-1

```
!
interface GigabitEthernet0/0/0/2
 vrf VALMART
 ipv4 address 10.0.0.33 255.255.255.252
!
```

Nota. Ejemplo de asignación VRF. Elaboración propia, realizado con EVE-NG.

Figura 109.

Asignación de VRF en PE-3

```
!
interface GigabitEthernet0/2
 vrf forwarding VALMART
 ip address 10.0.0.37 255.255.255.252
!
```

Nota. Ejemplo de asignación VRF. Elaboración propia, realizado con EVE-NG.

A continuación, se aplican rutas estáticas en los PEs para que se pueda completar la comunicación de forma bidireccional entre PE y CPE.

Figura 110.

Rutas estáticas en PE-1

```
!
router static
 vrf VALMART
  address-family ipv4 unicast
   172.16.1.0/24 10.0.0.34
   172.16.10.0/24 10.0.0.34
  !
 !
!
```

Nota. Ejemplo de rutas estáticas. Elaboración propia, realizado con EVE-NG.

Figura 111.

Rutas estáticas en PE-3

```
!
ip route vrf VALMART 172.16.2.0 255.255.255.0 10.0.0.38
ip route vrf VALMART 172.16.20.0 255.255.255.0 10.0.0.38
!
```

Nota. Ejemplo de rutas estáticas. Elaboración propia, realizado con EVE-NG.

Llegado a este punto de implementación sobre el cliente de VALMART se cuenta actualmente con las configuraciones para que pueda existir la comunicación de PE-1 hacia CPE-1 de igual forma en el otro punto la comunicación existe para PE-3 hacia CPE-2, aún falta la comunicación de extremo a extremo que logre comunicar de CPE-1 hacia CPE-2, para esto se introducirá una nueva sesión sobre BGP del tipo VPNV4 que dará el transporte a todos los prefijos que son enviados sobre una VRF anunciándolo con las etiquetas de RD y RT.

Para crear las sesiones del tipo VPNV4 se debe ir hacia la red MPLS de la red que se ha configurado con anterioridad y se debe asignar sobre la configuración de BGP la introducción de una nueva familia de direccionamiento del tipo VPNv4 unicast, como primer paso sobre los equipos P se ingresa en el modo de configuración global y dentro del proceso de BGP se habilita la familia de direccionamiento VPNV4 a cada vecino; adicional a esto se escoge que los enrutadores P sigan siendo reflectores de ruta, así como fueron seleccionados con anterioridad dentro de la familia de IPv4 unicast.

Figura 112.

VPNV4 sobre BGP en P-1

```
!
router bgp 100
 !
 address-family vpnv4 unicast
 !
 neighbor 10.100.0.10
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 neighbor 10.100.0.20
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 !
 neighbor 10.100.0.30
  address-family vpnv4 unicast
   route-reflector-client
  !
```

Nota. Ejemplo de VPNV4/BGP. Elaboración propia, realizado con EVE-NG.

Figura 113.

VPNV4 sobre BGP en P-2

```
!
router bgp 100
 !
 address-family vpnv4 unicast
 !
 neighbor 10.100.0.10
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
!
 neighbor 10.100.0.20
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
!
 neighbor 10.100.0.30
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
!
```

Nota. Ejemplo de VPNV4/BGP. Elaboración propia, realizado con EVE-NG.

Figura 114.

VPNV4 sobre BGP en PE-1

```
!
router bgp 100
 !
 address-family vpnv4 unicast
 !
 neighbor 10.100.0.1
  address-family vpnv4 unicast
  !
 !
!
 neighbor 10.100.0.2
  address-family vpnv4 unicast
  !
 !
!
```

Nota. Ejemplo de VPNV4/BGP. Elaboración propia, realizado con EVE-NG.

Figura 115.

VPNV4 sobre BGP en PE-2

```
!
router bgp 100
 !
 address-family vpnv4
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.1 send-community extended
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community extended
 !
```

Nota. Ejemplo de VPNV4/BGP. Elaboración propia, realizado con EVE-NG.

Figura 116.

VPNV4 sobre BGP en PE-3

```
!
router bgp 100
 !
 address-family vpnv4
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.1 send-community extended
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community extended
 !
!
```

Nota. Ejemplo de VPNV4/BGP. Elaboración propia, realizado con EVE-NG.

Para verificar que las configuraciones sobre la nueva familia de direccionamiento se encuentren correctamente aplicadas, se utilizan los comandos de verificación que se muestran a continuación.

Figura 117.

Verificación de sesión VPNv4 sobre BGP en P-1

```
RP/0/0/CPU0:P-1#show bgp vpnv4 unicast summary
Fri May 13 03:35:16.514 UTC
BGP router identifier 10.100.0.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 1
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               1          1          1          1           1           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.100.0.10       0   100     266     285        1    0    0 00:01:00          0
10.100.0.20       0   100     306     293        1    0    0 00:04:50          0
10.100.0.30       0   100     283     278        1    0    0 00:04:50          0

RP/0/0/CPU0:P-1#
```

Nota. Ejemplo de verificación VPNV4/BGP. Elaboración propia, realizado con EVE-NG.

Figura 118.

Verificación de sesión VPNv4 sobre BGP en P-2

```
RP/0/0/CPU0:P-2#show bgp vpnv4 unicast summary
Fri May 13 03:35:42.652 UTC
BGP router identifier 10.100.0.2, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 1
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker               1          1          1          1           1           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.100.0.10       0   100     245     262        1    0    0 00:01:32          0
10.100.0.20       0   100     297     283        1    0    0 00:13:08          0
10.100.0.30       0   100     277     266        1    0    0 00:13:09          0

RP/0/0/CPU0:P-2#
```

Nota. Ejemplo de verificación VPNV4/BGP. Elaboración propia, realizado con EVE-NG.

En este momento se tienen las configuraciones necesarias en cada sitio remoto para establecer la comunicación entre PE, CPE. Además las sesiones VPNv4 ya se encuentran establecidas, pero si se observa, en las pruebas anteriores se verifica que la cantidad de prefijos aprendidos aún permanecen en cero esto significa que desde la perspectiva de los equipos P los enrutadores PE-1 y PE-3 aun no comparten los prefijos aprendidos de parte de los CPEs aunque ya se tengan configuradas las VRF en ambos PE, aun así, si se consulta sobre la tabla de enrutamiento sobre los PEs se tienen las siguientes tablas donde no conocen las rutas para llegar a los sitios remotos conociendo únicamente las rutas estáticas previamente configuradas.

Figura 119.

Enrutamiento sobre PE-1

```
RP/0/0/CPU0:PE-1#SHOW IP ROUTE VRF VALMART
Fri May 13 02:50:28.200 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

C    10.0.0.32/30 is directly connected, 00:18:46, GigabitEthernet0/0/0/2
L    10.0.0.33/32 is directly connected, 00:18:46, GigabitEthernet0/0/0/2
S    172.16.1.0/24 [1/0] via 10.0.0.34, 00:00:17
S    172.16.10.0/24 [1/0] via 10.0.0.34, 00:00:17
RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de enrutamiento. Elaboración propia, realizado con EVE-NG.

Figura 120.

Enrutamiento sobre PE-3

```
PE-3#show ip route vrf VALMART

Routing Table: VALMART
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.36/30 is directly connected, GigabitEthernet0/2
L        10.0.0.37/32 is directly connected, GigabitEthernet0/2
      172.16.0.0/24 is subnetted, 2 subnets
S        172.16.2.0 [1/0] via 10.0.0.38
S        172.16.20.0 [1/0] via 10.0.0.38
PE-3#
```

Nota. Ejemplo de enrutamiento. Elaboración propia, realizado con EVE-NG.

Ahora para completar la comunicación de extremo a extremo es necesario que BGP aplique una redistribución de los prefijos aprendidos de parte de los PEs en donde se encuentre la VRF instalada, en otras palabras, el protocolo BGP es el encargado de dar transporte a los prefijos entre los equipos PEs hasta completar la comunicación.

Para aplicar la redistribución sobre los prefijos que interesa, se ingresa al proceso de BGP para cada uno de los equipos de borde y se aplican lo siguientes comandos.

Figura 121.

Redistribución BGP de rutas estáticas en PE-1

```
!
router bgp 100
  address-family vpnv4 unicast
  !
 vrf VALMART
  rd 100:101
  address-family ipv4 unicast
   redistribute connected
   redistribute static
  !
```

Nota. Ejemplo de redistribución BGP. Elaboración propia, realizado con EVE-NG.

La primera línea de redistribución significa que tomara todos los prefijos que se encuentren directamente conectadas sobre la VRF del enrutador local mientras que la segunda línea de redistribución tomara todos los prefijos instalados por rutas estáticas y serán enviadas por BGP hacia los demás equipos que soliciten la información mediante los RD y RT asignados.

Figura 122.

Redistribución BGP de rutas estáticas en PE-3

```
router bgp 100
 !
 address-family ipv4 vrf VALMART
  redistribute connected
  redistribute static
 exit-address-family
```

Nota. Ejemplo de redistribución BGP. Elaboración propia, realizado con EVE-NG.

Por lo tanto, si se verifica la sesión VPNv4 en el PE-1 se observa que ya se conoce tres prefijos provenientes del PE-3.

Figura 123.

Verificación de sesión VPNv4 sobre BGP en P-1

```
RP/0/0/CPU0:PE-1#show bgp vpnv4 unicast summary
Fri May 13 04:18:59.387 UTC
BGP router identifier 10.100.0.10, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 12
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker              12         12         12         12          12           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.100.0.1        0   100     206     193       12    0    0 00:44:44          3
10.100.0.2        0   100     204     192       12    0    0 00:44:49          3

RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de verificación de sesión. Elaboración propia, realizado con EVE-NG.

Ahora en la información sobre los prefijos recibidos, se verifica que los prefijos enviados por PE-3 se encuentran con sus respectivos atributos, en este punto ya existe información compartida entre PE-1 Y PE-3, esto se verifica con los siguientes comandos.

Figura 124.

Prefijos recibidos sobre la VRF en PE-1

```
RP/0/0/CPU0:PE-1#show bgp vpnv4 unicast
Fri May 13 04:19:07.206 UTC
BGP router identifier 10.100.0.10, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 12
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:101 (default for vrf VALMART)
*> 10.0.0.32/30     0.0.0.0                  0         32768 ?
*>i10.0.0.36/30     10.100.0.30              0    100      0 ?
* i                  10.100.0.30              0    100      0 ?
*>i172.16.2.0/24    10.100.0.30              0    100      0 ?
* i                  10.100.0.30              0    100      0 ?
*>i172.16.20.0/24   10.100.0.30              0    100      0 ?
* i                  10.100.0.30              0    100      0 ?

Processed 4 prefixes, 7 paths
RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de prefijos recibidos. Elaboración propia, realizado con EVE-NG.

Figura 125.

Prefijos recibidos sobre la VRF en PE-3

```
PE-3#show bgp vpnv4 unicast all
BGP table version is 15, local router ID is 10.100.0.30
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:101 (default for vrf VALMART)
 * i 10.0.0.32/30     10.100.0.10              0    100      0 ?
 *>i                  10.100.0.10              0    100      0 ?
 *>  10.0.0.36/30     0.0.0.0                  0         32768 ?
 * i 172.16.1.0/24    10.100.0.10              0    100      0 ?
 *>i                  10.100.0.10              0    100      0 ?
 *>  172.16.2.0/24    10.0.0.38                0         32768 ?
 * i 172.16.10.0/24   10.100.0.10              0    100      0 ?
 *>i                  10.100.0.10              0    100      0 ?
 *>  172.16.20.0/24   10.0.0.38                0         32768 ?
PE-3#
```

Nota. Ejemplo de prefijos recibidos. Elaboración propia, realizado con EVE-NG.

Finalizados los pasos anteriores se verifica mediante una prueba de conectividad si CPE-1 logra tener comunicación hacia la otra sucursal, se aplica la prueba mediante el protocolo ICMP y el resultado muestra que existe comunicación de forma exitosa finalizando las configuraciones de VRF con rutas estáticas.

Figura 126.

Prueba de conectividad hacia CPE-2

```
CPE-1#ping 172.16.20.1 source Loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 13/13/14 ms
CPE-1#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 127.

Prueba de conectividad hacia CPE-1

```
CPE-2#ping 172.16.1.1 source loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.2.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 13/13/14 ms
CPE-2#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 128.

Traza hacia CPE-1

```
CPE-1#tracer 172.16.20.1 source Loopback1
Type escape sequence to abort.
Tracing the route to 172.16.20.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.33 9 msec 5 msec 3 msec
  2 10.0.0.17 [MPLS: Labels 24002/26 Exp 0] 14 msec 15 msec 13 msec
  3 10.0.0.37 [MPLS: Label 26 Exp 0] 14 msec 14 msec 14 msec
  4 10.0.0.38 15 msec 18 msec *
CPE-1#
```

Nota. Ejemplo de traza hacia CPE. Elaboración propia, realizado con EVE-NG.

Mediante la traza se comprueba la ruta que sigue el paquete hasta encontrar su destino, adicional se observa el etiquetado que asigna la red MPLS.

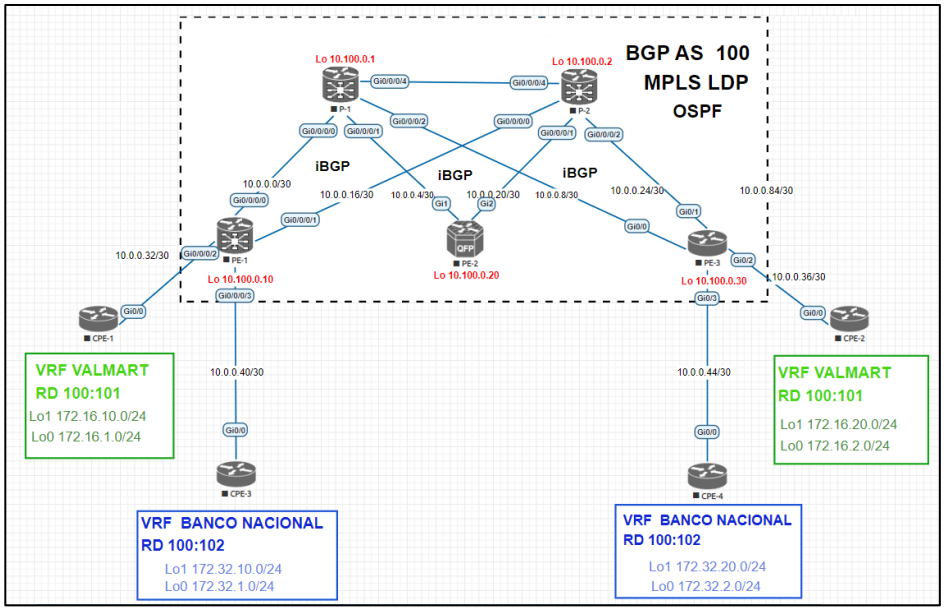
6.3. Configuración de CPEs con enrutamiento OSPF

En seguimiento con el proceso de configuración de nuestra red ISP se introduce el siguiente diseño que conforman los equipos CPE-3 y CPE-4 donde el cliente desea comunicar todos los puntos mediante un protocolo de enrutamiento dinámico con el objetivo de mantener un proceso de enrutamiento de forma automática en comparación con las rutas estáticas del caso anterior.

Por lo tanto, para dar inicio con las configuraciones sobre los equipos CPE-3 y CPE-4 se define el direccionamiento WAN hacia los equipos de borde, por lo que se observa la interfaz G0/0 para establecer la conexión física hacia los PEs correspondientes, de igual forma para simular las redes internas de los clientes se utilizan las interfaces de loopback 0 y 1.

Figura 129.

Topología de red del cliente Banco Nacional



Nota. Ejemplo de topología de red cliente Banco Nacional. Elaboración propia, realizado con EVE-NG.

Figura 130.

Direccionamiento IPv4 sobre CPE-3

```
!
interface Loopback0
 ip address 172.32.1.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback1
 ip address 172.32.10.1 255.255.255.0
 ip ospf network point-to-point
!
interface GigabitEthernet0/0
 ip address 10.0.0.42 255.255.255.252
!
```

Nota. Ejemplo de direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

Figura 131.

Direccionamiento IPv4 sobre CPE-4

```
!
interface Loopback0
 ip address 172.32.2.1 255.255.255.0
 ip ospf network point-to-point
!
interface Loopback1
 ip address 172.32.20.1 255.255.255.0
 ip ospf network point-to-point
!
interface GigabitEthernet0/0
 ip address 10.0.0.46 255.255.255.252
!
```

Nota. Ejemplo de direccionamiento IPv4. Elaboración propia, realizado con EVE-NG.

A continuación, se crea una adyacencia OSPF hacia cada uno de los PEs que se comparte a nivel WAN, por lo tanto, sobre el proceso se añaden los segmentos que se necesitan enviar a través de OSPF las cuales se encuentran configuradas sobre las interfaces de loopback y el segmento hacia los PEs para lograr la vecindad con el equipo, el identificador de cada CPE puede variar conforme el criterio del administrador de la red, de igual forma el número de área.

Figura 132.

Proceso OSPF en CPE-3

```
!
router ospf 102
 router-id 3.3.3.3
 network 10.0.0.40 0.0.0.3 area 0
 network 172.32.1.0 0.0.0.255 area 0
 network 172.32.10.0 0.0.0.255 area 0
!
```

Nota. Ejemplo de proceso OSPF. Elaboración propia, realizado con EVE-NG.

Figura 133.

Proceso OSPF en CPE-4

```
!
router ospf 102
 router-id 4.4.4.4
 network 10.0.0.44 0.0.0.3 area 0
 network 172.32.2.0 0.0.0.255 area 0
 network 172.32.20.0 0.0.0.255 area 0
!
```

Nota. Ejemplo de proceso OSPF. Elaboración propia, realizado con EVE-NG.

De esta forma se finalizan las configuraciones mínimas para permitir la comunicación entre los sitios remotos, luego se procede a completar las configuraciones restantes sobre los PE hasta levantar una sesión con el protocolo OSPF con su respectivo túnel L3VPN.

6.4. Configuración de VRF con enrutamiento OSPF

En este nuevo escenario se presenta el siguiente cliente de Banco Nacional que es una entidad financiera que se encuentra en todos los departamentos de un país, además por la cantidad de agencias distribuidas a lo largo de una región no es eficiente la gestión de las redes mediante la aplicación de rutas estáticas, por esta razón la implementación de un protocolo dinámico facilita la administración de parte del cliente en todas las agencias conectadas, de esta manera se estará utilizando el protocolo OSPF para implementar todo el proceso de enrutamiento del cliente.

Para comenzar con la implementación del cliente se define los parámetros que se estará utilizando para la VRF, siguiendo el mismo formato del cliente anterior se configura de la siguiente manera.

Figura 134.

Configuración de VRF sobre PE-1

```
!
vrf BANCONACIONAL
 address-family ipv4 unicast
  import route-target
   100:102
  !
  export route-target
   100:102
  !
```

Nota. Ejemplo de configuración VRF. Elaboración propia, realizado con EVE-NG.

Sobre la configuración global de la VRF de define el nombre como BANCONACIONAL y sobre la familia de direccionamiento se aplica el valor de RD 100:102 y los parámetros de importación y exportación aplicandose tanto en el PE-1 como en el PE-3.

Figura 135.

Configuración de VRF sobre PE-3

```
!
vrf definition BANCONACIONAL
 rd 100:102
 !
 address-family ipv4
  route-target export 100:102
  route-target import 100:102
 exit-address-family
!
```

Nota. Ejemplo de configuración VRF. Elaboración propia, realizado con EVE-NG.

Ya con la VRF definida sobre los equipos de borde se continua con las configuraciones, ahora sobre las interfaces físicas que estarán compartiendo la conexión sobre la capa 1 hacia los sitios remotos del cliente, en este caso se utiliza una interfaz individual sobre los PEs, que también es factible utilizar subinterfaces, o interfaces virtuales, de esta forma se consigue optimizar los recursos de hardware en los entornos reales.

Sobre esta simulación se seleccionan dos interfaces las cuales serán parte del dominio de la VRF las cuales se configuran a continuación.

Figura 136.

Configuración de VRF sobre la interfaz de PE-1

```
!
interface GigabitEthernet0/0/0/3
 vrf BANCONACIONAL
 ipv4 address 10.0.0.41 255.255.255.252
!
```

Nota. Ejemplo de configuración VRF. Elaboración propia, realizado con EVE-NG.

Figura 137.

Configuración de VRF sobre la interfaz de PE-3

```
!
interface GigabitEthernet0/3
 vrf forwarding BANCONACIONAL
 ip address 10.0.0.45 255.255.255.252
!
```

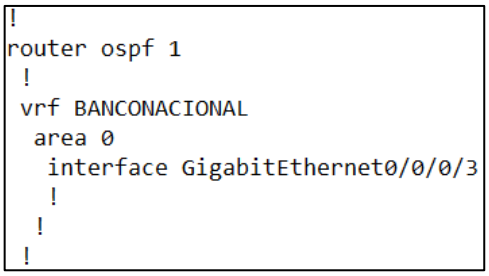
Nota. Ejemplo de configuración VRF. Elaboración propia, realizado con EVE-NG.

En este momento ya se tienen creadas las VRF sobre los enrutadores de borde, pero aun es necesario terminar de completar la vecindad OSPF, de esta forma habrá una sesión sobre el protocolo dinámico entre los sitios remotos con los PEs y CPEs, ahora entre los enrutadores PE-1 y PE-3 se estará enviándo por el túnel L3VPN sobre la red MPLS hasta completar la comunicación entre CPE-3 Y CPE-4.

Por lo tanto, se debe completar las configuraciones sobre OSPF con los siguientes comandos sobre los PEs.

Figura 138.

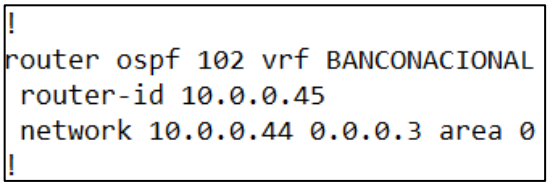
Proceso OSPF en PE-1



Nota. Ejemplo proceso OSPF. Elaboración propia, realizado con EVE-NG.

Figura 139.

Proceso OSPF en PE-3



Nota. Ejemplo proceso OSPF. Elaboración propia, realizado con EVE-NG.

Para comprobar las configuraciones sobre los enrutadores se utiliza el comando de comprobación de vecindad para verificar si la adyacencia fue completada de forma correcta.

Figura 140.

Verificación de vecindad entre PE-1 y CPE-3

```
RP/0/0/CPU0:PE-1#show ospf vrf  BANCONACIONAL neighbor
Wed May 18 04:26:43.851 UTC

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF 1, VRF BANCONACIONAL

Neighbor ID     Pri   State           Dead Time   Address         Interface
3.3.3.3         1     FULL/DR         00:00:31    10.0.0.42       GigabitEthernet0/0/0/3
    Neighbor is up for 00:01:14

Total neighbor count: 1
RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de verificación de vecindad. Elaboración propia, realizado con EVE-NG.

Figura 141.

Verificación de vecindad entre PE-3 y CPE-4

```
CPE-4#SHOW IP ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.0.0.45         1   FULL/BDR        00:00:34    10.0.0.45       GigabitEthernet0/0
CPE-4#
```

Nota. Ejemplo de verificación de vecindad. Elaboración propia, realizado con EVE-NG.

En este momento las configuraciones sobre la VRF de BANCONACIONAL ya se encuentran aplicadas permitiendo la comunicación entre los PEs y los CPEs, ahora si se realiza una prueba de conectividad entre los CPEs se observa que la prueba no se completa, falta la redistribución sobre BGP que da transporte

a todos los prefijos que comparten la misma VRF, para aplicar dicha redistribución es necesario tener en cuenta que ambos protocolos OSPF y BGP deben de compartir los prefijos de forma bidireccional por lo tanto desde el punto de vista del CPE estos envían sus prefijos mediante OSPF hacia su PE correspondiente para luego mandarlos sobre BGP hacia los diferentes sitios, esperando que la comunicación de extremo a extremo sea completada, desde la perspectiva del PE este recibe los prefijos desde los sitios remotos sobre BGP y en este punto BGP debe de aplicar una nueva redistribución hacia OSPF para que pueda ser enviado hacia los CPEs, de esta forma se completan la redistribución de ambos protocolos sobre el mismo enrutador de borde.

Se comparte las configuraciones necesarias para lograr las distribuciones sobre los PE-1 y PE-3 sobre los protocolos de enrutamiento.

Figura 142.

Redistribución de OSPF a BGP sobre PE-1

```
!
router ospf 1
 !
 vrf BANCONACIONAL
  redistribute bgp 100
!
```

Nota. Ejemplo de redistribución OSPF a BGP. Elaboración propia, realizado con EVE-NG.

Sobre el proceso de OSPF se aplica la redistribución de BGP sobre el sistema autónomo 100.

Figura 143.

Redistribución de BGP a OSPF sobre PE-1

```
!
router bgp 100
 !
 neighbor 10.100.0.90
 vrf BANCONACIONAL
  rd 100:102
  address-family ipv4 unicast
   redistribute ospf 1 match internal external
  !
 !
```

Nota. Ejemplo de redistribución BGP a OSPF. Elaboración propia, realizado con EVE-NG.

Sobre el proceso de BGP se indica que, se enviarán todos los prefijos aprendidos de parte de OSPF sobre el proceso 1 y que coincidan todas las rutas aprendidas entre áreas, intra-áreas y externas.

Figura 144.

Redistribución de OSPF a BGP sobre PE-3

```
!
router ospf 102 vrf BANCONACIONAL
 router-id 10.0.0.45
 redistribute bgp 100 subnets
 network 10.0.0.44 0.0.0.3 area 0
!
```

Nota. Ejemplo de redistribución OSPF a BGP. Elaboración propia, realizado con EVE-NG.

Figura 145.

Redistribución de BGP a OSPF sobre PE-3

```
!
 address-family ipv4 vrf BANCONACIONAL
  redistribute ospf 102 match internal external 1 external 2
 exit-address-family
 !
```

Nota. Ejemplo de redistribución BGP a OSPF. Elaboración propia, realizado con EVE-NG.

Llegando en este punto ya se completaron todos los pasos para lograr la comunicación de extremo a extremo por lo tanto si se realiza una prueba de conectividad mediante el protocolo ICMP se observa que se completa con éxito, finalizando así la transmisión de datos mediante OSPF.

Figura 146.

Prueba de conectividad de CPE-4

```
CPE-4#ping 172.32.10.1 source Loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.32.10.1, timeout is 2 seconds:
Packet sent with a source address of 172.32.2.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 15/16/19 ms
CPE-4#
```

Nota. Ejemplo de prueba de conectividad CPE-4. Elaboración propia, realizado con EVE-NG.

Figura 147.

Prueba de conectividad de CPE-3

```
CPE-3#ping 172.32.2.0 source Loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.32.2.0, timeout is 2 seconds:
Packet sent with a source address of 172.32.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 15/16/17 ms
CPE-3#
```

Nota. Ejemplo de prueba de conectividad CPE-3. Elaboración propia, realizado con EVE-NG.

Figura 148.

Enrutamiento de la VRF sobre PE-1

```
RP/0/0/CPU0:PE-1#show ip route vrf BANCONACIONAL
Wed May 18 04:58:29.840 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

C    10.0.0.40/30 is directly connected, 00:33:03, GigabitEthernet0/0/0/3
L    10.0.0.41/32 is directly connected, 00:33:03, GigabitEthernet0/0/0/3
B    10.0.0.44/30 [200/0] via 10.100.0.30 (nexthop in vrf default), 00:00:06
O    172.32.1.0/24 [110/2] via 10.0.0.42, 00:20:51, GigabitEthernet0/0/0/3
B    172.32.2.0/24 [200/2] via 10.100.0.30 (nexthop in vrf default), 00:00:06
O    172.32.10.0/24 [110/2] via 10.0.0.42, 00:20:46, GigabitEthernet0/0/0/3
B    172.32.20.0/24 [200/2] via 10.100.0.30 (nexthop in vrf default), 00:00:06
RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de enrutamiento VRF/PE-1. Elaboración propia, realizado con EVE-NG.

Figura 149.

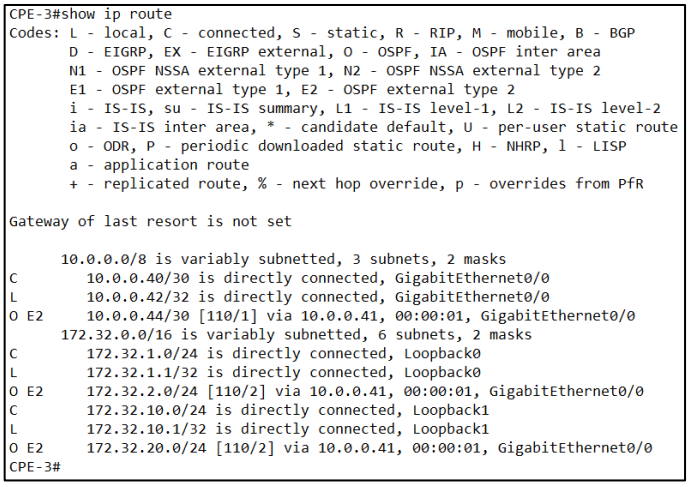
Enrutamiento de la VRF sobre PE-3

```
PE-3#show ip route vrf BANCONACIONAL

Routing Table: BANCONACIONAL
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B        10.0.0.40/30 [200/0] via 10.100.0.10, 00:00:59
C        10.0.0.44/30 is directly connected, GigabitEthernet0/3
L        10.0.0.45/32 is directly connected, GigabitEthernet0/3
      172.32.0.0/24 is subnetted, 4 subnets
B        172.32.1.0 [200/2] via 10.100.0.10, 00:00:59
O        172.32.2.0 [110/2] via 10.0.0.46, 00:19:25, GigabitEthernet0/3
B        172.32.10.0 [200/2] via 10.100.0.10, 00:00:59
O        172.32.20.0 [110/2] via 10.0.0.46, 00:19:35, GigabitEthernet0/3
PE-3#
```

Nota. Ejemplo de enrutamiento VRF/PE-3. Elaboración propia, realizado con EVE-NG.

Figura 150.

Enrutamiento de la VRF sobre CPE-3

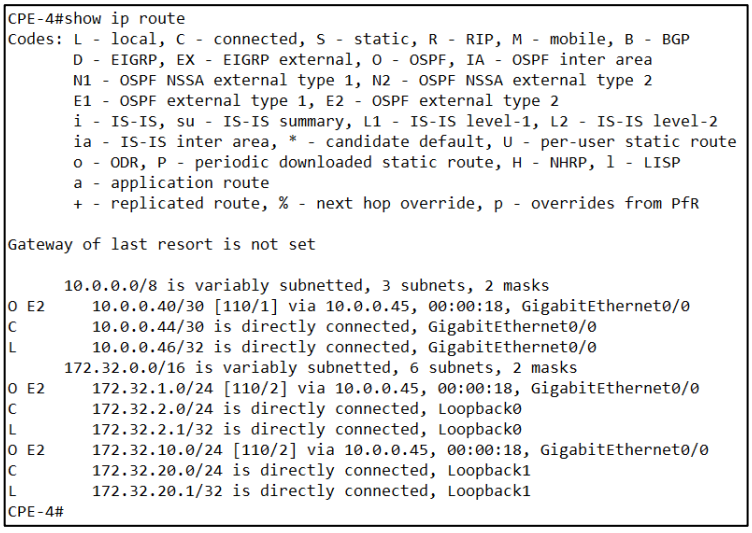


```
CPE-3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.40/30 is directly connected, GigabitEthernet0/0
L        10.0.0.42/32 is directly connected, GigabitEthernet0/0
O E2     10.0.0.44/30 [110/1] via 10.0.0.41, 00:00:01, GigabitEthernet0/0
      172.32.0.0/16 is variably subnetted, 6 subnets, 2 masks
C        172.32.1.0/24 is directly connected, Loopback0
L        172.32.1.1/32 is directly connected, Loopback0
O E2     172.32.2.0/24 [110/2] via 10.0.0.41, 00:00:01, GigabitEthernet0/0
C        172.32.10.0/24 is directly connected, Loopback1
L        172.32.10.1/32 is directly connected, Loopback1
O E2     172.32.20.0/24 [110/2] via 10.0.0.41, 00:00:01, GigabitEthernet0/0
CPE-3#
```

Nota. Ejemplo de enrutamiento VRF/CPE-3. Elaboración propia, realizado con EVE-NG.

Figura 151.

Enrutamiento de la VRF sobre CPE-4



```
CPE-4#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O E2     10.0.0.40/30 [110/1] via 10.0.0.45, 00:00:18, GigabitEthernet0/0
C        10.0.0.44/30 is directly connected, GigabitEthernet0/0
L        10.0.0.46/32 is directly connected, GigabitEthernet0/0
      172.32.0.0/16 is variably subnetted, 6 subnets, 2 masks
O E2     172.32.1.0/24 [110/2] via 10.0.0.45, 00:00:18, GigabitEthernet0/0
C        172.32.2.0/24 is directly connected, Loopback0
L        172.32.2.1/32 is directly connected, Loopback0
O E2     172.32.10.0/24 [110/2] via 10.0.0.45, 00:00:18, GigabitEthernet0/0
C        172.32.20.0/24 is directly connected, Loopback1
L        172.32.20.1/32 is directly connected, Loopback1
CPE-4#
```

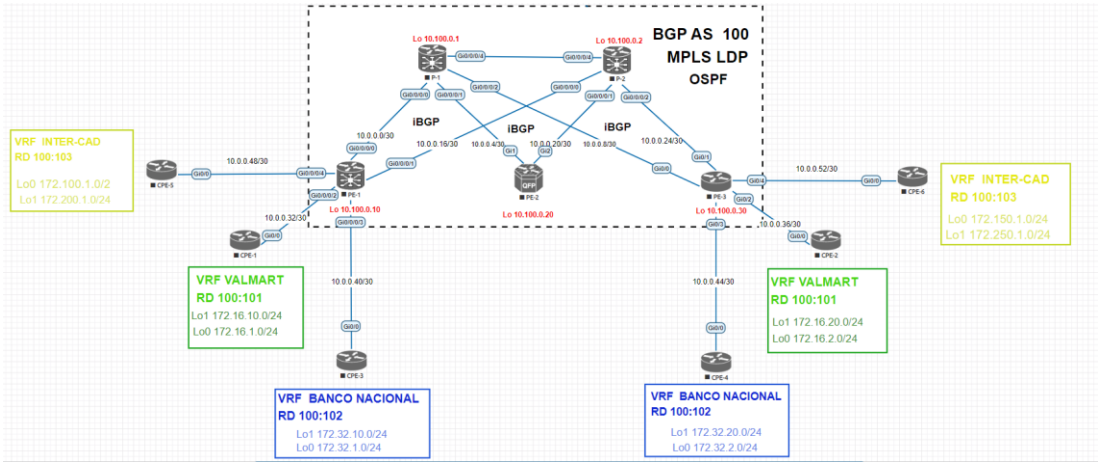
Nota. Ejemplo de enrutamiento VRF/CPE-4. Elaboración propia, realizado con EVE-NG.

6.5. Configuración de CPEs con enrutamiento BGP

Continuando con el proceso de simulación se agregan dos nuevos equipos como se indica en la siguiente topología de la figura 152, los CPE-5 y CPE-6 estos equipos se integran con el objetivo de analizar el comportamiento y configuración sobre el protocolo BGP a través de la red MPLS por lo tanto se necesita como primer paso las configuraciones sobre el direccionamiento IP sobre los equipos y levantar una sesión BGP para luego continuar con los procesos de VRF.

Figura 152.

Diseño de red sobre el cliente Intercad



Nota. Ejemplo de diseño de red cliente Intercad. Elaboración propia, realizado con EVE-NG.

Para iniciar con las configuraciones sobre los equipos CPE-5 y CPE-6 se define el direccionamiento WAN hacia los equipos PEs, por lo que se reserva la interfaz G0/0 para establecer la conexión física hacia los PEs que corresponden al diseño, de igual forma para simular la red interna del cliente en las sucursales se utiliza interfaces de loopback 0 y 1.

Figura 153.

Direccionamiento IP sobre CPE-5

```
!
interface Loopback0
 ip address 172.100.1.1 255.255.255.0
!
interface Loopback1
 ip address 172.200.1.1 255.255.255.0
!
interface GigabitEthernet0/0
 ip address 10.0.0.50 255.255.255.252
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Figura 154.

Direccionamiento IP sobre CPE-6

```
!
interface Loopback0
 ip address 172.150.1.1 255.255.255.0
!
interface Loopback1
 ip address 172.250.1.1 255.255.255.0
!
interface GigabitEthernet0/0
 ip address 10.0.0.54 255.255.255.252
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Los enrutadores CPE-5 y CPE-6 serán parte del propio sistema autónomo el cual pertenece el cliente por ser otra entidad que brinde servicios de conexión, utilizando el AS 100 de tránsito para ofrecer conectividad a sus propios suscriptores, en este caso se tiene el AS 60 del cliente en los sitios remotos y mediante una sesión eBGP se estarán compartiendo los prefijos entre AS, por lo tanto, se comparte las configuraciones básicas de los procesos de BGP utilizados en los CPEs.

Figura 155.

Proceso de BGP sobre CPE-5

```
!
router bgp 60
 bgp router-id 60.1.1.1
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.0.0.49 remote-as 100
 !
 address-family ipv4
  network 172.100.1.0 mask 255.255.255.0
  network 172.200.1.0 mask 255.255.255.0
  neighbor 10.0.0.49 activate
  neighbor 10.0.0.49 send-community both
 exit-address-family
!
```

Nota. Ejemplo de proceso BGP/CPE-5. Elaboración propia, realizado con EVE-NG.

Sobre los requisitos de configuración sobre el AS y el identificador del enrutador quedan al criterio del administrador del cliente, para este ejemplo se utilizó el AS 60 por ser otra entidad totalmente diferente a nuestra red ISP y se limita solo al intercambio de prefijos con su vecino en una sesión BGP de tipo externa, ahora se especifica con el comando network los prefijos que serán anunciados a nuestra red ISP y dentro de la configuración de la familia de direccionamiento se habilitan los atributos de las comunidades extendidas.

Figura 156.

Proceso de BGP sobre CPE-6

```
!
router bgp 60
 bgp router-id 70.1.1.1
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.0.0.53 remote-as 100
 !
 address-family ipv4
  network 172.150.1.0 mask 255.255.255.0
  network 172.250.1.0 mask 255.255.255.0
  neighbor 10.0.0.53 activate
  neighbor 10.0.0.53 send-community both
 exit-address-family
!
```

Nota. Ejemplo de proceso BGP/CPE-6. Elaboración propia, realizado con EVE-NG.

6.6. Configuración de VRF con enrutamiento BGP

Para el tercer ejemplo de simulación se presenta el cliente ISP Intercad Networks que presta servicios a otros clientes de forma nacional por lo que necesita un AS de tránsito para lograr brindar conectividad a un cliente en específico que se encuentra en otra región fuera de su cobertura y dentro del alcance del primer ISP. La principal ventaja del protocolo de BGP sobre los demás protocolos de enrutamiento se basa en la capacidad de compartir miles de prefijos y brindar transporte a las familias de direccionamiento como pueden ser IPv4, IPv6, VPNv4, Multicast, entre otros.

Teniendo en cuenta las configuraciones de los CPEs que fueron realizadas en la sección anterior se da inicio a la configuración de la VRF que se llamará con el nombre INTERCAD con los valores de RD 100:103 por ser el tercer

cliente del AS local, recordando que el valor de RD es el identificador único para cada VRF dentro del ISP, por lo que se comparte a continuación las configuraciones de VRF.

Figura 157.

Configuración de VRF en PE-1

```
!
vrf INTERCAD
 address-family ipv4 unicast
  import route-target
   100:103
  !
  export route-target
   100:103
  !
 !
!
```

Nota. Ejemplo de configuración de VRF-PE-1. Elaboración propia, realizado con EVE-NG.

Figura 158.

Configuración de VRF en PE-3

```
!
vrf definition INTERCAD
 rd 100:103
 !
 address-family ipv4
  route-target export 100:103
  route-target import 100:103
 exit-address-family
!
```

Nota. Ejemplo de configuración de VRF-PE-3. Elaboración propia, realizado con EVE-NG.

De igual forma que en los clientes anteriores se aparta una interfaz física para agregarlo al dominio de la VRF.

Figura 159.

Configuración de VRF sobre la interfaz en PE-1

```
!
interface GigabitEthernet0/0/0/3
 vrf BANCONACIONAL
 ipv4 address 10.0.0.41 255.255.255.252
!
```

Nota. Ejemplo de configuración de VRF/PE-1. Elaboración propia, realizado con EVE-NG.

Figura 160.

Configuración de VRF sobre la interfaz en PE-3

```
!
interface GigabitEthernet0/3
 vrf forwarding BANCONACIONAL
 ip address 10.0.0.45 255.255.255.252
!
```

Nota. Ejemplo de configuración de VRF/PE-3. Elaboración propia, realizado con EVE-NG.

Ya configuradas las VRF y las interfaces se procede con la configuración de BGP hacia los CPEs para completar la configuración de PE-CPE.

Figura 161.

Configuración de BGP sobre PE-1

```
!
router bgp 100
 !
 vrf INTERCAD
  rd 100:103
  bgp router-id 10.0.0.49
  address-family ipv4 unicast
  !
  neighbor 10.0.0.50
   remote-as 60
   address-family ipv4 unicast
    route-policy prefix-ipv4 in
    route-policy prefix-ipv4 out
   !
  !
 !
```

Nota. Ejemplo de configuración BGP/PE-1. Elaboración propia, realizado con EVE-NG.

Si se observa la configuración del enrutador XR se ven las siguientes líneas route-policy prefix-ipv4 in/out este comando que traducido significa políticas de enrutamiento, indica a los enrutadores tomar decisiones para anunciar, distribuir y modificar los prefijos en función de la política de enrutamiento previamente configurado, una particularidad sobre los enrutadores XR es que no permite el envío y recepción de prefijos sobre una sesión BGP de tipo externa si no se encuentran los prefijos a distribuir sobre una política de enrutamiento, de los contrario son descartados.

En este caso para permitir la distribución de los prefijos se crea la política de enrutamiento con nombre prefix-ipv4 con una sola regla que permita todos los prefijos entrantes y salientes hacia el vecino.

Figura 162.

Políticas de enrutamiento sobre PE-1

```
!
route-policy prefix-ipv4
  pass
end-policy
!
```

Nota. Ejemplo de enrutamiento/PE-1. Elaboración propia, realizado con EVE-NG.

Sobre esta simulación se configura de tal manera que permita todos los prefijos sin aplicar filtros sobre el enrutamiento, en entornos reales se declaran sobre un proceso de filtrado para mantener el control y evitar así problemas de enrutamiento.

Figura 163.

Configuración de BGP sobre PE-3

```
!
router bgp 100
 !
 address-family ipv4 vrf INTERCAD
  bgp router-id 10.0.0.53
  neighbor 10.0.0.54 remote-as 60
  neighbor 10.0.0.54 activate
  neighbor 10.0.0.54 send-community both
 exit-address-family
 !
```

Nota. Ejemplo de configuración BGP/PE-3. Elaboración propia, realizado con EVE-NG.

Para verificar que las configuraciones sobre BGP de tipo externo fueron correctamente configuradas se ejecuta nuevamente a los comandos de verificación de sesión.

Figura 164.

Verificación de sesión BGP en CPE-5

```
CPE-5#show ip bgp ipv4 unicast  summary
BGP router identifier 60.1.1.1, local AS number 60
BGP table version is 5, main routing table version 5
4 network entries using 576 bytes of memory
4 path entries using 320 bytes of memory
2/2 BGP path/bestpath attribute entries using 304 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1224 total bytes of memory
BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.49       4          100       7       8        5    0    0 00:03:59        2
CPE-5#
```

Nota. Ejemplo de verificación BGP. Elaboración propia, realizado con EVE-NG.

Figura 165.

Verificación de sesión de BGP en CPE-6

```
CPE-6#show bgp ipv4 unicast  summary
BGP router identifier 70.1.1.1, local AS number 60
BGP table version is 5, main routing table version 5
4 network entries using 576 bytes of memory
4 path entries using 320 bytes of memory
2/2 BGP path/bestpath attribute entries using 304 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1248 total bytes of memory
BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.53       4          100      10      10        5    0    0 00:05:08        2
CPE-6#
```

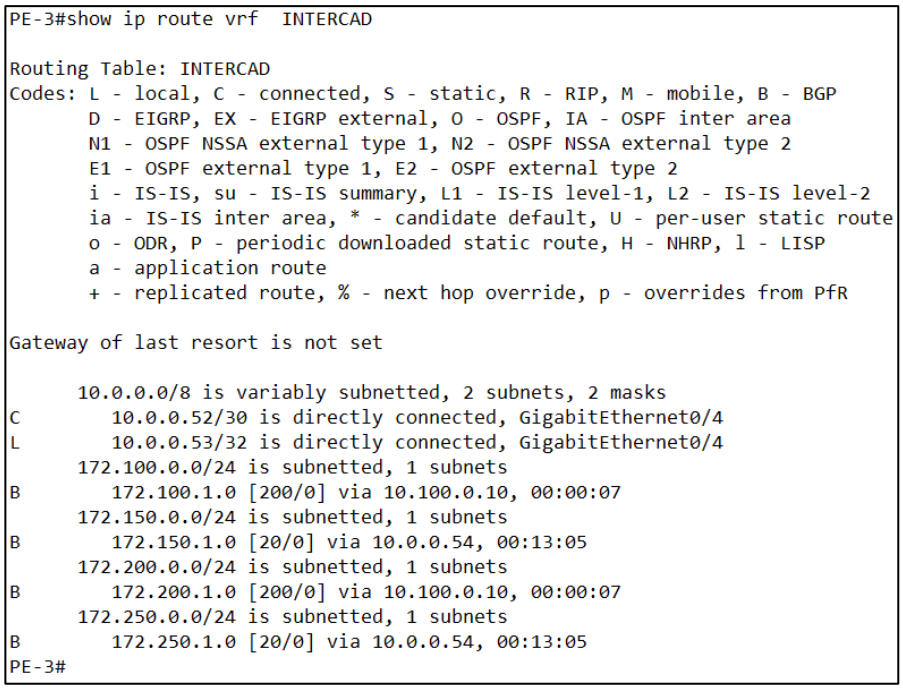
Nota. Ejemplo de verificación BGP. Elaboración propia, realizado con EVE-NG.

Con los comandos de BGP se nota que los prefijos enviados por CPE-5 ya se conocen en la tabla de enrutamiento de PE-3, debido que se utiliza el mismo proceso de BGP para crear la vecindad, por ende, ya no se necesita comandos adicionales de redistribución como lo era con los protocolos anteriores, de esta forma todos los prefijos anunciados por los CPEs se compartirán por toda la VRF permitida y sobre los que cumplan con los criterios de importación y exportación de la VRF.

Dicho lo anterior si se analizan las tablas de enrutamiento sobre ambos PE ya existen los prefijos externos como se muestran a continuación.

Figura 166.

Enrutamiento sobre PE-3

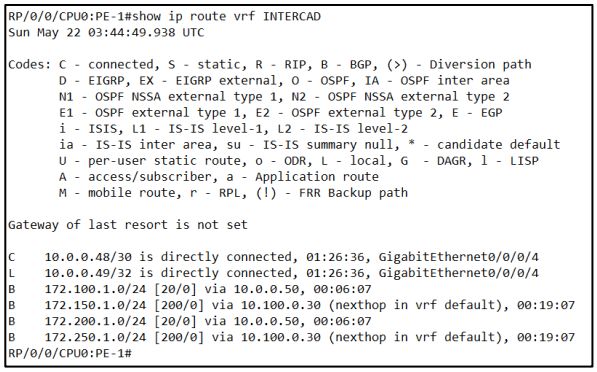


```
PE-3#show ip route vrf  INTERCAD

Routing Table: INTERCAD
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.52/30 is directly connected, GigabitEthernet0/4
L        10.0.0.53/32 is directly connected, GigabitEthernet0/4
      172.100.0.0/24 is subnetted, 1 subnets
B        172.100.1.0 [200/0] via 10.100.0.10, 00:00:07
      172.150.0.0/24 is subnetted, 1 subnets
B        172.150.1.0 [20/0] via 10.0.0.54, 00:13:05
      172.200.0.0/24 is subnetted, 1 subnets
B        172.200.1.0 [200/0] via 10.100.0.10, 00:00:07
      172.250.0.0/24 is subnetted, 1 subnets
B        172.250.1.0 [20/0] via 10.0.0.54, 00:13:05
PE-3#
```

Nota. Ejemplo de enrutamiento PE-3. Elaboración propia, realizado con EVE-NG.

Figura 167.

Enrutamiento sobre PE-1



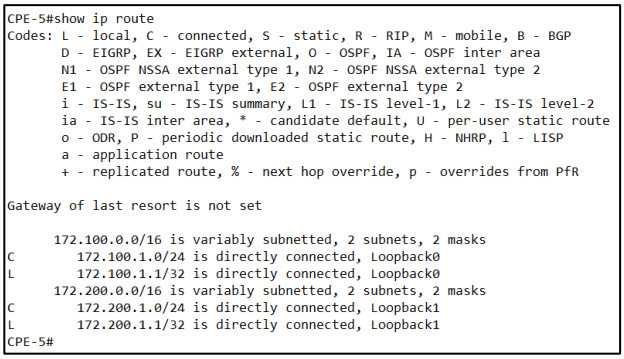
```
RP/0/0/CPU0:PE-1#show ip route vrf INTERCAD
Sun May 22 03:44:49.938 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

C    10.0.0.48/30 is directly connected, 01:26:36, GigabitEthernet0/0/0/4
L    10.0.0.49/32 is directly connected, 01:26:36, GigabitEthernet0/0/0/4
B    172.100.1.0/24 [20/0] via 10.0.0.50, 00:06:07
B    172.150.1.0/24 [200/0] via 10.100.0.30 (nexthop in vrf default), 00:19:07
B    172.200.1.0/24 [20/0] via 10.0.0.50, 00:06:07
B    172.250.1.0/24 [200/0] via 10.100.0.30 (nexthop in vrf default), 00:19:07
RP/0/0/CPU0:PE-1#
```

Nota. Ejemplo de enrutamiento PE-1. Elaboración propia, realizado con EVE-NG.

Si se realizan las pruebas de conectividad correspondientes entre CPE-5 hacia las direcciones de loopback de CPE-6 se observa que la comunicación no se completa con éxito, adicional si se consultan las tablas de enrutamiento sobre cada CPE se observa que existe solo las entradas hacia las rutas locales.

Figura 168.

Enrutamiento en CPE-5



```
CPE-5#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      172.100.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.100.1.0/24 is directly connected, Loopback0
L        172.100.1.1/32 is directly connected, Loopback0
      172.200.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.200.1.0/24 is directly connected, Loopback1
L        172.200.1.1/32 is directly connected, Loopback1
CPE-5#
```

Nota. Ejemplo de enrutamiento CPE-5. Elaboración propia, realizado con EVE-NG.

Figura 169.

Enrutamiento en CPE-6

```
CPE-6#show ip route
*Oct  6 23:42:57.944: %SYS-5-CONFIG_I: Configured from console by consolete
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      172.150.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.150.1.0/24 is directly connected, Loopback0
L        172.150.1.1/32 is directly connected, Loopback0
      172.250.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.250.1.0/24 is directly connected, Loopback1
L        172.250.1.1/32 is directly connected, Loopback1
CPE-6#
```

Nota. Ejemplo de enrutamiento CPE-6. Elaboración propia, realizado con EVE-NG.

Figura 170.

Prueba de conectividad en CPE-5

```
CPE-5#ping 172.150.1.1 source loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.150.1.1, timeout is 2 seconds:
Packet sent with a source address of 172.100.1.1
.....
Success rate is 0 percent (0/5)
CPE-5#
```

Nota. Ejemplo de prueba de conectividad CPE-5. Elaboración propia, realizado con EVE-NG.

La razón principal que la prueba de conectividad entre los CPES falla es por el mecanismo de prevención de loops, esta configuración ya se encuentra de forma implícita dentro de BGP, esta regla indica que un enrutador no puede recibir prefijos de parte de un vecino que anuncia el mismo AS del cual pertenece, por ejemplo desde la perspectiva de CPE-5 los anuncios de BGP le indican que recibe prefijos de un AS-60 y es del AS del cual forma parte, entonces el

enrutador verifica la procedencia de los prefijos y confirma que los prefijos que le envía PE-1 son del mismo AS, determinando que existe un loop dentro de la red y procede a descartar los paquetes.

Para solucionar este problema existen dos soluciones a nivel de configuración uno de ellos de parte del ISP y el otro es sobre la red del cliente, la solución sobre el ISP es mediante el cambio del AS de origen, el enrutador de borde PE ejecuta la instrucción de sobrescribir el número de sistema autónomo utilizado por un equipo CPE por el AS local, esto ocurre sobre una sesión BGP de tipo externa que contiene una VRF.

El comando para esta solución se aplica dentro del proceso de vecindad de los CPEs, el cual remplazará el AS del cliente por el AS del ISP sobre los anuncios de BGP.

Figura 171.

Anulación de AS sobre PE-1

```
!
router bgp 100
 !
 vrf INTERCAD
  !
  neighbor 10.0.0.50
    as-override
   !
  !
```

Nota. Ejemplo de anulación de AS/PE-1. Elaboración propia, realizado con EVE-NG.

El comando que permite este cambio de AS es, as-override el cual solo se encuentra disponible dentro de la familia de direccionamiento VPNv4.

Figura 172.

Anulación de AS sobre PE-3

```
!
router bgp 100
 address-family ipv4 vrf INTERCAD
  neighbor 10.0.0.54 as-override
 exit-address-family
 !
```

Nota. Ejemplo de anulación de AS/PE-3. Elaboración propia, realizado con EVE-NG.

Si se vuelve a consultar la tabla de enrutamiento sobre CPE-5 y CPE-6 se observa que hay una entrada hacia los prefijos remotos, ahora los atributos con los que acompañan los prefijos es el AS-PATH que contiene dos saltos de AS, de esta forma se verifica que se cambia el AS-60 del cual pertenece CPE-5 para sustituirlo por el AS del ISP, logrando redistribuir los prefijos de los clientes a todos los enrutadores involucrados sobre la misma VRF.

Figura 173.

Atributo de AS-PATH en CPE-5

```
CPE-5#show bgp ipv4 unicast
BGP table version is 9, local router ID is 60.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  172.100.1.0/24   0.0.0.0                  0         32768 i
 *>  172.150.1.0/24   10.0.0.49                              0 100 100 i
 *>  172.200.1.0/24   0.0.0.0                  0         32768 i
 *>  172.250.1.0/24   10.0.0.49                              0 100 100 i
CPE-5#
```

Nota. Ejemplo de atributo AS-PAHT/CPE-5. Elaboración propia, realizado con EVE-NG.

Figura 174.

Enrutamiento en CPE-5

```
CPE-5#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.48/30 is directly connected, GigabitEthernet0/0
L        10.0.0.50/32 is directly connected, GigabitEthernet0/0
      172.100.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.100.1.0/24 is directly connected, Loopback0
L        172.100.1.1/32 is directly connected, Loopback0
      172.150.0.0/24 is subnetted, 1 subnets
B        172.150.1.0 [20/0] via 10.0.0.49, 00:08:33
      172.200.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.200.1.0/24 is directly connected, Loopback1
L        172.200.1.1/32 is directly connected, Loopback1
      172.250.0.0/24 is subnetted, 1 subnets
B        172.250.1.0 [20/0] via 10.0.0.49, 00:08:33
CPE-5#
```

Nota. Ejemplo de enrutamiento CPE-5. Elaboración propia, realizado con EVE-NG.

Figura 175.

Enrutamiento en CPE-6

```
CPE-6#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.52/30 is directly connected, GigabitEthernet0/0
L        10.0.0.54/32 is directly connected, GigabitEthernet0/0
      172.100.0.0/24 is subnetted, 1 subnets
B        172.100.1.0 [20/0] via 10.0.0.53, 00:07:49
      172.150.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.150.1.0/24 is directly connected, Loopback0
L        172.150.1.1/32 is directly connected, Loopback0
      172.200.0.0/24 is subnetted, 1 subnets
B        172.200.1.0 [20/0] via 10.0.0.53, 00:07:49
      172.250.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.250.1.0/24 is directly connected, Loopback1
L        172.250.1.1/32 is directly connected, Loopback1
CPE-6#
```

Nota. Ejemplo de enrutamiento CPE-6. Elaboración propia, realizado con EVE-NG.

Ya solucionado el problema del envío de prefijos se vuelve a realizar una prueba de conectividad CPE-5 a CPE-6 y se verifica finalmente que la comunicación se completa de forma exitosa.

Figura 176.

Prueba de conectividad de CPE-5

```
CPE-5#ping 172.150.1.1 source lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.150.1.1, timeout is 2 seconds:
Packet sent with a source address of 172.100.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 15/22/48 ms
CPE-5#
```

Nota. Ejemplo conectividad CPE-5. Elaboración propia, realizado con EVE-NG.

Figura 177.

Traza hacia el CPE-6

```
CPE-5#traceroute 172.150.1.1 source lo0
Type escape sequence to abort.
Tracing the route to 172.150.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.49 8 msec 4 msec 5 msec
  2 10.0.0.1 [MPLS: Labels 24001/33 Exp 0] 26 msec 14 msec 15 msec
  3 10.0.0.53 [MPLS: Label 33 Exp 0] 16 msec 22 msec 16 msec
  4 10.0.0.54 18 msec 18 msec *
CPE-5#
```

Nota. Ejemplo de traza – CPE-6. Elaboración propia, realizado con EVE-NG.

6.7. Configuración de VRF sobre una interconexión MPLS Inter-AS

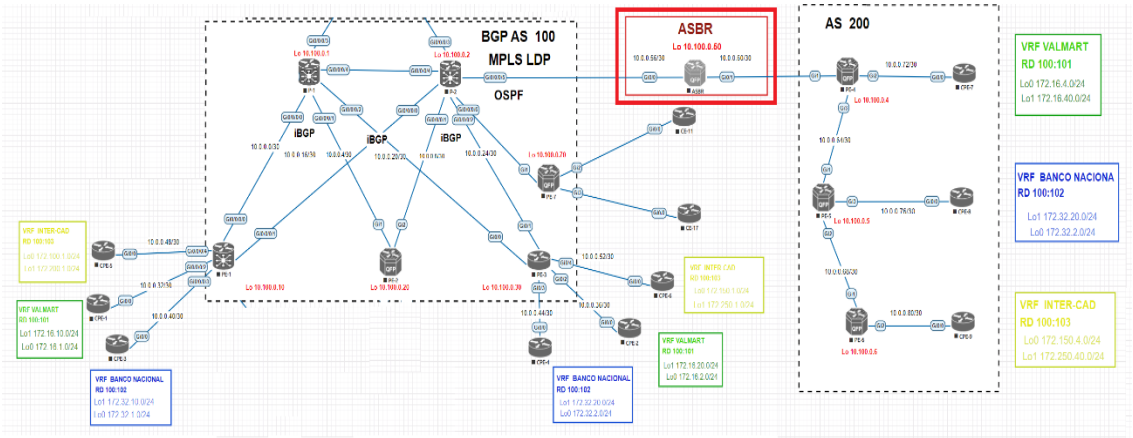
En el siguiente escenario, los clientes como Valmart, Intercad y Banco Nacional, necesitan establecer una conexión con nuevos sitios remotos en una región donde el ISP AS-100 no cuenta con la cobertura en ese país por lo que no

pueden establecer una conexión directa hacia los nuevos CPEs, por ende, los clientes contratan a otro proveedor que les ofrezca la comunicación de forma local dentro del nuevo AS-200. Los clientes conectados con el AS-100 desean mantener una comunicación activa con los CPEs del AS-200, donde es necesario ampliar los límites de enrutamiento de un AS a otro, entonces ambos ISP 100 y 200 deben de coordinarse para que la L3VPN se les proporcione a los clientes finales, dicha solución se llama Inter-AS.

El modelo Inter-AS es un modelo peer to peer que permite la extensión de la VPN a través de múltiples proveedores, esta solución permite a los proveedores ISP emparejarse entre sí y ofrecer conectividad VPN de extremo a extremo en ubicaciones geográficas extendidas para todo cliente que se encuentra fuera del alcance de un solo proveedor este modelo se implementará dentro de la siguiente topología.

Figura 178.

Diseño completo sobre la red L3VPN

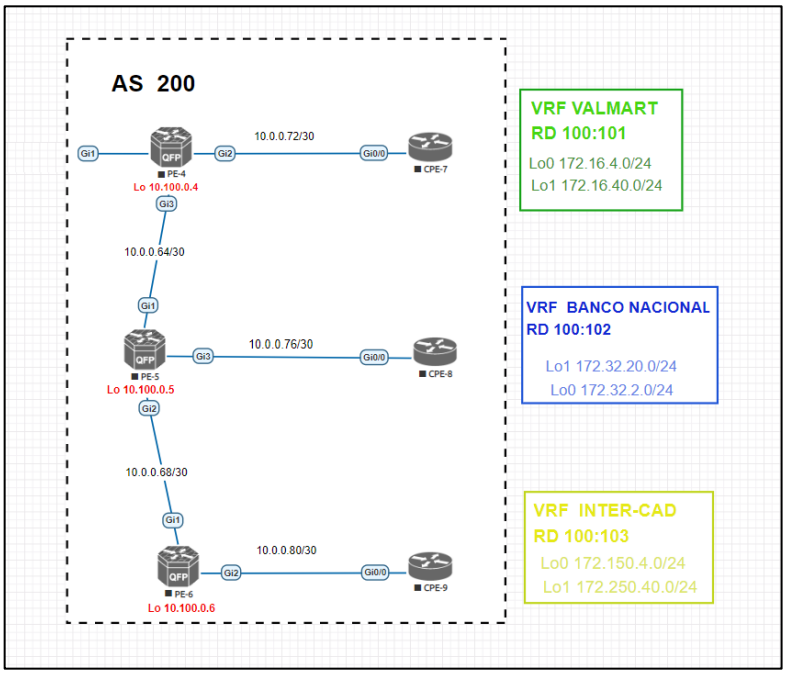


Nota. Ejemplo de diseño red L3VPN. Elaboración propia, realizado con EVE-NG.

Para crear el AS 200 se debe replicar las configuraciones de OSPF, BGP y MPLS hasta levantar todas las vecindades entre los protocolos para su correcto funcionamiento, se estarán utilizando tres enrutadores PE-4, PE-5 Y PE-6 siendo el PE-5 el equipo principal donde se estará configurando el reflector de ruta; ya mencionado el rol de cada equipo, se procede con su configuración de los equipos PEs del AS-200 como se muestra en el siguiente diseño.

Figura 179.

Topología de red del AS 200



Nota. Ejemplo de topología de red AS 200. Elaboración propia, realizado con EVE-NG.

Las configuraciones de cada equipo fueron completadas y se comparte las vecindades entre los PEs del AS 200 donde se comprueba su correcta configuración sobre los protocolos de OSPF, MPLS y BGP.

Figura 180.

Vecinos en OSPF

```
PE-5#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.100.0.6        1   FULL/DR         00:00:31    10.0.0.70       GigabitEthernet2
10.100.0.4        1   FULL/DR         00:00:32    10.0.0.65       GigabitEthernet1
PE-5#
```

Nota. Ejemplo de vecinos OSPF. Elaboración propia, realizado con EVE-NG.

Figura 181.

Vecinos en MPLS

```
PE-5#show mpls ldp neig | inc Peer
    Peer LDP Ident: 10.100.0.4:0; Local LDP Ident 10.100.0.5:0
    Peer LDP Ident: 10.100.0.6:0; Local LDP Ident 10.100.0.5:0
PE-5#
```

Nota. Ejemplo de vecinos MPLS. Elaboración propia, realizado con EVE-NG.

Figura 182.

Vecinos en BGP en PE-5

```
PE-5#show bgp ipv4 unicast summary
BGP router identifier 10.100.0.5, local AS number 200
BGP table version is 5, main routing table version 5
4 network entries using 992 bytes of memory
4 path entries using 544 bytes of memory
3/3 BGP path/bestpath attribute entries using 864 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
4 BGP extended community entries using 148 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2596 total bytes of memory
BGP activity 12/0 prefixes, 12/0 paths, scan interval 60 secs
4 networks peaked at 23:12:19 Oct 12 2022 UTC (00:01:40.202 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.100.0.4      4          200      12      15        5    0    0 00:02:34        2
10.100.0.6      4          200       9      16        5    0    0 00:02:38        1

PE-5#
```

Nota. Ejemplo de vecinos BGP. Elaboración propia, realizado con EVE-NG.

La configuración de los CPEs para cada PE se configura con las VRF de las secciones anteriores con el protocolo de enrutamiento correspondiente, para confirmar el correcto funcionamiento se realiza pruebas de conectividad de los enrutadores PE hacia sus CPEs.

Figura 183.

Prueba de conectividad hacia CPE-

```
PE-4#PING VRF VALMART 172.16.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.74, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/3/6 ms
PE-4#
```

Nota. Ejemplo de prueba de conectividad – CPE. Elaboración propia, realizado con EVE-NG.

Figura 184.

Prueba de conectividad hacia CPE-8

```
PE-5#ping vrf BANCONACIONAL 172.32.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.32.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/44/95 ms
PE-5#
```

Nota. Ejemplo de prueba de conectividad – CPE-8. Elaboración propia, realizado con EVE-NG.

Figura 185.

Prueba de conectividad hacia CPE-9

```
PE-6#ping vrf  INTERCAD 172.150.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.150.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/15/28 ms
```

Nota. Ejemplo de prueba de conectividad – CPE-9. Elaboración propia, realizado con EVE-NG.

Como primer paso para implementar el modelo Inter-AS es asignar un enrutador dedicado para este requerimiento, este equipo es llamado por su sigla ASBR que significa autonomous system boundary *router* que traducido es enrutador de límites de sistemas autónomos. En la topología principal se ubica el equipo ASBR entre P-2 y PE-4, por lo tanto, la conexión de P-2 hacia el ASBR. Se leventan las sesiones correspondientes de los protocolos OSPF, BGP y MPLS para indicar que forman parte del primer proveedor, la conexión hacia PE-4 que corresponde solamente hacia el AS-200 solo se habilita una sesión BGP IPv4, estableciendo una vecindad de transporte con el vecino, se observa el comando send label es asignado sobre el proceso de BGP, esto es necesario debido que todos los prefijos que son anunciados hacia el otro AS de forma global se le debe de agregar una etiqueta LDP, para ser agregado al dominio MPLS correspondiente.

A continuación, se comparten las primeras configuraciones sobre el equipo ASBR

Figura 186.

Direccionamiento IP en ASBR

```
!
interface Loopback0
 ip address 10.100.0.50 255.255.255.255
!
interface GigabitEthernet0/0
 ip address 10.0.0.58 255.255.255.252
!
interface GigabitEthernet0/1
 ip address 10.0.0.61 255.255.255.252
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Figura 187.

Proceso BGP en ASBR

```
!
router bgp 100
 bgp router-id 10.100.0.50
 no bgp default ipv4-unicast
 neighbor 10.0.0.62 remote-as 200
 !
 address-family ipv4
  network 10.100.0.50 mask 255.255.255.255
  neighbor 10.0.0.62 activate
  neighbor 10.0.0.62 send-community both
  neighbor 10.0.0.62 send-label
 exit-address-family
 !
```

Nota. Ejemplo de proceso BGP-ASBR. Elaboración propia, realizado con EVE-NG.

Figura 188.

Proceso BGP en PE-4

```
!
router bgp 200
 bgp router-id 10.100.0.4
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.0.0.61 remote-as 100
 !
 address-family ipv4
  network 10.100.0.4 mask 255.255.255.255
  neighbor 10.0.0.61 activate
  neighbor 10.0.0.61 send-community both
  neighbor 10.0.0.61 send-label
 exit-address-family
 !
```

Nota. Ejemplo de proceso BGP – PE-4. Elaboración propia, realizado con EVE-NG.

Se verifica la correcta configuración de vecindad mediante los siguientes comandos de comprobación.

Figura 189.

Verificación de sesión BGP en ASBR hacia PE-4

```
ASBR#show bgp ipv4 unicast  summary
BGP router identifier 10.100.0.50, local AS number 100
BGP table version is 13, main routing table version 13
8 network entries using 1152 bytes of memory
8 path entries using 640 bytes of memory
4/4 BGP path/bestpath attribute entries using 608 bytes of memory
1 BGP rrinfo entries using 24 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2448 total bytes of memory
BGP activity 8/0 prefixes, 8/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.62       4          200      18      20       13    0    0 00:13:04        3
ASBR#
```

Nota. Ejemplo de verificación BGP/ASBER – PE-4. Elaboración propia, realizado con EVE-NG.

Figura 190.

Verificación de sesión BGP en PE-4 hacia ASBR

```
PE-4#show bgp ipv4 unicast summary
BGP router identifier 10.100.0.4, local AS number 200
BGP table version is 14, main routing table version 14
8 network entries using 1984 bytes of memory
8 path entries using 1088 bytes of memory
4/4 BGP path/bestpath attribute entries using 1152 bytes of memory
1 BGP rrinfo entries using 40 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
2 BGP extended community entries using 64 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4352 total bytes of memory
BGP activity 16/0 prefixes, 20/4 paths, scan interval 60 secs
8 networks peaked at 23:19:30 Oct 12 2022 UTC (00:30:27.642 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.61       4          100      51      52       14    0    0 00:42:03        5

PE-4#
```

Nota. Ejemplo de verificación BGP/PE-4 – ASBR. Elaboración propia, realizado con EVE-NG.

Para lograr tener comunicación entre los enrutadores PE de ambos sistemas autónomos se debe anunciar por BGP las interfaces de loopback de todos los equipos PE para que puedan anunciarse entre los AS y de esta forma ubicar a cada PE dentro de la topología de red.

Figura 191.

Anuncio de interfaz loopback en PE-5

```
!
router bgp 200
 !
 address-family ipv4
  network 10.100.0.5 mask 255.255.255.255
 !
```

Nota. Ejemplo de anuncio de interfaz. Elaboración propia, realizado con EVE-NG.

Figura 192.

Anuncio de interfaz loopback en PE-4

```
!
router bgp 200
 !
 address-family ipv4
  network 10.100.0.4 mask 255.255.255.255
 !
```

Nota. Ejemplo de anuncio de interfaz. Elaboración propia, realizado con EVE-NG.

Figura 193.

Anuncio de interfaz loopback en PE-6

```
!
router bgp 200
 !
 address-family ipv4
 network 10.100.0.6 mask 255.255.255.255
 !
```

Nota. Ejemplo de anuncio de interfaz. Elaboración propia, realizado con EVE-NG.

Figura 194.

Anuncio de interfaz loopback en PE-3

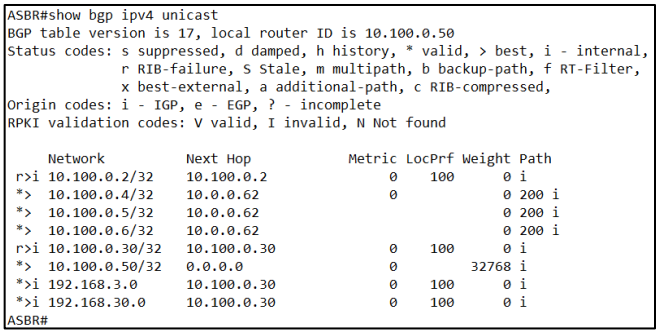
```
!
router bgp 100
 !
 address-family ipv4
  network 10.100.0.30 mask 255.255.255.255
 !
```

Nota. Ejemplo de anuncio de interfaz. Elaboración propia, realizado con EVE-NG.

Se verifica que los prefijos de las interfaces loopback se conozcan hacia el otro AS anunciadas por BGP y el enrutador ASBR deberá inyectar dentro de las tablas de enrutamiento los prefijos aprendidos y mediante una prueba de conectividad, se comprueba que se puedan alcanzar los PE de un AS a otro, como se observa a continuación.

Figura 195.

Prefijos BGP en ASBR

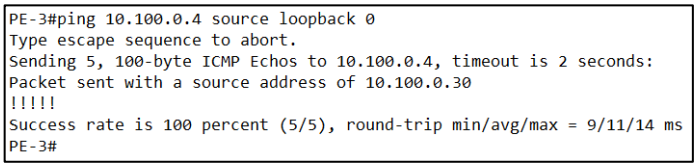


```
ASBR#show bgp ipv4 unicast
BGP table version is 17, local router ID is 10.100.0.50
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 r>i 10.100.0.2/32    10.100.0.2               0    100      0 i
 *>  10.100.0.4/32    10.0.0.62                0             0 200 i
 *>  10.100.0.5/32    10.0.0.62                              0 200 i
 *>  10.100.0.6/32    10.0.0.62                              0 200 i
 r>i 10.100.0.30/32   10.100.0.30              0    100      0 i
 *>  10.100.0.50/32   0.0.0.0                   0         32768 i
 *>i 192.168.3.0      10.100.0.30              0    100      0 i
 *>i 192.168.30.0     10.100.0.30              0    100      0 i
ASBR#
```

Nota. Ejemplo de prefijos. Elaboración propia, realizado con EVE-NG.

Figura 196.

Prueba de conectividad de PE-3 hacia PE-4

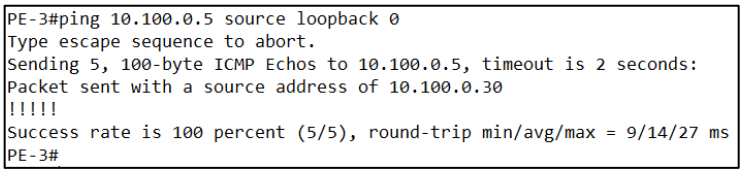


```
PE-3#ping 10.100.0.4 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.0.4, timeout is 2 seconds:
Packet sent with a source address of 10.100.0.30
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/11/14 ms
PE-3#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 197.

Prueba de conectividad de PE-3 hacia PE-5



```
PE-3#ping 10.100.0.5 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.0.5, timeout is 2 seconds:
Packet sent with a source address of 10.100.0.30
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/14/27 ms
PE-3#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 198.

Prueba de conectividad de PE-3 hacia PE-6

```
PE-3#ping 10.100.0.6 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.0.6, timeout is 2 seconds:
Packet sent with a source address of 10.100.0.30
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 11/15/27 ms
PE-3#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Continuando con el modelo Inter-AS los reflectores de rutas de cada ISP deberán formar adyacencia BGP de tipo externa sobre la familia de direccionamiento VPNv4 esto con el fin de brindar transporte a los prefijos que se encuentren en la VRF, por lo tanto, estas configuraciones se realizan dentro de P-2 Y PE-5.

Figura 199.

Configuración de vecindad VPNv4 hacia el AS 200

```
!
router bgp 100
 !
 neighbor 10.100.0.5
  remote-as 200
  ebgp-multihop 255
  update-source Loopback0
  address-family vpnv4 unicast
   next-hop-unchanged
  !
 !
```

Nota. Ejemplo de configuración de vecindad. Elaboración propia, realizado con EVE-NG.

El comando ebgp-multihop sirve para declarar que el vecino al que se desea levantar la vecindad, se encuentra a más de un salto de la red, el valor de 255 es el límite de saltos que puede dar el paquete antes de ser descartado, este valor se puede editar por el administrador y colocar el número real de saltos hacia el vecino, si no se coloca el número se saltos exactos, el enrutador coloca el 255 por defecto, ahora el comando next-hop-unchanged tiene la finalidad de conservar el atributo de siguiente salto cuando se anuncien los prefijos de un AS a otro.

Figura 200.

Configuración de vecindad VPNv4 hacia el AS 100

```
!
router bgp 200
 neighbor 10.100.0.2 remote-as 100
 neighbor 10.100.0.2 ebgp-multihop 255
 neighbor 10.100.0.2 update-source Loopback0
 !
 address-family vpnv4
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community both
  neighbor 10.100.0.2 next-hop-unchanged
 !
!
```

Nota. Ejemplo de configuración de vecindad. Elaboración propia, realizado con EVE-NG.

Se verifica la correcta configuración de vecindad mediante los siguientes comandos de comprobación.

Figura 201.

Verificación de vecindad de P-2 hacia PE-5

```
RP/0/0/CPU0:P-2#show bgp vpnv4 unicast  summary
Thu Oct 13 01:11:31.972 UTC
BGP router identifier 10.100.0.2, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 55
BGP NSR Initial initsync version 5 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.


Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker              55         55         55         55          55           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
10.100.0.5        0   200     135     118       55    0    0 01:51:56          8

RP/0/0/CPU0:P-2#
```

Nota. Ejemplo de configuración de vecindad. Elaboración propia, realizado con EVE-NG.

Por último, para completar la comunicación de extremo a extremo todos los prefijos aprendidos por el peer del AS a través de BGP se debe de inyectar por el protocolo IGP para que puedan agregarse el proceso de etiquetado de la red MPLS, por lo tanto, se finaliza la configuración en el ASBR con la redistribución de BGP hacia OSPF.

Figura 202.

Redistribución BGP de OSPF sobre el ASBR

```
!
router ospf 1
 redistribute bgp 100 subnets
!
```

Nota. Ejemplo de redistribución. Elaboración propia, realizado con EVE-NG.

Figura 203.

Redistribución BGP de OSPF sobre el PE-4

```
!
router ospf 1
 redistribute bgp 200
!
```

Nota. Ejemplo de redistribución. Elaboración propia, realizado con EVE-NG.

En este momento ya se está listos para verificar la conectividad de los sitios remotos ubicados en el AS-200 hacia el AS-100, por lo que se adjuntan pruebas de conectividad de los sitios remotos y se observa que la comunicación se completa con éxito.

Figura 204.

Prueba de conectividad hacia CPE-2

```
CPE-7#ping  172.16.20.1 source Loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.4.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/19/24 ms
CPE-7#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 205.

Traza hacia CPE-2

```
CPE-7#traceroute 172.16.20.1 source Loopback0
Type escape sequence to abort.
Tracing the route to 172.16.20.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.73 5 msec 3 msec 4 msec
  2 10.0.0.61 [MPLS: Labels 20/27 Exp 0] 23 msec 22 msec 19 msec
  3 10.0.0.57 [MPLS: Labels 24000/27 Exp 0] 28 msec 13 msec 16 msec
  4 10.0.0.37 [MPLS: Label 27 Exp 0] 22 msec 18 msec 16 msec
  5 10.0.0.38 21 msec 22 msec *
CPE-7#
```

Nota. Ejemplo de traza. Elaboración propia, realizado con EVE-NG.

Figura 206.

Prueba de conectividad hacia CPE-4

```
CPE-8#ping  172.32.2.1 source lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.32.2.1, timeout is 2 seconds:
Packet sent with a source address of 172.32.4.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/20/27 ms
CPE-8#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 207.

Traza hacia CPE-4

```
CPE-8#traceroute 172.32.2.1 source lo0
Type escape sequence to abort.
Tracing the route to 172.32.2.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.77 5 msec 6 msec 3 msec
  2 10.0.0.65 [MPLS: Labels 26/30 Exp 0] 29 msec 257 msec 96 msec
  3 10.0.0.61 [MPLS: Labels 20/30 Exp 0] 17 msec 18 msec 26 msec
  4 10.0.0.57 [MPLS: Labels 24000/30 Exp 0] 16 msec 21 msec 19 msec
  5 10.0.0.45 [MPLS: Label 30 Exp 0] 18 msec 18 msec 19 msec
  6 10.0.0.46 20 msec 22 msec *
CPE-8#
```

Nota. Ejemplo de traza. Elaboración propia, realizado con EVE-NG.

Figura 208.

Prueba de conectividad hacia CPE-6

```
CPE-9#ping 172.250.1.1 source Loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.250.1.1, timeout is 2 seconds:
Packet sent with a source address of 172.150.4.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/21/23 ms
CPE-9#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 209.

Traza hacia CPE-6

```
CPE-9#traceroute  172.250.1.1 source Loopback0
Type escape sequence to abort.
Tracing the route to 172.250.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.81 6 msec 3 msec 3 msec
  2 10.0.0.69 [MPLS: Labels 24/32 Exp 0] 22 msec 656 msec 36 msec
  3 10.0.0.65 [MPLS: Labels 26/32 Exp 0] 14 msec 171 msec 383 msec
  4 10.0.0.61 [MPLS: Labels 20/32 Exp 0] 22 msec 30 msec 31 msec
  5 10.0.0.57 [MPLS: Labels 24000/32 Exp 0] 19 msec 14 msec 17 msec
  6 10.0.0.53 [MPLS: Label 32 Exp 0] 18 msec 21 msec 17 msec
  7 10.0.0.54 19 msec 22 msec *
CPE-9#
```

Nota. Ejemplo de traza. Elaboración propia, realizado con EVE-NG.

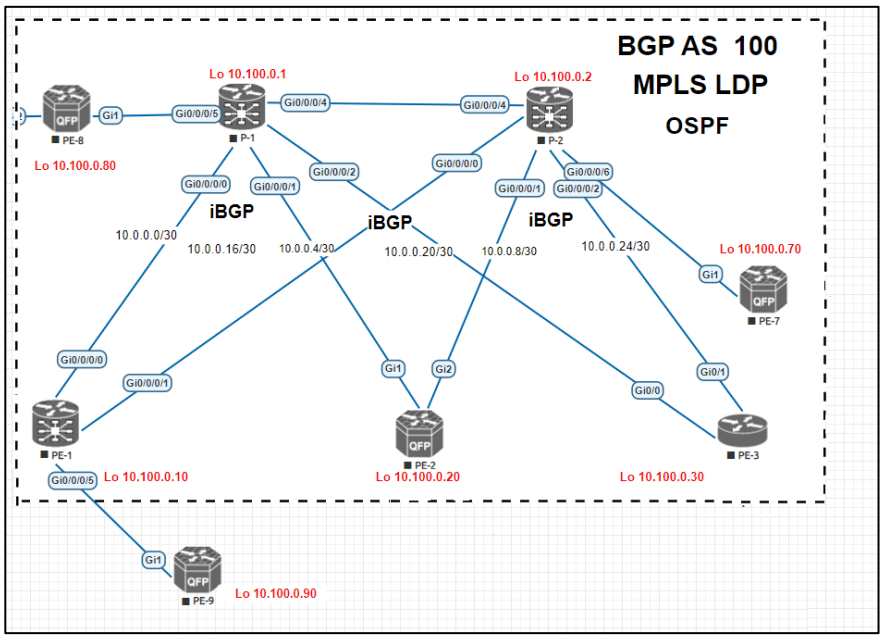
7. SIMULACIÓN DE SERVICIOS VPN DE CAPA 2 SOBRE MPLS (L2VPN)

En este nuevo capítulo se seguirá aprovechando la misma topología de red utilizada en el capítulo anterior, donde se presentó los servicios L3VPN para interconectar redes corporativas distribuidas a lo largo de una región geográfica, en nuestra topología se introducen cambios a nivel de equipos que permita ofrecer servicios de capa 2 sin involucrar procesos de enrutamiento bajo los requerimientos de los clientes que contraten este servicio.

Antes de comenzar con las configuraciones de los servicios L2VPN, se agregan tres equipos PE al diseño original los cuales son PE-7, PE-8 y PE-9 estos serán los encargados de brindar el transporte a los servicios de los clientes, por lo tanto, estos equipos PE se agregan al dominio MPLS del AS-100 para aprovechar los mecanismos de etiquetado de la red previamente configurada.

Figura 210.

Topología de red L2VPN



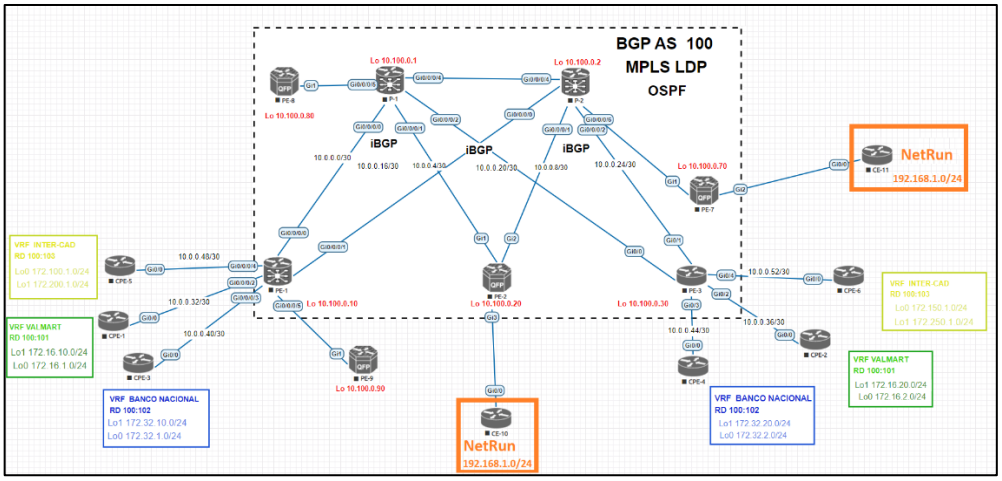
Nota. Ejemplo de topología. Elaboración propia, realizado con EVE-NG.

7.1. Configuración de VPWS sobre MPLS

En este escenario se tiene al siguiente cliente Netrun quien solicita extender su red LAN con el direccionamiento 192.168.1.0/24 hacia otra zona a varios kilómetros de sus instalaciones, por lo tanto requiere un servicio de VPWS para conectar dos sucursales mediante la capa 2 sin intervenir en procesos de enrutamiento, en la siguiente topología se observan los equipos CE que viene el inglés *Customer Equipment* que traducido quiere decir enrutadores administrados por el propio cliente y como ISP solo se le ofrece el transporte hasta conectar ambos puntos remotos sobre la capa 2.

Figura 211.

Diseño de red del cliente Netrun



Nota. Ejemplo de diseño de red Netrun. Elaboración propia, realizado con EVE-NG.

Antes de iniciar las configuraciones del túnel es necesario que los equipos CE-10 y CE-11 ubicados en las instalaciones del cliente cuenten con las configuraciones mínimas para establecer la comunicación entre los extremos, en este escenario no se involucran protocolos de enrutamiento para distribuir la información solo se agrega el direccionamiento IP dentro de la configuración principal como se muestra a continuación.

Figura 212.

Direccionamiento IP de CE-10

```
interface GigabitEthernet0/0
ip address 192.168.1.2 255.255.255.0
no shutdown
```

Nota. Ejemplo de direccionamiento. Elaboración propia, realizado con EVE-NG.

Figura 213.

Direccionamiento IP de CE-11

```
interface GigabitEthernet0/0
ip address 192.168.1.3 255.255.255.0
no shutdown
```

Nota. Ejemplo de direccionamiento. Elaboración propia, realizado con EVE-NG.

En las configuraciones anteriores ambos equipos utilizan el mismo segmento de red para comunicarse entre sí, en casos donde se involucra procesos de enrutamiento esto no es posible debido que cada segmento de red es diferente en cada sitio con el objetivo de identificar cada sucursal.

Para establecer el túnel de capa 2 es necesario ya contar con una red MPLS ya establecida con los equipos PE involucrados en el transporte, en este caso la topología de la figura 211 ya se encuentra en el dominio MPLS.

Se inicia la configuración sobre PE-2 y PE-7 definiendo el pseudowire con un nombre de clase el cual se llamará WIRE-1 y el tipo de encapsulación del túnel, en este caso será del tipo MPLS.

Figura 214.

Pseudowire en PE-2

```
!
pseudowire-class WIRE-1
 encapsulation mpls
!
```

Nota. Ejemplo de Pseudowire. Elaboración propia, realizado con EVE-NG.

Figura 215.

Pseudowire en PE-7

```
!
pseudowire-class WIRE-1
 encapsulation mpls
!
```

Nota. Ejemplo de Pseudowire. Elaboración propia, realizado con EVE-NG.

Ahora se asigna la interfaz física que conecta cada enrutador CE hacia su PE correspondiente, luego mediante la configuración de PW dentro de la interfaz se establece una conexión punto a punto hasta el otro PE que se define mediante su IP de loopback incluyendo el identificador del túnel que tendrá el valor del número 10 y su respectiva clase mencionada anteriormente.

Figura 216.

Pseudowire en PE-7 sobre interfaz de acceso

```
!
interface GigabitEthernet2
 xconnect 10.100.0.20 10 encapsulation mpls pw-class WIRE-1
!
```

Nota. Ejemplo de Pseudowire. Elaboración propia, realizado con EVE-NG.

La sintaxis de este comando es la misma para todos los demás equipos PE, únicamente cambia la dirección de loopback el cual se establece el túnel punto a punto.

Figura 217.

Pseudowire en PE-2 sobre interfaz de acceso

```
!
interface GigabitEthernet3
 xconnect 10.100.0.40 10 encapsulation mpls pw-class WIRE-1
!
```

Nota. Ejemplo de Pseudowire. Elaboración propia, realizado con EVE-NG.

Ya finalizadas las configuraciones sobre el túnel se comprueban por medio de los comandos de verificación para ver su correcto funcionamiento.

Figura 218.

Verificación del túnel xconnect en PE-2

```
PE-2#show xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
UP pri   ac Gi3:9(Ethernet)              UP mpls 10.100.0.40:10               UP

PE-2#
```

Nota. Ejemplo de verificación de túnel. Elaboración propia, realizado con EVE-NG.

El campo se XC muestra el estado actual del PW, las cuales pueden estar en los siguientes estados. UP: el circuito PW se encuentra activo, tanto el segmento 1 como el segmento 2 deben de estar activos para que el circuito permanezca activo.

- DN: el circuito PW se encuentra inactivo, tanto el segmento 1 como el segmento 2 se encuentran caídos.

- IA: se utiliza cuando el PW está configurado de forma redundante.

- Los segmentos 1 y 2 brindan la información sobre el tipo de interfaz, tipo de encapsulación y dirección IP que utiliza el PW, las cuales también pueden estar en los siguientes estados.

- UP: el segmento se encuentra activo.

- DN: el segmento se encuentra caído.

- AD: el segmento se encuentra administrativamente apagado.

Figura 219.

Verificación del túnel xconnect en PE-7

```
PE-7#show xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
UP pri   ac Gi2:8(Ethernet)              UP mpls 10.100.0.20:10               UP

PE-7#
```

Nota. Ejemplo de verificación de túnel. Elaboración propia, realizado con EVE-NG.

Ya llegado en este punto se verifica con una prueba de conectividad de CE-10 hacia CE-11 y viceversa para comprobar si existe comunicación entre los extremos, con la prueba de ping se observa que la comunicación entre los sitios remotos se completa de forma exitosa.

Figura 220.

Prueba de conectividad hacia CE-11

```
CE-10#ping  192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/4/9 ms
CE-10#
```

Nota. Ejemplo de verificación de túnel. Elaboración propia, realizado con EVE-NG.

Figura 221.

Prueba de conectividad hacia CE-10

```
CE-11#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 11/13/17 ms
CE-11#
```

Nota. Ejemplo de verificación de túnel. Elaboración propia, realizado con EVE-NG.

Figura 222.

Traza hacia CE-10

```
CE-11#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 11/13/17 ms
CE-11#
```

Nota. Ejemplo de traza. Elaboración propia, realizado con EVE-NG.

Se observa en la traza solo muestra un salto hasta su extremo esto es porque todo el transporte se realiza mediante la capa 2.

7.2. Configuración de VPLS LDP sobre MPLS

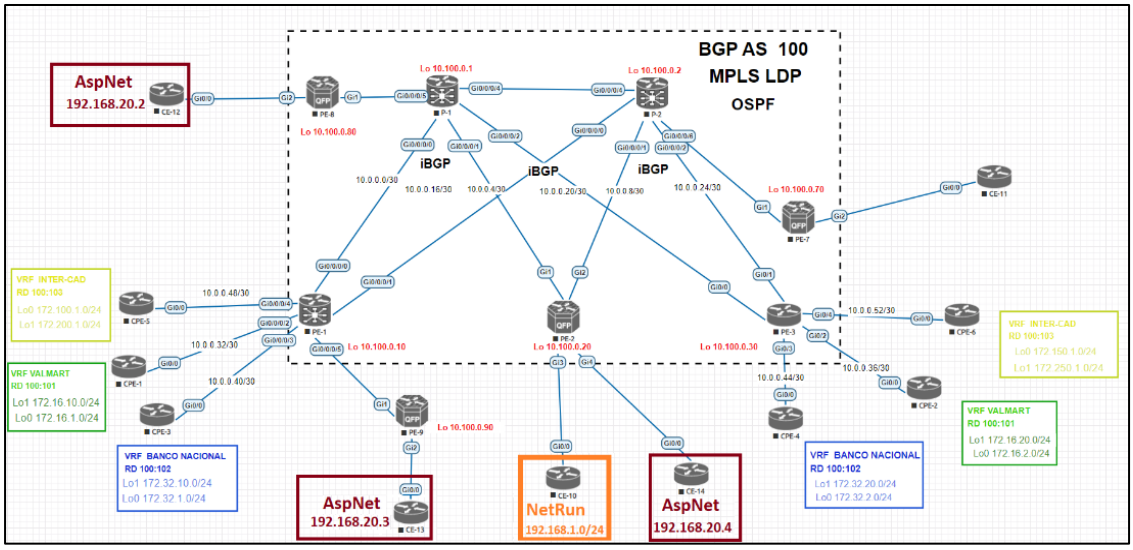
Los servicios L2VPN también ofrece soluciones de red punto a multipunto el cual tiene como objetivo interconectar varios sitios remotos sobre el mismo segmento LAN el cual aprovecha el aprendizaje de direcciones MAC utilizándolos para múltiples propósitos que requiera el cliente por estar en el mismo dominio de broadcast, Desde el punto de vista del cliente se simula un conmutador a gran escala para ofrecer comunicación sobre la capa 2 a múltiples sitios remotos.

Las redes configuradas con VPLS adopta el tipo de red hub and spoke por lo que es necesario asignar un equipo el cual tendrá el rol central para comunicarse con todos los sitios remotos, entendiendo este modelo se simplifica las configuraciones debido que el hub tendrá la configuración de VPLS mientras que los sitios remotos levantaran PW individuales hacia su Hub, de lo contrario se levantarían PW independientes para cada sitio lo que conlleva una mayor complejidad de configuración y de administración.

Antes de comenzar con las configuraciones de VPLS se añaden los siguientes equipos CE del cliente AspNet quien solicita mantener una comunicación activa con varios sitios remotos mediante la capa 2 como se muestra en el siguiente diseño, los equipos CE-12, CE-13 y CE-14 serán los equipos ubicados en los sitios del cliente, el equipo PE-8 será el equipo Hub mientras que los equipos PE-9 Y PE-2 como el spoke, a continuación, se comparte el direccionamiento sobre los equipos CE correspondientes.

Figura 223.

Diseño de red del cliente AspNet



Nota. Ejemplo de diseño de red AspNet. Elaboración propia, realizado con EVE-NG.

Figura 224.

Direccionamiento IP de CE-12

```
!
interface GigabitEthernet0/0.20
 encapsulation dot1Q 20
 ip address 192.168.20.2 255.255.255.0
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Figura 225.

Direccionamiento IP de CE-13

```
!
interface GigabitEthernet0/0
 encapsulation dot1Q 20
 ip address 192.168.20.3 255.255.255.0
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Figura 226.

Direccionamiento IP de CE-14

```
!
interface GigabitEthernet0/0
 encapsulation dot1Q 20
 ip address 192.168.20.4 255.255.255.0
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Se inicia la configuración sobre PE-8 el cual será el equipo que tendrá la única configuración de VPLS, en primer lugar, se define la VFI de forma manual que se nombra como LAN-1 con su respectivo ID de VPN con valor de 100, se establece el dominio VPLS utilizando la vlan 20 definida como bridge-domain el cual ayudará a identificar el tráfico del cliente, adicional se especifica a los vecinos que se estará estableciendo el túnel de capa 2 con la clase de PW y su respectivo tipo de encapsulación siendo esta MPLS y su VC ID de 20 para identificar los PW hacia los spokes.

Figura 227.

Configuración de VPLS en PE-8

```
!
l2 vfi LAN-1 manual
 vpn id 100
 bridge-domain 20
 neighbor 10.100.0.20 20 pw-class WIRE-2
 neighbor 10.100.0.90 20 pw-class WIRE-2
!
!
pseudowire-class WIRE-2
 encapsulation mpls
!
```

Nota. Ejemplo de configuración VPLS. Elaboración propia, realizado con EVE-NG.

Para la interfaz hacia el equipo CE se debe de configurar de modo que las tramas se entreguen de forma etiquetada utilizando la vlan 20, para este caso por ser un equipo de IOS XE y en entornos ISP de utiliza el bridge domain, de esta forma se encargan de que las tramas lleguen al CE con la etiqueta 20.

Figura 228.

Bridge-domain en interfaz de acceso

```
!
interface BDI20
 no shutdown
!
interface GigabitEthernet2
 service instance 20 ethernet
 encapsulation dot1q 20
 rewrite ingress tag pop 1 symmetric
 bridge-domain 20
 !
!
```

Nota. Ejemplo Bridge-domain. Elaboración propia, realizado con EVE-NG.

Se finalizó la configuración de VPLS en PE-8 ahora se encargan de la configuración sobre los enrutadores PE-2 y PE-9, en este caso solo es necesario que exista un PW hacia el PE-8 para levantar la sesión de capa 2 y de esta forma se completa toda la comunicación hacia todos los sitios remotos.

Figura 229.

Configuración de túnel en PE-9

```
!
interface GigabitEthernet3
 xconnect 10.100.0.80 20 encapsulation mpls
!
```

Nota. Ejemplo configuración de túnel. Elaboración propia, realizado con EVE-NG.

Figura 230.

Configuración de túnel en PE-2

```
!
interface GigabitEthernet4
 xconnect 10.100.0.80 20 encapsulation mpls
!
```

Nota. Ejemplo configuración de túnel. Elaboración propia, realizado con EVE-NG.

Ya finalizadas todas las configuraciones del servicio VPLS se recurre a las pruebas de comprobación para verificar que todos los comandos fueron ingresados de forma exitosa.

Figura 231.

Verificación del túnel en PE-8

```
PE-8#show xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down           AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby    RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
UP pri  vfi LAN-1                        UP mpls 10.100.0.90:20               UP
UP pri  vfi LAN-1                        UP mpls 10.100.0.20:20               UP
UP pri   bd 20                           UP  vfi LAN-1                        UP
PE-8#
```

Nota. Ejemplo configuración de túnel. Elaboración propia, realizado con EVE-NG.

Figura 232.

Verificación del bridge-domain en PE-8

```
PE-8#show bridge-domain 20
Bridge-domain 20 (4 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
Maximum address limit: 65536
    BDI20  (up)
    GigabitEthernet2 service instance 20
    vfi LAN-1 neighbor 10.100.0.20 20
    vfi LAN-1 neighbor 10.100.0.90 20
   AED MAC address    Policy  Tag       Age  Pseudoport
   0   5000.001B.0000 forward dynamic   213  GigabitEthernet2.EFP20
   -   001E.E5BB.FFBF to_bdi  static    0    BDI20
   0   5000.001E.0000 forward dynamic   298  LAN-1.404012
   0   5000.0022.0000 forward dynamic   298  LAN-1.404013
PE-8#
```

Nota. Ejemplo de verificación bridge-domain. Elaboración propia, realizado con EVE-NG.

Con este comando se observa las direcciones MAC que se aprenden de forma dinámica de todos los equipos CE con su tiempo de aprendizaje y por el PW que se conoce.

Figura 233.

Verificación de VPLS en PE-8

```
PE-8#show l2vpn service vfi all detail
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down     IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering     NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                  Prio  St  XC St
  ---------          -----       -------------                  ----  --  -----
VPLS name: LAN-1, State: UP
  pw100002                       LAN-1(VFI)                     0     UP  UP
  pw100004           core_pw     10.100.0.90:20(MPLS)           0     UP  UP
                                 Local VC label 17
                                 Remote VC label 16
                                 pw-class: WIRE-2
  pw100003           core_pw     10.100.0.20:20(MPLS)           0     UP  UP
                                 Local VC label 16
                                 Remote VC label 17
                                 pw-class: WIRE-2
```

Nota. Ejemplo de verificación. Elaboración propia, realizado con EVE-NG.

Se verifica que el túnel ya se encuentra correctamente establecido con sus vecinos, ahora si se realiza una prueba de conectividad hacia los sitios remotos se nota que la comunicación se completa de forma exitosa.

Figura 234.

Prueba de conectividad hacia CE-13 y CE-14

```
CE-12#ping 192.168.20.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/11/13 ms
CE-12#
```

```
CE-12#ping 192.168.20.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/11/13 ms
CE-12#
```

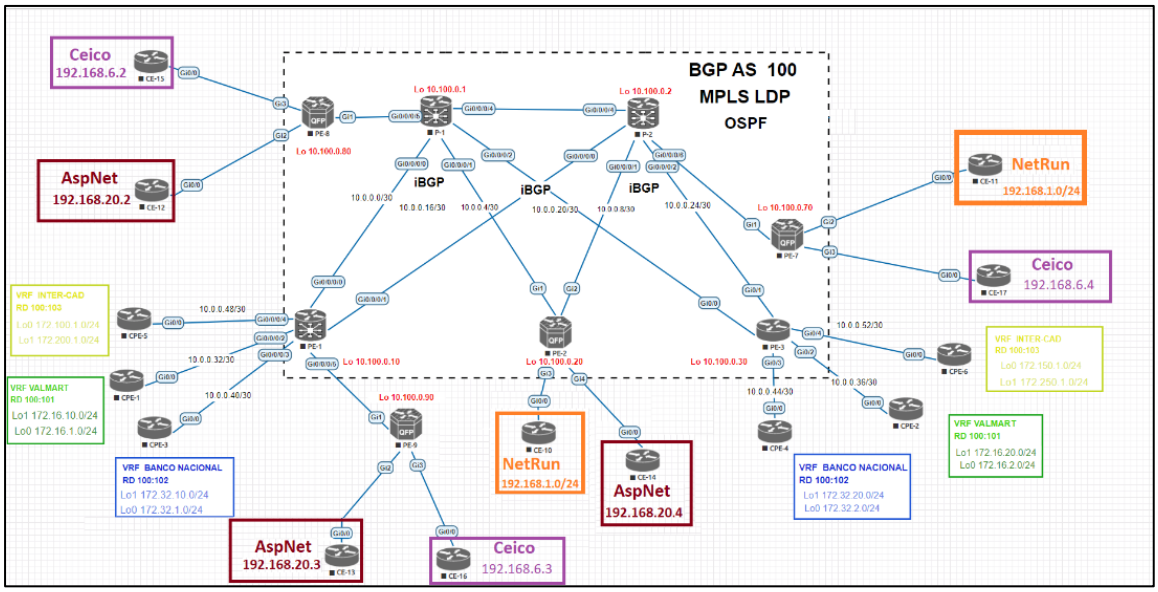
Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

7.3. Configuración de VPLS BGP sobre MPLS

Para este nuevo escenario se introducen los equipos CE-15,16 y 17 que pertenecen al cliente de Ceico los cuales se encuentran distribuidos a lo largo del territorio nacional el cual solicita compartir el mismo segmento LAN 192.168.6.0/24 entre todos los puntos remotos, debido a este requerimiento se le aplica de nuevo el mecanismo de envío de VPLS por ser tipo de enlace multipunto, pero en este caso se construirá mediante el autodescubrimiento de BGP habilitando una nueva familia de direccionamiento de l2vpn.

Figura 235.

Diseño de red del cliente Ceico



Nota. Ejemplo diseño de red Ceico. Elaboración propia, realizado con EVE-NG.

Antes de iniciar con las configuraciones se comparte el direccionamiento IP sobre los sitios remotos para dar continuidad al mecanismo de VPLS.

Figura 236.

Direccionamiento IP de CE-15

```
!
interface GigabitEthernet0/0.30
 encapsulation dot1Q 30
 ip address 192.168.6.2 255.255.255.0
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Figura 237.

Direccionamiento IP de CE-16

```
!
interface GigabitEthernet0/0.30
 encapsulation dot1Q 30
 ip address 192.168.6.3 255.255.255.0
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

Figura 238.

Direccionamiento IP de CE-17

```
!
interface GigabitEthernet0/0.30
 encapsulation dot1Q 30
 ip address 192.168.6.4 255.255.255.0
!
```

Nota. Ejemplo de direccionamiento IP. Elaboración propia, realizado con EVE-NG.

De momento los equipos CE ya se encuentran configuradas, pero aún falta establecer el túnel L2 que permita la transferencia de las tramas, para iniciar con dichas configuraciones se introduce una nueva familia de direccionamiento sobre BGP el cual es de tipo l2vpn, este tipo de vecindad se estará aplicando en los equipos PEs de donde dependan los equipos CE.

Se ingresa al proceso de BGP de todos los equipos y bajo la configuración de la familia l2vpn se activan los vecinos que formarán parte de esta nueva vecindad entre los PE-8, PE-7 y PE-9, el proceso de configuración se adjunta a continuación.

Figura 239.

Proceso BGP de l2vpn en PE-8

```
!
router bgp 100
 bgp router-id 10.100.0.80
 !
 address-family l2vpn vpls
  neighbor 10.100.0.40 activate
  neighbor 10.100.0.40 send-community both
  neighbor 10.100.0.90 activate
  neighbor 10.100.0.90 send-community both
 exit-address-family
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 240.

Proceso BGP de l2vpn en PE-9

```
!
router bgp 100
 bgp router-id 10.100.0.90
 !
 address-family l2vpn vpls
  neighbor 10.100.0.40 activate
  neighbor 10.100.0.40 send-community both
  neighbor 10.100.0.80 activate
  neighbor 10.100.0.80 send-community both
 exit-address-family
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Figura 241.

Proceso BGP de l2vpn en PE-7

```
!
router bgp 100
 bgp router-id 10.100.0.40
 !
 address-family l2vpn vpls
  neighbor 10.100.0.80 activate
  neighbor 10.100.0.80 send-community both
  neighbor 10.100.0.90 activate
  neighbor 10.100.0.90 send-community both
 exit-address-family
!
```

Nota. Ejemplo de proceso BGP. Elaboración propia, realizado con EVE-NG.

Se comprobaron las configuraciones anteriores con los comandos de verificación donde se puede observar la adyacencia de parte de la nueva familia de direccionamiento se haya completado de forma correcta.

Figura 242.

Verificación de vecindad Lvpn2 en PE-8

```
PE-8#show bgp l2vpn vpls all summary
BGP router identifier 10.100.0.80, local AS number 100
BGP table version is 3, main routing table version 3
2 network entries using 528 bytes of memory
2 path entries using 288 bytes of memory
2/2 BGP path/bestpath attribute entries using 576 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1512 total bytes of memory
BGP activity 6/0 prefixes, 6/0 paths, scan interval 60 secs
2 networks peaked at 02:12:36 Oct 7 2022 UTC (00:39:52.220 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.100.0.40     4          100      54      53        3    0    0 00:47:11        1
10.100.0.90     4          100      47      49        3    0    0 00:41:28        1
```

Nota. Ejemplo de verificación de vecindad. Elaboración propia, realizado con EVE-NG.

Al momento se configurar VPLS mediante BGP se obtienen las siguientes ventajas que se pueden aprovechar para aumentar el rendimiento de la red las cuales se mencionan a continuación.

- El descubrimiento de vecinos VPLS se detecta de forma automática, esto es una ventaja para la red ISP extensas que facilita la administración de la red.

- Se puede utilizar el reflector de ruta para evitar las mallas completas sobre la red.

- Utiliza las actualizaciones de BGP para intercambiar los mensajes relacionados a la VFI, incluso puede ser ejecutado en entornos empresariales donde no se cuente con una red MPLS configurada.

- Ya levantadas las sesiones sobre la familia de direccionamiento l2vpn se inician las configuraciones de VPLS sobre los equipos PE para ello se debe tener en cuenta las siguientes instrucciones de configuración.

- Se asignó una instancia VPLS VFI para que todos los PW se puedan definir como miembros de un contexto VFI esto se define en la primera línea con el comando l2vpn vfi context y se llamará con el nombre de VPLS-1.

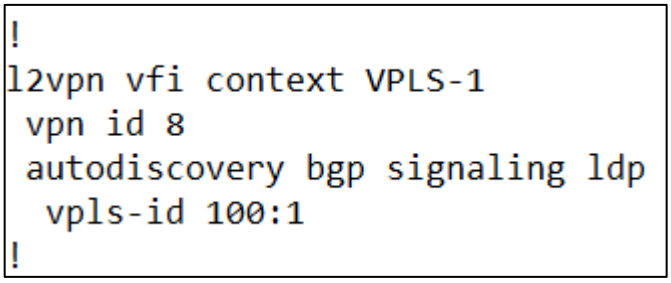
- El identificador de VPN ID se asigna para cada equipo con un valor diferente para identificar a cada PE dentro del dominio de VPLS para el PE-8 tendrá un valor de 8 respectivamente.

- Se especificó el modo de autodescubrimiento de vecinos mediante BGP.

- El VPLS id corresponde al dominio VPLS en el cual estarán todos los equipos PE este id será idéntico a todos los demás, utiliza el formato del número del sistema autónomo y el id que identifica al cliente.

Figura 243.

Configuración de VPLS BGP en PE-8



Nota. Ejemplo de configuración de VPLS BGP. Elaboración propia, realizado con EVE-NG.

Figura 244.

Configuración de VPLS BGP en PE-9

```
!
l2vpn vfi context VPLS-1
 vpn id 9
 autodiscovery bgp signaling ldp
  vpls-id 100:1
!
```

Nota. Ejemplo de configuración de VPLS BGP. Elaboración propia, realizado con EVE-NG.

Figura 245.

Configuración de VPLS BGP en PE-7

```
!
l2vpn vfi context VPLS-1
 vpn id 7
 autodiscovery bgp signaling ldp
  vpls-id 100:1
!
```

Nota. Ejemplo de configuración de VPLS BGP. Elaboración propia, realizado con EVE-NG.

Ya declarados los VFI sobre los equipos PE que participarán sobre el dominio VPLS se sigue con la configuración sobre las interfaces físicas que serán parte también del dominio, en este caso se creó un bridge-domain con el valor de la vlan 30 y se declara sobre la interfaz que conectara hacia los equipos CE para cada sitio remoto.

Figura 246.

Configuración de interfaz en PE- 8

```
!
interface GigabitEthernet3
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
```

Nota. Ejemplo de configuración de interfaz. Elaboración propia, realizado con EVE-NG.

Figura 247.

Configuración de interfaz en PE- 9

```
!
interface GigabitEthernet4
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
```

Nota. Ejemplo de configuración de interfaz. Elaboración propia, realizado con EVE-NG.

Figura 248.

Configuración de interfaz en PE- 7

```
!
interface GigabitEthernet4
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
```

Nota. Ejemplo de configuración de interfaz. Elaboración propia, realizado con EVE-NG.

Si se observan las configuraciones ya se han declarado los VFI y las interfaces hacia los CE que serán parte del dominio, pero aun hace falta la instrucción que relacione la VFI con las interfaces participantes, en este caso se ingresa al modo de configuración del bridge domain y se le indica que interfaces y VFI serán parte del dominio.

Figura 249.

Bridge-domain en PE-8

```
!
bridge-domain 30
 member GigabitEthernet3 service-instance 30
 member vfi VPLS-1
!
```

Nota. Ejemplo de bridge-domain. Elaboración propia, realizado con EVE-NG.

Figura 250.

Bridge-domain en PE-7

```
!
bridge-domain 30
 member GigabitEthernet4 service-instance 30
 member vfi VPLS-1
!
```

Nota. Ejemplo de bridge-domain. Elaboración propia, realizado con EVE-NG.

Figura 251.

Bridge-domain en PE-9

```
!
bridge-domain 30
 member GigabitEthernet4 service-instance 30
 member vfi VPLS-1
!
```

Nota. Ejemplo de bridge-domain. Elaboración propia, realizado con EVE-NG.

En este momento se han finalizado todas las configuraciones del servicio L2VPN, por lo tanto, para verificar que las configuraciones fueron realizadas de forma correcta, se comprueban con el siguiente comando entre los equipos PE.

Figura 252.

Verificación del bridge-domain en PE-8

```
PE-8#show bridge-domain 30
Bridge-domain 30 (3 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
Maximum address limit: 65536
    GigabitEthernet3 service instance 30
    vfi VPLS-1 neighbor 10.100.0.90 8
    vfi VPLS-1 neighbor 10.100.0.40 8
   AED MAC address    Policy  Tag       Age  Pseudoport
   0   5000.0021.0000 forward dynamic   146  GigabitEthernet3.EFP30
   0   5000.001F.0000 forward dynamic   145  VPLS-1.404015
   0   5000.0020.0000 forward dynamic   144  VPLS-1.404016
PE-8#
```

Nota. Ejemplo de verificación de bridge-domain. Elaboración propia, realizado con EVE-NG.

Figura 253.

Verificación de VPLS BGP en PE-8

```
PE-8#show l2vpn service vfi all detail
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down     IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering     NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                  Prio  St  XC St
  ---------          -----       -------------                  ----  --  -----
VPLS name: VPLS-1, State: UP
  pw100001                       VPLS-1(VFI)                    0     UP  UP
  pw100006           core_pw     10.100.0.40:8(MPLS)            0     UP  UP
                                 Local VC label 34
                                 Remote VC label 34

  pw100005           core_pw     10.100.0.90:8(MPLS)            0     UP  UP
                                 Local VC label 33
                                 Remote VC label 34

PE-8#
```

Nota. Ejemplo de verificación VPLS BGP. Elaboración propia, realizado con EVE-NG.

Si realizamos una prueba de conectividad hacia los sitios remotos desde CE-15 se observa que la comunicación se completa, finalizando las configuraciones de VPLS BGP.

Figura 254.

Prueba de conectividad hacia CE-16

```
CE-15#ping 192.168.6.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/11/13 ms
CE-15#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

Figura 255.

Prueba de conectividad hacia CE-17

```
CE-15#ping 192.168.6.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/12/15 ms
CE-15#
```

Nota. Ejemplo de prueba de conectividad. Elaboración propia, realizado con EVE-NG.

CONCLUSIONES

1. Se determina que la red MPLS es una técnica de red eficiente y escalable que permite a los proveedores ISP ofrecer servicios de alta calidad ya que utiliza etiquetas para enrutar paquetes a través de la red, lo que reduce los problemas de latencia y mejora el rendimiento de la red, a pesar de la aparición de nuevas tecnologías, MPLS sigue siendo ampliamente utilizado debido a su confiabilidad y capacidad para soportar diferentes tipos de tráfico.

2. Las tecnologías de L3VPN y L2VPN son soluciones de red seguras cuando se implementan y configuran correctamente sin embargo siempre es recomendable mantener actualizados todos los protocolos para garantizar una red segura y confiable.

3. La simulación de redes corporativas antes de su implementación en los entornos reales es imprescindible porque ayuda a identificar problemas de configuración que puedan impactar de forma negativa a otros clientes, por lo tanto, es una práctica que puede reducir los riesgos y errores en la red.

RECOMENDACIONES

1. Analizar el diseño y planificación con antelación de la red que cubran los aspectos de ancho de banda, asignación de etiquetas y cantidad de equipos a utilizar.

2. Considerar la topología hub and spoke ya que es de los diseños más utilizados por los proveedores de servicios por brindar opciones de redundancia y contención ante posibles fallas que puedan ocurrir dentro de la red.

3. Verificar la correcta configuración de los protocolos de enrutamiento para cada cliente esto incluye la selección del protocolo más adecuado para garantizar la estabilidad de la red y la transferencia optima de los paquetes de información.

4. Realizar todas las pruebas necesarias mediante un simulador de red ya que es importante para detectar y corregir problemas de configuración, estas pruebas deben de cubrir todas las situaciones posibles para confirmar que la solución funcione de manera confiable.

REFERENCIAS

Cisco. (2019). *Implementing MPLS Layer 3 VPNs* [Implementando MPLS Layer 3 VPNs]. Cisco. https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5000/vpn/61x/b-ncs5000-l3vpn-configuration-guide-61x/b-ncs5000-l3vpn-configuration-guide-60x_chapter_01.html

Cisco. (2019). *Layer 3 VPNs (L3VPN).* Cisco. https://www.cisco.com/c/en/us/products/ios-nx-os-software/layer-3-vpns-l3vpn/index.html

Cisco. (2019). *IP routing: BGP Configuration Guide* [Enrutamiento IP: Guía de configuración de BGP]. *Cisco ASR 1000 Series Aggregation* [Agregación de la serie Cisco ASR 1000]. Cisco. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/configuring-a-basic-bgp-network.html

Cisco. (2023). *Guia de diseño OSPF.* Cisco. https://www.cisco.com/c/es_mx/support/docs/ip/open-shortest-path-first-ospf/7039-1.html

Cisco, S. (2012). *Implementing Cisco Service Provider Next-Generation Edge Network Services* [Implementación de servicios de red perimetral próxima generación del proveedor de servicios Cisco]. Vol. 1. Cisco System.

De Ghein, L. (2007). Fundamentos de MPLS. *Cisco Press, 1*(1), 24-95. https://doc.lagout.org/network/Cisco/CCIE/CCIE%20SP/CiscoPress%20-%20MPLS%20Fundamentals.pdf

Frausto, A. (2019). *L2VPN: Layer 2 Virtual Private Network, solución de transporte para proveedores de servicio y análisis de problemas*. Cisco. https://community.cisco.com/t5/discusiones-routing-y-switching/l2vpn-layer-2-virtual-private-network-soluci%C3%B3n-de-transporte/td-p/2269310

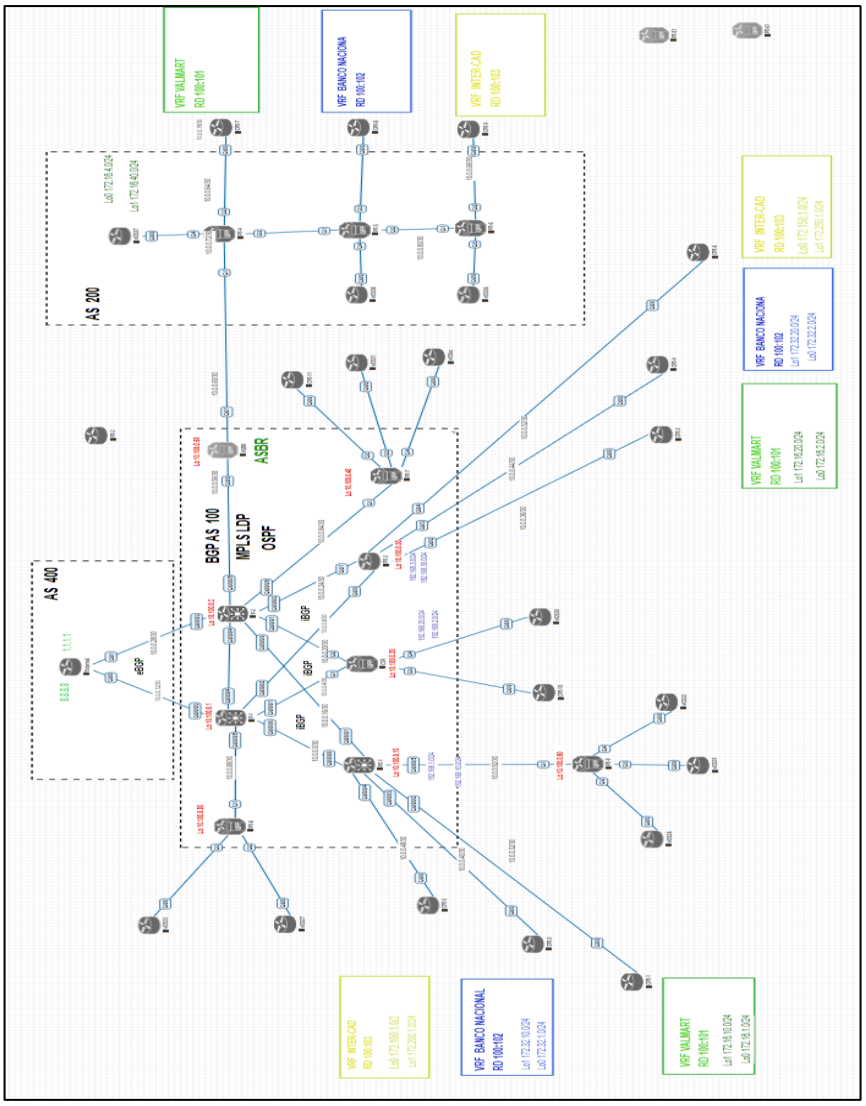
Molenaar, R. (2020). *Introduction to MPLS* [Introducción a MPLS]. NetworkLessons.com. https://networklessons.com/mpls/introduction-to-mpls

Penaloza, D. S. (2019). *Introducción a MPLS*. Cisco. Community.https://community.cisco.com/t5/documentos-routing-y-switching/introducci%C3%B3n-a-mpls/ta-p/3407436

APÉNDICES

Apéndice 1

Diagrama completo de la red



Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 2.

Código final del enrutador P-1

```
Building configuration...
!! IOS XR Configuration 6.0.1
!! Last configuration change at Tue Jun 21 02:58:15 2022 by cisco
!
hostname P-1
interface Loopback0
 ipv4 address 10.100.0.1 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0
 ipv4 address 10.0.0.1 255.255.255.252
!
interface GigabitEthernet0/0/0/1
 ipv4 address 10.0.0.5 255.255.255.252
!
interface GigabitEthernet0/0/0/2
 ipv4 address 10.0.0.9 255.255.255.252
!
interface GigabitEthernet0/0/0/3
 ipv4 address 10.0.0.14 255.255.255.252
!
interface GigabitEthernet0/0/0/4
 shutdown
!
interface GigabitEthernet0/0/0/5
 ipv4 address 10.0.0.89 255.255.255.252
!
!
interface GigabitEthernet0/0/0/6
 shutdown
!
prefix-set ip-ipv4
  192.168.10.0/24,
```

Continuación del apéndice 2.

```
  192.168.1.0/24,
  192.168.20.0/24,
  192.168.2.0/24,
  192.168.30.0/24,
  192.168.3.0/24
end-set
!
route-policy peers
  if destination in ip-ipv4 then
    pass
  else
    drop
  endif
end-policy
!
route-policy ext-route
  pass
end-policy
!
!
router ospf 1
 router-id 10.100.0.1
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
  interface GigabitEthernet0/0/0/5
  !
 !
!
!
```

Continuación del apéndice 2.

```
router bgp 100
 bgp router-id 10.100.0.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 10.0.0.13
  remote-as 400
  address-family ipv4 unicast
   route-policy ext-route in
   route-policy peers out
  !
 !
 neighbor 10.100.0.10
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
   next-hop-self
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 neighbor 10.100.0.20
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
   next-hop-self
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 !
```

Continuación del apéndice 2.

```
            neighbor 10.100.0.30
             remote-as 100
             update-source Loopback0
             address-family ipv4 unicast
              route-reflector-client
              next-hop-self
             !
             address-family vpnv4 unicast
              route-reflector-client
             !
            !
            neighbor 10.100.0.80
             remote-as 100
             update-source Loopback0
             address-family ipv4 unicast
              route-reflector-client
              next-hop-self
             !
             address-family vpnv4 unicast
              route-reflector-client
             !
            !
           !
           !
           mpls ldp
            router-id 10.100.0.1
            interface GigabitEthernet0/0/0/0
            !
            interface GigabitEthernet0/0/0/1
            !
            interface GigabitEthernet0/0/0/2
            !
            interface GigabitEthernet0/0/0/5
            !
           !
End
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 3.

Código final del enrutador P-2

```
Building configuration...
!! IOS XR Configuration 6.0.1
!! Last configuration change at Mon Jun 20 04:05:53 2022 by cisco
!
hostname P-2
interface Loopback0
 ipv4 address 10.100.0.2 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0
 ipv4 address 10.0.0.17 255.255.255.252
!
interface GigabitEthernet0/0/0/1
 ipv4 address 10.0.0.21 255.255.255.252
!
interface GigabitEthernet0/0/0/2
 ipv4 address 10.0.0.25 255.255.255.252
!
interface GigabitEthernet0/0/0/3
 ipv4 address 10.0.0.30 255.255.255.252
!
interface GigabitEthernet0/0/0/4
 shutdown
!
interface GigabitEthernet0/0/0/5
 ipv4 address 10.0.0.57 255.255.255.252
!
interface GigabitEthernet0/0/0/6
 ipv4 address 10.0.0.85 255.255.255.252
!
!
prefix-set ip-ipv4
  192.168.10.0/24,
```

Continuación del apéndice 3.

```
  192.168.1.0/24,
  192.168.20.0/24,
  192.168.2.0/24,
  192.168.30.0/24,
  192.168.3.0/24
end-set
!
route-policy peers
  if destination in ip-ipv4 then
    pass
  else
    drop
  endif
end-policy
!
route-policy ext-route
  pass
end-policy
!
!
router ospf 1
 router-id 10.100.0.2
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
  interface GigabitEthernet0/0/0/5
  !
  interface GigabitEthernet0/0/0/6
  !
 !
```

Continuación del apéndice 3.

```
!
!
router bgp 100
 bgp router-id 10.100.0.2
 address-family ipv4 unicast
  network 10.100.0.2/32
 !
 address-family vpnv4 unicast
 !
 neighbor 10.0.0.29
  remote-as 400
  address-family ipv4 unicast
   route-policy ext-route in
   route-policy peers out
  !
 !
 neighbor 10.100.0.5
  remote-as 200
  ebgp-multihop 255
  update-source Loopback0
  address-family vpnv4 unicast
   route-policy ext-route in
   route-policy ext-route out
   next-hop-unchanged
  !
 !
 neighbor 10.100.0.10
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
   next-hop-self
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
```

Continuación del apéndice 3.

```
 !
!
 neighbor 10.100.0.20
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
   next-hop-self
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 neighbor 10.100.0.30
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
   next-hop-self
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 neighbor 10.100.0.40
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
   next-hop-self
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 !
```

Continuación del apéndice 3.

```
 neighbor 10.100.0.50
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
  !
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
!
!
mpls ldp
 router-id 10.100.0.2
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 interface GigabitEthernet0/0/0/5
 !
 interface GigabitEthernet0/0/0/6
 !
!
end
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 4.

Código final del enrutador PE-1

```
Building configuration...
!! IOS XR Configuration 6.0.1
!! Last configuration change at Tue Jun 21 04:21:28 2022 by cisco
!
hostname PE-1
vrf VALMART
 address-family ipv4 unicast
  import route-target
   100:101
  !
  export route-target
   100:101
  !
 !
!
vrf INTERCAD
 address-family ipv4 unicast
  import route-target
   100:103
  !
  export route-target
   100:103
  !
 !
!
vrf BANCONACIONAL
 address-family ipv4 unicast
  import route-target
   100:102
  !
  export route-target
   100:102
  !
 !
!
```

Continuación del apéndice 4.

```
!
interface Loopback0
 ipv4 address 10.100.0.10 255.255.255.255
!
interface Loopback1
 ipv4 address 192.168.1.1 255.255.255.0
!
interface Loopback2
 ipv4 address 192.168.10.1 255.255.255.0
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0
 ipv4 address 10.0.0.2 255.255.255.252
!
interface GigabitEthernet0/0/0/1
 ipv4 address 10.0.0.18 255.255.255.252
!
interface GigabitEthernet0/0/0/2
 vrf VALMART
 ipv4 address 10.0.0.33 255.255.255.252
!
interface GigabitEthernet0/0/0/3
 vrf BANCONACIONAL
 ipv4 address 10.0.0.41 255.255.255.252
!
interface GigabitEthernet0/0/0/4
 vrf INTERCAD
 ipv4 address 10.0.0.49 255.255.255.252
!
interface GigabitEthernet0/0/0/5
 ipv4 address 10.0.0.93 255.255.255.252
!
interface GigabitEthernet0/0/0/6
 shutdown
```

Continuación del apéndice 4.

```
!
route-policy prefix-ipv4
  pass
end-policy
!
router static
 vrf VALMART
  address-family ipv4 unicast
   172.16.1.0/24 10.0.0.34
   172.16.10.0/24 10.0.0.34
  !
 !
!
router ospf 1
 log adjacency changes
 router-id 10.100.0.10
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/5
  !
 !
 !
 vrf BANCONACIONAL
  redistribute bgp 100
  area 0
   interface GigabitEthernet0/0/0/3
   !
  !
 !
!
!
```

Continuación del apéndice 4.

```
router bgp 100
 bgp router-id 10.100.0.10
 address-family ipv4 unicast
  network 192.168.1.0/24
  network 192.168.10.0/24
 !
 address-family vpnv4 unicast
 !
 neighbor 10.0.0.50
  remote-as 60
 !
 neighbor 10.100.0.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
 !
 neighbor 10.100.0.2
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
 !
 neighbor 10.100.0.90
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   route-reflector-client
   next-hop-self
  !
  !
  address-family vpnv4 unicast
```

Continuación del apéndice 4.

```
 !
!
!
vrf VALMART
 rd 100:101
 address-family ipv4 unicast
  redistribute connected
  redistribute static
 !
!
vrf INTERCAD
 rd 100:103
 bgp router-id 10.0.0.49
 address-family ipv4 unicast
 !
 neighbor 10.0.0.50
  remote-as 60
  address-family ipv4 unicast
   route-policy prefix-ipv4 in
   route-policy prefix-ipv4 out
   as-override
  !
 !
!
vrf BANCONACIONAL
 rd 100:102
 address-family ipv4 unicast
  redistribute ospf 1 match internal external
 !
!
!
!
mpls ldp
 router-id 10.100.0.10
 interface GigabitEthernet0/0/0/0
 !
```

Continuación del apéndice 4.

```
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/5
 !
!
end
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 5.

Código final del enrutador PE-2

```
Building configuration...

Current configuration: 7546 bytes
!
! Last configuration change at 01:01:53 UTC Fri Oct 7 2022
!
version 16.12
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname PE-2
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
 destination transport-method http
```

Continuación del apéndice 5.

```
  no destination transport-method email
!
login on-success log
!
subscriber templating
!
mpls label protocol ldp
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-534679223
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-534679223
 revocation-check none
 rsakeypair TP-self-signed-534679223
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
license udi pid CSR1000V sn 9U1HE7FASR8
diagnostic bootup level minimal
memory free low-watermark processor 72406
!
spanning-tree extend system-id
!
redundancy
!
pseudowire-class WIRE-1
 encapsulation mpls
!
interface Loopback0
 ip address 10.100.0.20 255.255.255.255
!
interface Loopback1
 ip address 192.168.2.1 255.255.255.0
!
```

Continuación del apéndice 5.

```
interface Loopback2
 ip address 192.168.20.1 255.255.255.0
!
interface GigabitEthernet1
 ip address 10.0.0.6 255.255.255.252
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 ip address 10.0.0.22 255.255.255.252
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
!
interface GigabitEthernet3
 no ip address
 negotiation auto
 no keepalive
 no mop enabled
 no mop sysid
 xconnect 10.100.0.40 10 encapsulation mpls pw-class WIRE-1
!
interface GigabitEthernet4
 no ip address
 negotiation auto
 no keepalive
 no mop enabled
 no mop sysid
 xconnect 10.100.0.80 20 encapsulation mpls
!
```

Continuación del apéndice 5.

```
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
!
interface GigabitEthernet7
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet8
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
router ospf 1
 router-id 10.100.0.20
 network 10.0.0.4 0.0.0.3 area 0
 network 10.0.0.20 0.0.0.3 area 0
 network 10.100.0.20 0.0.0.0 area 0
!
!
```

Continuación del apéndice 5.

```
router bgp 100
 bgp router-id 10.100.0.20
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.100.0.1 remote-as 100
 neighbor 10.100.0.1 update-source Loopback0
 neighbor 10.100.0.2 remote-as 100
 neighbor 10.100.0.2 update-source Loopback0
 !
 address-family ipv4
  network 192.168.2.0
  network 192.168.20.0
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.2 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.1 send-community extended
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community extended
 exit-address-family
!
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!
!
!
!
mpls ldp router-id Loopback0
!
!
```

Continuación del apéndice 5.

```
        !
        control-plane
        !
        !
        !
        !
        !
        !
        line con 0
         stopbits 1
        line vty 0 4
         login
        !
        !
        !
        !
        !
        !
end
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 6.

Código final del enrutador PE-3

```
!
interface Loopback0
 ip address 10.100.0.30 255.255.255.255
!
interface Loopback1
 ip address 192.168.3.1 255.255.255.0
!
interface Loopback2
 ip address 192.168.30.1 255.255.255.0
!
interface GigabitEthernet0/0
 ip address 10.0.0.10 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls ip
!
interface GigabitEthernet0/1
 ip address 10.0.0.26 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls ip
!
interface GigabitEthernet0/2
 vrf forwarding VALMART
 ip address 10.0.0.37 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
!
interface GigabitEthernet0/3
 vrf forwarding BANCONACIONAL
 ip address 10.0.0.45 255.255.255.252
```

Continuación del apéndice 6.

```
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/4
 vrf forwarding INTERCAD
 ip address 10.0.0.53 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/5
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router ospf 102 vrf BANCONACIONAL
 router-id 10.0.0.45
 redistribute bgp 100 subnets
 network 10.0.0.44 0.0.0.3 area 0
!
!
router ospf 1
 router-id 10.100.0.30
 network 10.0.0.8 0.0.0.3 area 0
 network 10.0.0.24 0.0.0.3 area 0
 network 10.100.0.30 0.0.0.0 area 0
!
router bgp 100
 bgp router-id 10.100.0.30
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.100.0.1 remote-as 100
 neighbor 10.100.0.1 update-source Loopback0
 neighbor 10.100.0.2 remote-as 100
```

Continuación del apéndice 6.

```
 neighbor 10.100.0.2 update-source Loopback0
 !
 address-family ipv4
  network 10.100.0.30 mask 255.255.255.255
  network 192.168.3.0
  network 192.168.30.0
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.2 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.1 send-community extended
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community extended
 exit-address-family
 !
!
 address-family ipv4 vrf BANCONACIONAL
  redistribute ospf 102 match internal external 1 external 2
 exit-address-family
 !
 address-family ipv4 vrf INTERCAD
  bgp router-id 10.0.0.53
  neighbor 10.0.0.54 remote-as 60
  neighbor 10.0.0.54 activate
  neighbor 10.0.0.54 send-community both
  neighbor 10.0.0.54 as-override
 exit-address-family
 !
 address-family ipv4 vrf VALMART
  redistribute connected
  redistribute static
 exit-address-family
!
ip forward-protocol nd
!
```

Continuación del apéndice 6.

```
!
no ip http server
no ip http secure-server
ip route vrf VALMART 172.16.2.0 255.255.255.0 10.0.0.38
ip route vrf VALMART 172.16.20.0 255.255.255.0 10.0.0.38
!
!
!
mpls ldp router-id Loopback0
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 7.

Código final del enrutador PE-4

```
Building configuration...
Current configuration : 7894 bytes
!
! Last configuration change at 03:20:47 UTC Fri Jun 24 2022
!
version 16.12
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname PE-4
!
boot-start-marker
boot-end-marker
!
!
vrf definition VALMART
 rd 100:101
```

Continuación del apéndice 7.

```
 !
 address-family ipv4
  route-target export 100:101
  route-target import 100:101
 exit-address-family
!
!
mpls label protocol ldp
multilink bundle-name authenticated
l2vpn vfi context VPLS-1
 vpn id 5
 autodiscovery bgp signaling ldp
  vpls-id 200:1
!
!
redundancy
bridge-domain 30
 member GigabitEthernet4 service-instance 30
 member vfi VPLS-1
!
!
!
interface Loopback0
 ip address 10.100.0.4 255.255.255.255
!
interface GigabitEthernet1
 ip address 10.0.0.62 255.255.255.252
 negotiation auto
 mpls bgp forwarding
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 vrf forwarding VALMART
 ip address 10.0.0.73 255.255.255.252
 negotiation auto
 no mop enabled
```

Continuación del apéndice 7.

```
 no mop sysid
!
interface GigabitEthernet3
 mtu 9216
 ip address 10.0.0.65 255.255.255.252
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
interface GigabitEthernet4
 no ip address
 negotiation auto
 no mop enabled
 no mop sysid
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
!
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
router ospf 1
 router-id 10.100.0.4
```

Continuación del apéndice 7.

```
 redistribute bgp 200
 network 10.0.0.64 0.0.0.3 area 0
 network 10.100.0.4 0.0.0.0 area 0
 mpls ldp autoconfig
!
!
router bgp 200
 bgp router-id 10.100.0.4
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.0.0.61 remote-as 100
 neighbor 10.100.0.5 remote-as 200
 neighbor 10.100.0.5 update-source Loopback0
 !
 address-family ipv4
  network 10.100.0.4 mask 255.255.255.255
  neighbor 10.0.0.61 activate
  neighbor 10.0.0.61 send-community both
  neighbor 10.0.0.61 send-label
  neighbor 10.100.0.5 activate
  neighbor 10.100.0.5 send-community
  neighbor 10.100.0.5 next-hop-self
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.5 activate
  neighbor 10.100.0.5 send-community both
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.100.0.5 activate
  neighbor 10.100.0.5 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VALMART
  redistribute connected
  redistribute static
 exit-address-family
```

Continuación del apéndice 7.

```
        !
        !
        ip route vrf VALMART 172.16.4.0 255.255.255.0 10.0.0.74
        ip route vrf VALMART 172.16.40.0 255.255.255.0 10.0.0.74
        !
        !
        !
        !
        mpls ldp router-id Loopback0
!
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 8.

Código final del enrutador PE-5

```
        Building configuration...
        Current configuration : 8521 bytes
        !
        ! Last configuration change at 02:59:59 UTC Fri Jun 24 2022
        !
        version 16.12
        service timestamps debug datetime msec
        service timestamps log datetime msec
        service call-home
        platform qfp utilization monitor load 80
        no platform punt-keepalive disable-kernel-core
        platform console serial
        !
        hostname PE-5
        !
        boot-start-marker
        boot-end-marker
        !
        !
        vrf definition BANCONACIONAL
         rd 100:102
```

Continuación del apéndice 8.

```
 !
 address-family ipv4
  route-target export 100:102
  route-target import 100:102
 exit-address-family
!
!
!
mpls label protocol ldp
multilink bundle-name authenticated
l2vpn vfi context VPLS-1
 vpn id 5
 autodiscovery bgp signaling ldp
  vpls-id 200:1
!
!
redundancy
bridge-domain 30
 member GigabitEthernet4 service-instance 30
 member vfi VPLS-1
!
!
interface Loopback0
 ip address 10.100.0.5 255.255.255.255
!
interface GigabitEthernet1
 mtu 9216
 ip address 10.0.0.66 255.255.255.252
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 ip address 10.0.0.69 255.255.255.252
 negotiation auto
 mpls ip
```

Continuación del apéndice 8.

```
 no mop enabled
 no mop sysid
!
interface GigabitEthernet3
 vrf forwarding BANCONACIONAL
 ip address 10.0.0.77 255.255.255.252
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet4
 no ip address
 negotiation auto
 no mop enabled
 no mop sysid
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
router ospf 102 vrf BANCONACIONAL
 router-id 10.0.0.77
 redistribute bgp 200
```

Continuación del apéndice 8.

```
 network 10.0.0.76 0.0.0.3 area 0
!
router ospf 1
 router-id 10.100.0.5
 network 10.0.0.64 0.0.0.3 area 0
 network 10.0.0.68 0.0.0.3 area 0
 network 10.100.0.5 0.0.0.0 area 0
!
!
router bgp 200
 bgp router-id 10.100.0.5
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.100.0.2 remote-as 100
 neighbor 10.100.0.2 ebgp-multihop 255
 neighbor 10.100.0.2 update-source Loopback0
 neighbor 10.100.0.4 remote-as 200
 neighbor 10.100.0.4 update-source Loopback0
 neighbor 10.100.0.6 remote-as 200
 neighbor 10.100.0.6 update-source Loopback0
 !
 address-family ipv4
  network 10.100.0.5 mask 255.255.255.255
  neighbor 10.100.0.4 activate
  neighbor 10.100.0.4 send-community
  neighbor 10.100.0.4 route-reflector-client
  neighbor 10.100.0.6 activate
  neighbor 10.100.0.6 send-community
  neighbor 10.100.0.6 route-reflector-client
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community both
  neighbor 10.100.0.2 next-hop-unchanged
  neighbor 10.100.0.4 activate
  neighbor 10.100.0.4 send-community both
```

Continuación del apéndice 8.

```
          neighbor 10.100.0.4 route-reflector-client
           neighbor 10.100.0.6 activate
           neighbor 10.100.0.6 send-community both
           neighbor 10.100.0.6 route-reflector-client
          exit-address-family
          !
         !
          address-family l2vpn vpls
           neighbor 10.100.0.4 activate
           neighbor 10.100.0.4 send-community both
           neighbor 10.100.0.4 route-reflector-client
           neighbor 10.100.0.6 activate
           neighbor 10.100.0.6 send-community both
           neighbor 10.100.0.6 route-reflector-client
          exit-address-family
          !
          address-family ipv4 vrf BANCONACIONAL
           redistribute ospf 102 match internal external 1 external 2
          exit-address-family
         !
         ip forward-protocol nd
         ip http server
         ip http authentication local
         ip http secure-server
         !
         !
         !
         mpls ldp router-id Loopback0
!
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 9.

Código final del enrutador PE-6

```
Building configuration...

Current configuration : 7383 bytes
!
! Last configuration change at 02:03:55 UTC Fri Jun 24 2022
!
version 16.12
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname PE-6
!
boot-start-marker
boot-end-marker
!
!
vrf definition INTERCAD
 rd 100:103
 !
 address-family ipv4
  route-target export 100:103
  route-target import 100:103
 exit-address-family
!
!
mpls label protocol ldp
multilink bundle-name authenticated
!
!
!
interface Loopback0
```

Continuación del apéndice 9.

```
 ip address 10.100.0.6 255.255.255.255
!
interface GigabitEthernet1
 ip address 10.0.0.70 255.255.255.252
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 vrf forwarding INTERCAD
 ip address 10.0.0.81 255.255.255.252
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet3
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet4
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
!
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
```

Continuación del apéndice 9.

```
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
router ospf 1
 router-id 10.100.0.6
 network 10.0.0.68 0.0.0.3 area 0
 network 10.100.0.6 0.0.0.0 area 0
!
!
router bgp 200
 bgp router-id 10.100.0.6
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.100.0.5 remote-as 200
 neighbor 10.100.0.5 update-source Loopback0
 !
 address-family ipv4
  network 10.100.0.6 mask 255.255.255.255
  neighbor 10.100.0.5 activate
  neighbor 10.100.0.5 send-community
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.5 activate
  neighbor 10.100.0.5 send-community both
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.100.0.5 activate
  neighbor 10.100.0.5 send-community both
 exit-address-family
 !
```

Continuación del apéndice 9.

```
 address-family ipv4 vrf INTERCAD
  bgp router-id 10.0.0.81
  neighbor 10.0.0.82 remote-as 60
  neighbor 10.0.0.82 activate
  neighbor 10.0.0.82 send-community both
  neighbor 10.0.0.82 as-override
 exit-address-family
!
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!
!
!
!
mpls ldp router-id Loopback0
!
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 10.

Código final del enrutador PE-7

```
Building configuration...

Current configuration : 7722 bytes
!
! Last configuration change at 04:09:24 UTC Fri Jun 24 2022
!
version 16.12
service timestamps debug datetime msec
service timestamps log datetime msec
```

Continuación del apéndice 10.

```
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname PE-7
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email address
to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
  no destination transport-method email
!
!
!
mpls label protocol ldp
multilink bundle-name authenticated
l2vpn vfi context VPLS-1
 vpn id 8
 autodiscovery bgp signaling ldp
  vpls-id 100:1
!
!
spanning-tree extend system-id
!
!
redundancy
```

Continuación del apéndice 10.

```
bridge-domain 30
 member GigabitEthernet4 service-instance 30
 member vfi VPLS-1
!
!
pseudowire-class WIRE-1
 encapsulation mpls
!
!
interface Loopback0
 ip address 10.100.0.40 255.255.255.255
!
interface GigabitEthernet1
 ip address 10.0.0.86 255.255.255.252
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 no ip address
 negotiation auto
 no keepalive
 no mop enabled
 no mop sysid
 xconnect 10.100.0.20 10 encapsulation mpls pw-class WIRE-1
!
interface GigabitEthernet3
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet4
 no ip address
 negotiation auto
```

Continuación del apéndice 10.

```
 no mop enabled
 no mop sysid
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
!
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet6
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet7
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet8
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
!
```

Continuación del apéndice 10.

```
router ospf 1
 router-id 10.100.0.40
 network 10.0.0.84 0.0.0.3 area 0
 network 10.100.0.40 0.0.0.0 area 0
!
!
router bgp 100
 bgp router-id 10.100.0.40
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.100.0.2 remote-as 100
 neighbor 10.100.0.2 update-source Loopback0
 neighbor 10.100.0.80 remote-as 100
 neighbor 10.100.0.80 update-source Loopback0
 neighbor 10.100.0.90 remote-as 100
 neighbor 10.100.0.90 update-source Loopback0
 !
 address-family ipv4
  neighbor 10.100.0.2 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community extended
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.100.0.80 activate
  neighbor 10.100.0.80 send-community both
  neighbor 10.100.0.90 activate
  neighbor 10.100.0.90 send-community both
 exit-address-family
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
```

Continuación del apéndice 10.

```
!
mpls ldp router-id Loopback0
!
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 11.

Código final del enrutador PE-8

```
Building configuration...

Current configuration : 7979 bytes
!
! Last configuration change at 00:11:11 UTC Fri Jun 24 2022
!
version 16.12
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname PE-8
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email address
to send SCH notifications.
```

Continuación del apéndice 11.

```
contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
  no destination transport-method email
!
!
!
!
login on-success log
!
!
!
!
!
!
!
subscriber templating
!
!
!
!
!
!
mpls label protocol ldp
multilink bundle-name authenticated
l2vpn vfi context VPLS-1
 vpn id 8
 autodiscovery bgp signaling ldp
  vpls-id 100:1
!
!
!
!
!
!!
license udi pid CSR1000V sn 9CHNO03AIA8
```

Continuación del apéndice 11.

```
diagnostic bootup level minimal
memory free low-watermark processor 72406
!
!
spanning-tree extend system-id
!
!
redundancy
bridge-domain 20
bridge-domain 30
 member GigabitEthernet3 service-instance 30
 member vfi VPLS-1
!
!
!
!
pseudowire-class WIRE-2
 encapsulation mpls
!
l2 vfi LAN-1 manual
 vpn id 100
 bridge-domain 20
 neighbor 10.100.0.20 20 pw-class WIRE-2
 neighbor 10.100.0.90 20 pw-class WIRE-2
!
!
!
interface Loopback0
 ip address 10.100.0.80 255.255.255.255
!
interface GigabitEthernet1
 ip address 10.0.0.90 255.255.255.252
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
```

Continuación del apéndice 11.

```
interface GigabitEthernet2
 no ip address
 negotiation auto
 no mop enabled
 no mop sysid
 service instance 20 ethernet
  encapsulation dot1q 20
  rewrite ingress tag pop 1 symmetric
  bridge-domain 20
 !
!
interface GigabitEthernet3
 no ip address
 negotiation auto
 no mop enabled
 no mop sysid
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
!
interface GigabitEthernet4
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet5
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet6
```

Continuación del apéndice 11.

```
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet7
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
!
interface GigabitEthernet8
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface BDI20
 no ip address
 no mop enabled
 no mop sysid
!
router ospf 1
 router-id 10.100.0.80
 network 10.0.0.88 0.0.0.3 area 0
 network 10.100.0.80 0.0.0.0 area 0
!
!
router bgp 100
 bgp router-id 10.100.0.80
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.100.0.1 remote-as 100
```

Continuación del apéndice 11.

```
neighbor 10.100.0.1 update-source Loopback0
 neighbor 10.100.0.40 remote-as 100
 neighbor 10.100.0.40 update-source Loopback0
 neighbor 10.100.0.90 remote-as 100
 neighbor 10.100.0.90 update-source Loopback0
 !
 address-family ipv4
  neighbor 10.100.0.1 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.1 activate
  neighbor 10.100.0.1 send-community extended
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.100.0.40 activate
  neighbor 10.100.0.40 send-community both
  neighbor 10.100.0.90 activate
  neighbor 10.100.0.90 send-community both
 exit-address-family
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!
mpls ldp router-id Loopback0
!
!
!
control-plane
!
```

Continuación del apéndice 11.

```
        !
        !
        line con 0
         stopbits 1
        line vty 0 4
         login
        !
        !
        !
end
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 12.

Código final del enrutador PE-9

```
        Building configuration...

        Current configuration : 7257 bytes
        !
        ! Last configuration change at 03:35:08 UTC Fri Jun 24 2022
        !
        version 16.12
        service timestamps debug datetime msec
        service timestamps log datetime msec
        service call-home
        platform qfp utilization monitor load 80
        no platform punt-keepalive disable-kernel-core
        platform console serial
        !
        hostname PE-9
        !
        boot-start-marker
        boot-end-marker
        !
```

Continuación del apéndice 12.

```
!
!
no aaa new-model
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email address
to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
  no destination transport-method email
!
!
!
!
mpls label protocol ldp
multilink bundle-name authenticated
l2vpn vfi context VPLS-1
 vpn id 8
 autodiscovery bgp signaling ldp
  vpls-id 100:1
!
!
!
redundancy
bridge-domain 30
 member GigabitEthernet4 service-instance 30
 member vfi VPLS-1
!
!
!
interface Loopback0
 ip address 10.100.0.90 255.255.255.255
!
interface GigabitEthernet1
 ip address 10.0.0.94 255.255.255.252
```

Continuación del apéndice 12.

```
 negotiation auto
 mpls ip
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet3
 no ip address
 negotiation auto
 no keepalive
 no mop enabled
 no mop sysid
 xconnect 10.100.0.80 20 encapsulation mpls
!
!
interface GigabitEthernet4
 no ip address
 negotiation auto
 no mop enabled
 no mop sysid
 service instance 30 ethernet
  encapsulation dot1q 30
  rewrite ingress tag pop 1 symmetric
 !
!
router ospf 1
 router-id 10.100.0.90
 network 10.0.0.92 0.0.0.3 area 0
 network 10.100.0.90 0.0.0.0 area 0
!
!
```

Continuación del apéndice 12.

```
router bgp 100
 bgp router-id 10.100.0.90
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.100.0.10 remote-as 100
 neighbor 10.100.0.10 update-source Loopback0
 neighbor 10.100.0.40 remote-as 100
 neighbor 10.100.0.40 update-source Loopback0
 neighbor 10.100.0.80 remote-as 100
 neighbor 10.100.0.80 update-source Loopback0
 !
 address-family ipv4
  neighbor 10.100.0.10 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.10 activate
  neighbor 10.100.0.10 send-community extended
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.100.0.40 activate
  neighbor 10.100.0.40 send-community both
  neighbor 10.100.0.80 activate
  neighbor 10.100.0.80 send-community both
 exit-address-family
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!
mpls ldp router-id Loopback0
!
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.

Apéndice 13.

Código final del enrutador PE-ASBR

```
Building configuration...

Current configuration : 3839 bytes
!
version 15.6
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ASBR
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
ethernet lmi ce
!
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
!
!
!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
mpls label protocol ldp
!
```

Continuación del apéndice 13.

```
!
!
interface Loopback0
 ip address 10.100.0.50 255.255.255.255
!
interface GigabitEthernet0/0
 ip address 10.0.0.58 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls ip
!
interface GigabitEthernet0/1
 ip address 10.0.0.61 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls bgp forwarding
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
!
router ospf 1
 router-id 10.100.0.50
 redistribute bgp 100 subnets
```

Continuación del apéndice 13.

```
 network 10.0.0.56 0.0.0.3 area 0
 network 10.100.0.50 0.0.0.0 area 0
!
!
router bgp 100
 bgp router-id 10.100.0.50
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.0.0.62 remote-as 200
 neighbor 10.100.0.2 remote-as 100
 neighbor 10.100.0.2 update-source Loopback0
 !
 address-family ipv4
  network 10.100.0.50 mask 255.255.255.255
  neighbor 10.0.0.62 activate
  neighbor 10.0.0.62 send-community both
  neighbor 10.0.0.62 send-label
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community both
  neighbor 10.100.0.2 next-hop-self
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.100.0.2 activate
  neighbor 10.100.0.2 send-community both
  neighbor 10.100.0.2 next-hop-self
 exit-address-family
!
!
mpls ldp router-id Loopback0
```

Nota. Red completa. Elaboración propia, realizado con EVE-NG.