ANÁLISIS DEL DESEMPEÑO DE UNA RED BGP/IPV6/VPN/MPLS APLICANDO LOS MÉTODOS *ROUTE REFLECTOR* Y *CONFEDERATIONS BGP* EN UN ENTORNO DE PRUEBAS VIRTUALIZADO

Universidad
del Cauca

Carlos Alberto Córdoba Paredes
Luis Miguel Buitrón Mondragón

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Línea de Investigación Redes y Servicios de Telecomunicaciones
Popayán, 2023

ANÁLISIS DEL DESEMPEÑO DE UNA RED BGP/IPV6/VPN/MPLS APLICANDO LOS MÉTODOS *ROUTE REFLECTOR* Y *CONFEDERATIONS BGP* EN UN ENTORNO DE PRUEBAS VIRTUALIZADO

Universidad
del Cauca

Documento final de Trabajo de Grado presentado a la Facultad de Ingeniería en Electrónica y Telecomunicaciones de la Universidad del Cauca para optar por el título de Ingeniero en Electrónica y Telecomunicaciones

Carlos Alberto Córdoba Paredes
Luis Miguel Buitrón Mondragón

Director:
Msc. Francisco Javier Terán Cuarán

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Línea de Investigación Redes y Servicios de Telecomunicaciones
Popayán, 2023

Dedico este trabajo a mis Padres y mi Esposa
por ser siempre mi apoyo incondicional,
a mis maestros por darme la orientación
de ser un gran profesional, y agradezco
a Dios por obtener este logro, el cuál es
la culminación de un gran periodo,
y el inicio de nuevas oportunidades.

Carlos Alberto Córdoba Paredes

Dedico este logro a mi familia, quienes han sido
mi fuente constante de inspiración y apoyo a
lo largo de toda mi vida. Agradezco profundamente
su amor incondicional y su sabiduría que han sido
fundamentales para guiar mis pasos hasta cumplir
este logro. También quiero dedicar este trabajo
a mis amigos y seres queridos que me han alentado,
brindado su amistad y apoyo a lo largo de esta travesía.
Sin su aliento y apoyo, este logro no habría sido posible.

Luis Miguel Buitrón Mondragón

AGRADECIMIENTOS

Los autores expresan su agradecimiento al Msc. Francisco Javier Terán Cuarán, director del trabajo de grado, experto en Redes y Seguridad por su valiosa orientación.

A la Línea de Investigación de Redes y Servicios de Telecomunicaciones, al Departamento de Telecomunicaciones de la Facultad de Ingeniería en Electrónica y Telecomunicaciones de la Universidad del Cauca, por sus aportes y contribuciones en el desarrollo de este trabajo de grado.

También se expresa un agradecimiento muy especial a su familia y conocidos por el acompañamiento y apoyo incondicional, sin los cuales no sería posible el desarrollo del Proyecto.

RESUMEN

En la actualidad la fabricación de tecnología como celulares, tabletas, computadores, entre otros dispositivos está en auge. Estos dispositivos tienen la capacidad de conectarse y navegar hacia internet ya sea por medio cableado o wifi a través de una dirección de Protocolo de Internet versión 4 (*Internet Protocol version 4,* IPv4); pero este protocolo tiene un número limitado de direcciones al ser números binarios de 32 bits, conformado por 4 octetos donde la máxima cantidad de IP que se pueden obtener es de aproximadamente 4000 millones, por lo cual no se podrán conectar a nivel global los nuevos dispositivos cuando se agoten estas. En la evolución del Internet de las Cosas (*Internet of Things*, IOT) la alta demanda de direcciones IP de los dispositivos y las pocas disponibles, hace que se migre la tecnología a otro protocolo que brinde mayor cantidad de direcciones para que soporte todos los dispositivos que quieran conectarse a internet, el cual es el Protocolo de Internet versión 6 (*Internet Protocol version 6,* IPv6), un protocolo que brinda un número mucho mayor (340 decillones) con el que se generan nuevos tipos de direcciones IP más largos, robustos y con mejores características, asegurando que se tendrán IP disponibles para que se conecten billones de dispositivos durante los próximos años.

En un mundo impulsado por la interconexión digital y la expansión constante de las tecnologías de comunicación, la eficiencia y el rendimiento de las redes se han convertido en factores cruciales para garantizar la transmisión segura y confiable de datos. La convergencia de tecnologías como BGP, IPv6, VPN y MPLS han revolucionado la arquitectura de las comunicaciones, permitiendo una conectividad avanzada y una segmentación efectiva de datos. El Protocolo de Pasarela de Borde (*Border Gateway Protocol*, BGP) que es el más usado en la actualidad por los Proveedores de Servicios de Internet (*Internet Service Provider*, ISP), se encarga del enrutamiento y la comunicación de los paquetes con otros *Routers* pertenecientes a diferentes Sistemas Autónomos (*Autonomous System*, AS), pero este acarrea problemas de seguridad, señalización, escalabilidad y mayor uso de ancho de banda; para minimizar este problema se utilizan las redes sobre MPLS/VPN/BGP, Protocolo de Pasarela de Borde (BGP), Red Privada Virtual (*Virtual Private Network*, VPN), Conmutación de Etiquetas Multiprotocolo (*Multiprotocol Label Switching*, MPLS), siendo MPLS una tecnología que hace uso de etiquetas y proporciona una alta velocidad en la transferencia de diferentes tipos de datos a través de una misma red, mejora el flujo de trabajo de Internet y el aislamiento de los clientes. Por otra parte, la VPN mejora la seguridad y privacidad, permitiendo al proveedor optimizar la red, al interconectar varios clientes VPN a un mismo *Router* disminuyendo conexiones físicas.

Cuando se implementa el protocolo BGP en AS, la regla *Split Horizon* prohíbe a un *Router* que recibe una actualización de enrutamiento por una interfaz, reenviarla por la misma, es decir una ruta aprendida por un *Router* con Protocolo de Pasarela de Borde Interno *(Internal Border Gateway Protocol,* iBGP*)* no puede ser propagada a otro *Router* iBGP vecino, por lo cual se necesita de una topología mallada iBGP *Full Mesh,* donde el número de conexiones BGP-TCP se calcula mediante la ecuación 1:

$$N° \, Sesiones \, (BGP - TCP) = \frac{N * (N-1)}{2}, \qquad N = Cantidad \, de \, Routers \tag{1}$$

Esta regla evita problemas de bucles de enrutamiento, pero levanta demasiadas sesiones BGP-TCP, para lo cual el método *Route Reflector,* modifica la regla para que el reflector de rutas pueda propagar las rutas aprendidas de un iBGP a otro iBGP vecino reduciendo el número de sesiones BGP-TCP en el AS. El método *Confederations BGP* también es otra solución para resolver los problemas de escalabilidad creados al tener todos los *Routers* iBGP totalmente mallados, por lo que subdivide efectivamente un AS grande en subsistemas. Al igual que con *Route Reflector* se reduce la cantidad de sesiones TCP para mantener conexiones entre dispositivos de enrutamiento iBGP.

La red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) representa una combinación sinérgica de tecnologías que ha permitido la creación de redes escalables y seguras, esenciales para el funcionamiento de aplicaciones críticas y la comunicación global. Sin embargo, la complejidad de estas redes, junto con la diversidad de aplicaciones y la expansión constante de usuarios, ha planteado desafíos en términos de convergencia, latencia y eficiencia en el enrutamiento.

Los métodos de *Route Reflector* y *Confederations BGP* emergen como posibles soluciones para abordar los problemas de escalabilidad y convergencia, donde el método *Route Reflector* busca optimizar la propagación de rutas reduciendo la necesidad de conexiones físicas y el método *Confederations BGP* busca segmentar la red en subdominios autónomos para mejorar la escalabilidad y el enrutamiento interno.

ABSTRACT

Currently, the manufacture of technology such as cell phones, tablets, and computers, among other devices is booming. These devices have the ability to connect and surf to the Internet either by wired means or wifi through an address of Internet Protocol version 4 (IPv4); but this protocol has a limited number of addresses being binary 32-bit numbers, made up of 4 bytes where the maximum amount of IP that can be obtained is approximately 4000 million, so that new devices will not be able to connect globally when they are exhausted. In the evolution of the Internet of Things (IOT), the high demand for IP addresses from devices and the few available ones has led the technology to migrate to another protocol that provides more addresses to support all devices that want to connect to the Internet, which is the Internet Protocol version 6 (IPv6), a protocol that provides a much larger number (340 decillions) with which new types of longer, more robust and better-featured IP addresses are generated, ensuring that there will be IPs available for billions of devices to connect over the next few years.

In a world driven by digital interconnection and the constant expansion of communication technologies, the efficiency and performance of networks have become crucial factors in ensuring secure and reliable data transmission. The convergence of technologies such as BGP, IPv6, VPN, and MPLS have revolutionized the architecture of communications, allowing advanced connectivity and effective data segmentation. The Border Gateway Protocol (BGP), which is currently the most widely used by Internet Service Providers (ISPs), is responsible for the routing and communication of packets with other Routers belonging to different Autonomous Systems (AS), but this causes problems of security, signaling, scalability and increased bandwidth usage; to minimize this problem, networks on MPLS/VPN/BGP, Border Gateway Protocol (BGP), Virtual Private Network (VPN), Multiprotocol Label Switching (MPLS), MPLS is a technology that makes use of labels and provides a high speed in the transfer of different types of data over the same network, improves the Internet workflow and the isolation of customers. On the other hand, the VPN improves security and privacy, allowing the provider to optimize the network by interconnecting several VPN clients to the same Router reducing physical connections.

When implementing the BGP protocol in AS, the Split Horizon rule prohibits a Router that receives a routing update through an interface, from forwarding it through it, i. e. a route learned by a Router with Internal Border Gateway Protocol (iBGP) cannot be propagated to another neighboring iBGP Router, so a mesh iBGP Full Mesh topology is needed, where the number of BGP-TCP connections is calculated by equation 1:

$$N° \, Sesiones \, (BGP - TCP) = \frac{N * (N - 1)}{2}, \qquad N = Cantidad \, de \, Routers \tag{1}$$

This rule avoids routing loop problems but raises too many BGP-TCP sessions, for which the Route Reflector method modifies the rule so that the Route Reflector can propagate learned routes from one iBGP to another neighboring iBGP by reducing the number of BGP-TCP sessions in the AS. The Confederations BGP method is also another solution to solve the scalability problems created by having all iBGP Routers fully mesh, thus effectively subdividing a large AS into subsystems. As with Route Reflector, the number of TCP sessions to maintain connections between iBGP routing devices is reduced.

The BGP/IPV6/VPN/MPLS (6VPE over MPLS) network represents a synergistic combination of technologies that has enabled the creation of scalable and secure networks, essential for the operation of critical applications and global communication. However, the complexity of these networks, together with the diversity of applications and the ever-expanding number of users, has posed challenges in terms of convergence, latency, and routing efficiency.

The Route Reflector and Confederations BGP methods emerge as possible solutions to address scalability and convergence issues, where the Route Reflector method seeks to optimize route propagation by reducing the need for physical connections and the Confederations BGP method seeks to segment the network into autonomous subdomains to improve scalability and internal routing.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

ACRÓNIMOS

ACK	*Acknowledgment,* Reconocimiento.
AS	*Autonomous System*, Sistema Autónomo.
AS-Path	*Autonomous System Path*, Ruta de Sistema Autónomo.
BGP	*Border Gateway Protocol*, Protocolo de Pasarela de Borde.
CE	*Customer Edge,* Borde de Cliente.
eBGP	*External Border Gateway Protocol*, Protocolo de Pasarela de Borde Externo.
EGP	*External Gateway Protocol,* Protocolo de Pasarela Exterior.
FEC	*Forwarding Equivalence Class,* Clase de Equivalencia de Reenvío.
GNS3	*Graphic Network Simulation*, Simulador Gráfico de Redes
iBGP	*Internal Border Gateway Protocol*, Protocolo de Pasarela de Borde Interno.
ICMPv6	*Internet Control Message Protocol version 6,* Protocolo de Mensajes de Control de Internet versión 6.
IETF	*Internet Engineering Task Force,* Grupo Operativo de Ingeniería de Internet.
IGP	*Internal Gateway Protocol,* Protocolo de Pasarela Interior
IOS	*Internetwork Operating System,* Sistema Operativo de Interconexión.
IP	*Internet Protocol*, Protocolo de Internet.
IoT	*Internet of Things, Internet de las Cosas.*
IPsec	*Internet Protocol Security,* Protocolo de Internet Seguro.
IPv4	*Internet Protocol Version 4*, Protocolo de Internet Versión 4.
IPv6	*Internet Protocol Version 6*, Protocolo de Internet Versión 6.
iRTT	*Initial Round Trip Time,* Tiempo de Ida y Vuelta Inicial.
ISP	*Internet Service Provider*, Proveedor de Servicios de Internet.
LDP	*Label Distribution Protocol,* Protocolo de Distribución de Etiquetas.
LER	*Router Label Edge,* Enrutador de Borde de Etiqueta.
LIB	*Label Information Base,* Tabla de Información de Etiquetas.
LFIB	*Label Forwarding Information Base,* Tabla de Enrutamiento de Etiquetas.
LSP	*Label Switched Patch,* Camino Virtual de Etiqueta Conmutada.
LSR	*Router Label Switch,* Enrutador de Conmutación de Etiquetas.
MED	*Multi-Exit Discriminator,* Discriminador de Múltiples Salidas.
MP-BGP	*Multi Protocol Border Gateway,* Multiprotocolo de Pasarela de Borde.
MPLS	*Multiprotocol Label Switching*, Conmutación de Etiquetas Multiprotocolo.
OSI	*Open Systems Interconnection,* Interconexion de Sistemas Abiertos.
P	*Provider,* Proveedor.
PE	*Provider Edge*, Borde del Proveedor.
QoS	*Quality of Services*, Calidad de Servicio.
RD	*Route Distinguishers,* Distintor de Ruta.
RAM	*Random Access Memory,* Memoria de Acceso Aleatorio.
RR	*Route Reflector,* Reflector de Ruta.
RRC	*Route Reflector Client,* Cliente del Reflector de Ruta.
RSVP	*Resource Reservation Protocol*, Protocolo de Reserva de Recursos.
RT	*Route Target,* Objetivo de Ruta.
RTT	*Round Trip Time,* Tiempo de Ida y Vuelta.
SSD	*Solid State Drive,* Unidad de Estado Sólido.
SYN	*Message Synchronize,* Mensaje de Sincronización.
SYN-ACK	*Message Synchronize- Acknowledgment,* Mensaje de Sincronización y Reconocimiento.
TCP	*Transmission Control Protocol*, Protocolo de Control de Transmisión.

TE	*Traffic Engineering*, Ingeniería de Tráfico.
TTL	*Time to Live,* Tiempo de Vida.
UDP	*User Datagram Protocol,* Protocolo de Datagramas de Usuario.
VPE	*Virtual Provider Edge,* Enrutador Virtual de Borde de Proveedor.
VPLS	*Virtual Private LAN Services,* Servicios de LAN Privada Virtual.
VPN	*Virtual Private Network*, Red Privada Virtual.
VPRN	*Virtual Private Routed Network,* Red Enrutada Privada Virtual.
VR	*Virtual Router,* Enrutador Virtual.
VRF	*Virtual Routing and Forwarding*, Enrutamiento Virtual y Reenvío.
WAN	*Wide Area Network,* Red de Área Amplia.
6VPE	*IPv6 Virtual Provider Edge,* Enrutador Virtual de Borde de Proveedor IPv6.

INTRODUCCIÓN

En la era actual de interconexión global y creciente dependencia de las redes de comunicación, la optimización y el rendimiento de las infraestructuras de red son cruciales para garantizar la eficiencia y la seguridad en la transmisión de datos. El uso del Protocolo de Internet versión 6 (*Internet Protocol version 6*, IPv6), Redes Privadas Virtuales (*Virtual Private Network*, VPN), Enrutadores Virtuales (*Virtual Router*, VR), Tablas Virtuales de Enrutamiento (*Virtual Routing Forwarding*, VRF), en el contexto del Protocolo Conmutación de Etiquetas Multiprotocolo (*Multiprotocol Label Switching*, MPLS), el Protocolo de Pasarela de Borde (*Border Gateway Protocol*, BGP) y las redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) han surgido como pilares fundamentales en la arquitectura para garantizar la seguridad y escalabilidad de las redes modernas. Sin embargo, el desafío persiste en cómo mantener un equilibrio entre la escalabilidad, la convergencia rápida y la calidad de las conexiones en medio del constante aumento de usuarios y aplicaciones.

Este trabajo de grado plantea el diseño y análisis del rendimiento de una red basada en BGP/IPV6/VPN/MPLS (*6VPE over MPLS*), con un enfoque en la aplicación de dos métodos clave para la optimización: *Route Reflector* y *Confederations BGP*. El estudio se lleva a cabo en un entorno de pruebas virtualizado (GNS3[1]) para evaluar cómo estos métodos influyen en la eficiencia y calidad de la red. El *Route Reflector* permite optimizar la distribución de rutas en la red y reduce la necesidad de conexiones físicas (*Full Mesh*) mejorando la convergencia. Por otro lado, *Confederations BGP* se emplea para subdividir la red en subdominios autónomos, mejora la escalabilidad y facilita el enrutamiento interno.

La metodología de investigación se divide en varias etapas. En primer lugar, se establece una topología de pruebas que simula el comportamiento de una red real donde se configura la red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) en un entorno virtualizado. Posteriormente, se aplican los métodos *Route Reflector* y *Confederations BGP* en escenarios específicos.

Los resultados se recopilan y analizan detalladamente, enfocándose en métricas clave como tiempos de respuesta, la calidad de las conexiones y la latencia, evaluando el impacto en el rendimiento de la red, a través de comparaciones exhaustivas entre los escenarios sin y con la implementación de estos métodos. Los hallazgos obtenidos a partir de este análisis proporcionan información valiosa sobre cómo los métodos de *Route Reflector* y *Confederations BGP* impactan el desempeño de una red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) en un entorno virtualizado. Estos resultados pueden ser utilizados como base para la toma de decisiones en el diseño y la optimización de redes IPv6 en un mundo cada vez más interconectado, orientado a la seguridad y la eficiencia.

El presente trabajo de grado se sumerge en este contexto dinámico abordando la pregunta esencial: ¿Cómo pueden los métodos *Route Reflector* y *Confederations BGP* influir en el desempeño de una red BGP/IPV6/VPN/MPLS? El objetivo general de este proyecto es analizar el desempeño de una red BGP/IPV6/VPN/MPLS aplicando los métodos *Route Reflector y Confederations BGP* para un Proveedor de Servicios de Internet en un entorno de pruebas virtualizado. Para alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

[1] GNS3 es un *software* gratuito y de código abierto que permite emular dispositivos de red de *Cisco* y otros fabricantes en un entorno virtual.

- Diseñar un escenario de red BGP/IPV6/VPN/MPLS en un entorno de pruebas virtualizado sin los métodos *Route Reflector y Confederations BGP*.
- Implementar un escenario de red BGP/IPV6/VPN/MPLS en un entorno de pruebas virtualizado con los métodos *Route Reflector y Confederations BGP*.
- Evaluar el desempeño de los escenarios implementados, mediante los parámetros *Jitter, Round Trip Time y Frame Arrival Delay* enfocados en los tiempos de ejecución del enrutamiento.

Se implementa un entorno de pruebas virtualizado, donde se simulan situaciones reales y se implementan los métodos mencionados en escenarios controlados. Mediante un análisis exhaustivo de métricas clave como tiempos de convergencia, la calidad de las conexiones y la latencia (*Round Trip Time, Frame Arrival Delay y Jitter*), se busca comprender el impacto de estas soluciones en el rendimiento general de la red.

Esta investigación no solo proporcionará una visión más profunda de cómo los métodos de *Route Reflector* y *Confederations BGP* influyen en una red BGP/IPV6/VPN/MPLS, sino que también ofrecerá valiosa información para la toma de decisiones en el diseño y la optimización de redes similares. En última instancia, se busca contribuir al desarrollo de infraestructuras de comunicación más eficientes, escalables y seguras en un mundo donde la conectividad es fundamental.

A continuación, se presenta un breve resumen de cada capítulo para dar una idea del contenido de este trabajo de grado.

Capitulo 1: Conceptos preliminares. A lo largo de esta investigación, se analiza cómo los métodos *Route Reflector* y *Confederatiosn BGP* impactan en la implementación de redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*). Se analizan las capacidades con las que cuenta MPLS, que permiten un enrutamiento más eficiente y flexible en redes, siendo un pilar fundamental para la implementación de VPN de los clientes por medio de BGP.

Capítulo 2: Diseño e implementación de la red BGP/IPV6/VPN/MPLS de un Proveedor de Servicios de Internet (*Internet Service Provider*, ISP) en un entorno de pruebas virtualizado. En este capítulo se define la metodología, las herramientas de simulación y análisis de redes. Posteriormente se diseña la red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) ISP y se emula cada escenario de pruebas con los protocolos necesarios para el cumplimiento de los objetivos. La simulación se lleva a cabo a través del *software* GNS3, donde en primera medida se realiza la simulación sin los métodos y luego se aplican de forma independiente los métodos *Route Reflector*, *Confederations BGP* y la combinación de los mismos.

Capítulo 3: Pruebas de funcionalidad red BGP/IPV6/VPN/MPLS sin métodos (Escenario 1). En este capítulo se verifica la funcionalidad de la red BGP/IPV6/VPN/MPLS de un ISP del escenario 1 sin métodos y de los protocolos TCP, BGP y MPLS para medir el desempeño, el cuál será el procedimiento que se realizará a todos los escenarios propuestos anteriormente en el siguente capítulo. Este análisis de funcionalidad de la red se lleva a cabo a través de la herramienta de análisis de protocolos y rastreo *Wireshark* que viene implementado en el *Software* GNS3.

Capítulo 4: Mediciones y Análisis de Resultados en los Escenarios. En este capítulo se muestran los resultados obtenidos para cada escenario y se realiza la comparación de parámetros de desempeño *Jitter, Round Trip Time y Frame Arrival Delay* enfocados en los tiempos de ejecución del enrutamiento. Se presentan tablas comparativas de las medidas

recopiladas con el fin de obtener una visión más clara acerca del escenario más eficiente con respecto a los parámetros de medición.

Capítulo 5: Conclusiones, recomendaciones y trabajos futuros. En este capítulo se presentan las conclusiones obtenidas a partir del desarrollo del trabajo, recomendaciones a tener en cuenta en la implementación de redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) en un entorno virtualizado y posibles trabajos futuros.

1. CONCEPTOS PRELIMINARES

En esta primera parte se resumen las tecnologías, los protocolos y el tipo de redes necesarias para el desarrollo de este proyecto.

1.1 IPv6

El Protocolo de Internet versión 6 (*Internet Protocol version 6*, IPv6) es la versión más reciente y avanzada del Protocolo de Internet IP, diseñada para abordar la limitación y agotamiento de direcciones IPv4. Ofrece una cantidad inagotable de direcciones IP únicas con un total de direcciones IP, aproximadamente 340 sextillones de direcciones.[1] Utiliza direcciones IP de 128 bits y se representa usando la notación hexadecimal en ocho bloques de cuatro dígitos separados por el carácter dos puntos (:), por ejemplo, así es como se ve una dirección IPv6 “2001:0448:1024:0000:0000:0000:0010:0001" [2].

Para estas direcciones existen reglas de representación simplificada, que consisten en [3]:

- Si en la dirección hay uno o más cuartetos consecutivos formados únicamente por ceros, estos pueden ser sustituidos por par de dos puntos (::), de este modo el ejemplo anterior se puede representar como 2001:0448:1024::0010:0001.
- La otra regla consiste en que los ceros iniciales de cada bloque se pueden omitir, por ejemplo, al aplicar esta regla a la dirección de la anterior regla que de la siguiente manera 2001:448:1024::10:1.

Otro beneficio que se obtiene con IPv6 son las mejoras en la eficiencia del enrutamiento y en la seguridad de las comunicaciones. Su diseño incluye características como autoconfiguración de direcciones, soporte nativo para seguridad a través del Protocolo de Seguridad de la Capa de Internet (*Internet Protocol Security,* IPsec) que se encarga de garantizar la seguridad de las comunicaciones a través de una red IP y un enrutamiento más eficiente gracias a la simplificación de las tablas de enrutamiento [4].

Para comprender mejor este protocolo, se proporciona una introducción a la estructura y el significado de los datagramas IPv6. La figura 1.1 muestra el formato de los datagramas IPv6.

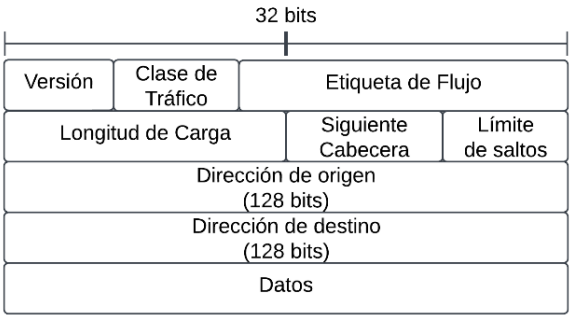


Figura 1.1 Datagrama IPv6 [5]

Los campos clave que cumplen funciones específicas en los datagramas de IPv6 son los siguientes [5]:

- Versión: Este campo contiene 4 bits, se encuentra en el encabezado del datagrama que indica la versión del protocolo IP a la que pertenece el datagrama.
- Clase de tráfico: Campo de 8 bits, se utiliza para dar prioridad a los datagramas o calidad de servicio.
- Etiqueta de flujo: Campo de 20 bits, permite que un origen y un destino marquen grupos de paquetes con requisitos similares, creando una especie de conexión virtual en la red. Esto es útil para paquetes que requieren un tratamiento especial, como garantías de retardo o ancho de banda reservado.
- Longitud de carga: Campo de 16 bits que indica la cantidad de bytes que componen la carga útil del datagrama IPv6. Esta carga es la información real que se envía, que sigue inmediatamente a la cabecera del datagrama que tiene una longitud fija de 40 *bytes*.
- Siguiente cabecera: Identifica el protocolo al que se enviará el contenido del datagrama, como el Protocolo de Control de Transmisión (*Transmission Control Protocol*, TCP) o el Protocolo de Datagramas de Usuario (*User Datagram Protocol*, UDP). Este campo utiliza los mismos valores que el campo de protocolo en la cabecera IPv4 para especificar el tipo de protocolo que manejará los datos en el datagrama IPv6.
- Límite de saltos: Funciona de manera similar al campo de Tiempo de Vida (*Time to Live*, TTL) en IPv4 y se utiliza para limitar la vida útil de un datagrama en la red disminuyendo su valor en cada salto.
- Dirección de Origen: Es la dirección IPv6 del remitente del datagrama.
- Dirección de Destino: Es la dirección IPv6 del destinatario del datagrama.
- Datos: Los "Datos" en un datagrama IPv6 son la parte de la carga útil que se extraerá y entregará al protocolo especificado en el campo "Siguiente Cabecera" cuando el datagrama alcance su destino final.

Existen tres tipos de direcciones IPv6. *Unicast, Anycast* y *Multicast* que están contenidos en dos clasificaciones, las direcciones *Global Unicast* (públicas) y *Local Unicast* (privadas) [6]:

1. *Unicast*: Una dirección *unicast* es una dirección IPv6 utilizada para identificar un único nodo de red, Los paquetes dirigidos a una dirección *unicast* se envían a una única interfaz.
2. *Multicast*: Las direcciones *multicast* se utilizan para enviar paquetes a un grupo de nodos en lugar de a un único nodo. Los *Routers* y dispositivos de red pueden enviar datos a un grupo de nodos que se han unido a una dirección *multicast* específica.
3. *Anycast*: Identifica a una o más interfaces, pero los paquetes se envían al nodo más cercano en términos de distancia de red o según sea el mejor destino desde el punto de vista de la topología de la red. Las direcciones *Anycast* son usadas para el balanceo de carga.

El Protocolo de Mensajes de Control de Internet versión 6 (*Internet Control Message Protocol version 6*, ICMPv6) es un protocolo básico de IPv6 utilizado para intercambio de mensajes de información y errores de accesibilidad en la red entre dispositivos, definido por el IETF en el RFC 4443 [7]. Controla la autoconfiguración de direcciones IPv6, detección de vecinos, selección de rutas y control de errores de otros enlaces relacionados [8].

Al ejecutar un *ping* desde un dispositivo hacia otro, se envía un paquete que incluye un mensaje ICMPv6 quien hace una solicitud al destino *"Echo Request"* que, al ser recibida por el destino, este devuelve una respuesta con un paquete que contiene un mensaje de

respuesta *"Echo Reply"* indicando que la conexión fue exitosa, como se muestra en la figura 1.2.

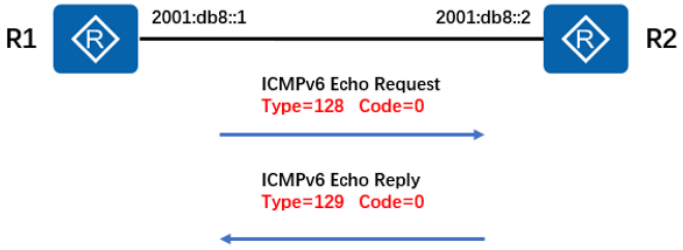


Figura 1.2 ICMPv6: Ping [8]

Por otra parte, si el mensaje enviado por el origen no encuentra al destinatario, la red se encarga de enviar una respuesta desfavorable indicando que el destino no ha sido encontrado "*Destination Unreachable*".

1.2 MPLS

El Protocolo de Conmutación de Etiquetas Multiprotocolo (*MultiProtocol Label Switching*, MPLS) es una tecnología que está diseñada para optimizar y agilizar el enrutamiento de paquetes en Redes de Comunicación, definido por Grupo Operativo de Ingeniería de Internet (*Internet Engineering Task Force*, IETF) en la RFC 3031 [9] para aplicarse a todo tipo de redes basadas en la capa de red. Trabaja entre la Capa 2 y 3 del modelo OSI (Capa 2.5) [10], como se observa en la figura 1.3.

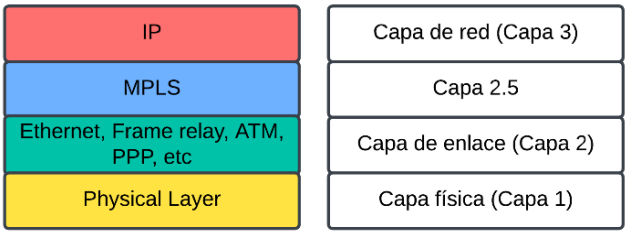


Figura 1.3 Posición de MPLS en el modelo OSI [10]

Mejora la velocidad y eficacia en el reenvío de datos a través de etiquetas de enrutamiento, donde la etiqueta decide qué camino tomar a través de la red, y deja de examinar detalladamente las direcciones IP de destino en las tablas de enrutamiento de cada *Router* de manera tradicional, se incluye la cabecera MPLS entre las cabeceras de la capa 2 y 3 como se muestra en la figura 1.4, es por ello que se dice que MPLS combina la inteligencia y la escalabilidad de protocolos de capa 3, con la confiabilidad y la gestión de la capa 2 [11].

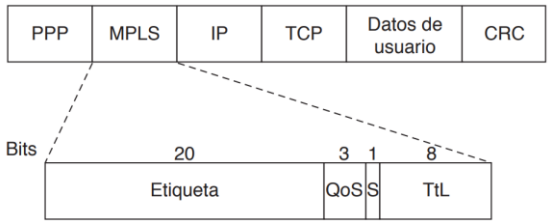


Figura 1.4 Cabecera MPLS [11]

La cabecera de MPLS está conformada por 32 bits de longitud, divididos en 4 campos [11] [12]:

- Etiqueta: Este campo está conformado por 20 bits, donde se asigna un valor numérico (etiqueta) a un paquete cuando ingresa a una red MPLS para poder identificarlo y posteriormente realizar el enrutamiento deseado estableciendo las configuraciones para ello; esta tarea de asignación es realizada por un Enrutador de Borde de Etiqueta (*Router Label Edge*, LER), donde a medida que el paquete va avanzando hacia el destino, esas etiquetas van cambiando y ese proceso lo realizan los Enrutadores de Conmutación de Etiqueta (*Router Label Switch*, LSR).
- QoS: Campo conformado por 3 bits que corresponden a la calidad del servicio.
- S: conformado por 1 bit, Este campo indica si la etiqueta actual es la última en una pila de etiquetas (1) o si hay más etiquetas que se deben procesar (0).
- TTL: Este campo indica el número de saltos que puede hacer un paquete enviado, con esto se busca evitar bucles infinitos en el enrutamiento, pues ese número de saltos va disminuyendo a medida que pasa por cada nodo y en caso de llegar a cero, el paquete se descarta.

1.2.1 *Label Stack* (Pila de Etiquetas)

MPLS tiene la capacidad de operar en múltiples niveles, esto lo hace agregando varias etiquetas una sobre la otra, a esto se le denomina pila de etiquetas, uno de sus objetivos es crear túneles dentro de otros túneles, y así contar con caminos alternativos para aumentar la disponibilidad en la red [10]. La figura 1.5 muestra un ejemplo de una pila de etiquetas.

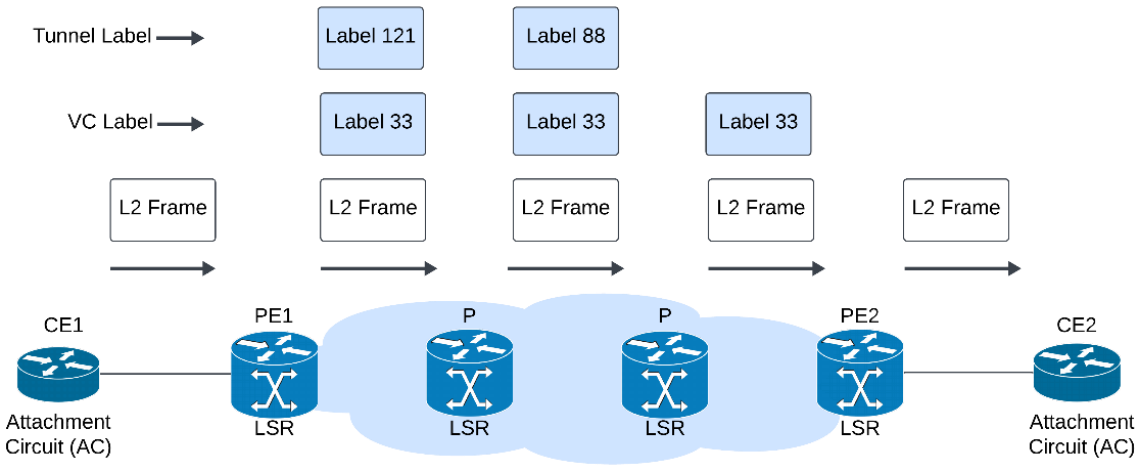


Figura 1.5 Ejemplo de Label Stack a través de la red [10]

1.2.2 Arquitectura de Red MPLS

Una red MPLS está conformada por varios componentes que trabajan en conjunto para permitir la conmutación de etiquetas y un enrutamiento eficiente, se define la arquitectura de red de MPLS en la figura 1.6.

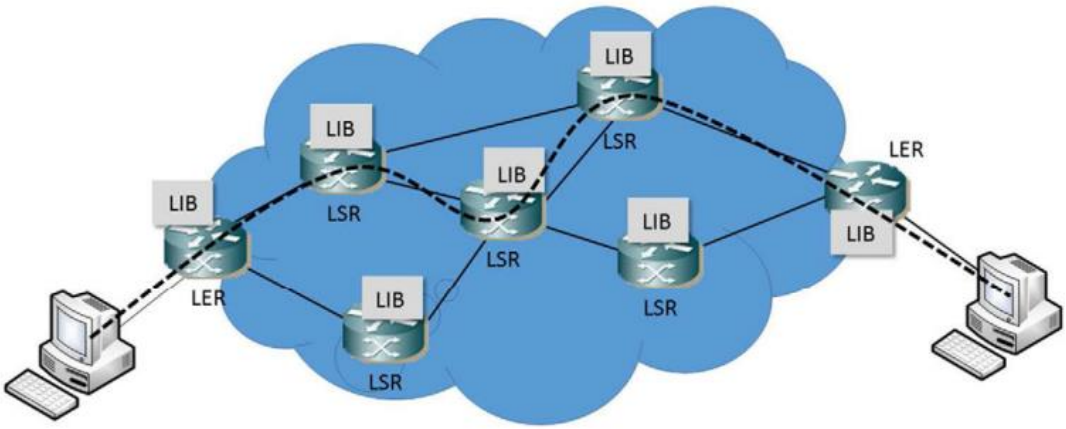


Figura 1.6 Arquitectura de una red MPLS [13]

Los componentes de la red MPLS son [10] [12] [13]:

- Etiquetas: Son identificadores numéricos que se utilizan para identificar rutas dentro de una red MPLS, y a su vez permite una conmutación ágil y eficiente.
- *Router Label Edge* (LER): Son los *Routers* que se ubican en los bordes de una red MPLS o también llamados *Routers* de frontera, encargados de convertir el tráfico entrante IP a MPLS, o en caso de salir de la red, se encargan de transformar tráfico MPLS a IP, teniendo en cuenta el destino, clasifican el paquete IP que entra a la red para asignar una etiqueta y establecer una ruta, dicha etiqueta va conmutando hasta llegar al LER de destino el cual elimina la etiqueta y encamina el paquete hacia su destino.
- *Router Label Switch* (LSR): *Routers* que están ubicados al interior del núcleo de la red MPLS, los cuales realizan la conmutación de etiquetas. Cuando un paquete entra por una interfaz del LSR, este extrae el valor de la etiqueta de entrada de la cabecera MPLS, busca en la tabla de conmutación la etiqueta de salida y por cual interfaz debe salir, reenvía el paquete por el camino preestablecido y agrega una nueva cabecera MPLS. También se encarga de extraer la cabecera MPLS al detectar si el próximo salto es un LER, este último LER al no realizar conmutación no necesita que llegue etiquetas a él, esto hace que se reduzcan las cabeceras innecesarias.
- *Label Information Base* (LIB): Es una tabla que contiene las etiquetas construidas en los LSR y los LER.
- *Label Forwarding Information Base* (LFIB): Relaciona cada etiqueta con una interfaz, ya sea de entrada como de salida, las cuales están almacenadas en la tabla LIB.
- *Label Switched Path (LSP):* Es el camino virtual establecido a través de la red MPLS por el cual el paquete hace su recorrido pasando por cada SLR hasta llegar al destino. Los LSP se pueden configurar para que el paquete pase por ciertos nodos o para que siga una ruta definida, ya sea por cuestiones de congestión en la red, para minimizar el número de salto o simplemente por especificar una ruta.
- *Label Distribution Protocol (LDP):* Es un protocolo que se utiliza para la distribución de etiquetas de los dispositivos que componen la red, ayudando a intercambiar etiquetas entre los *Routers* de una red MPLS.
- *Forwarding Equivalence Class* (FEC): es un grupo paquetes que son enviados sobre un mismo camino y comparten el mismo tratamiento sobre la red MPLS, sin importar el destino final.

En el establecimiento de las sesiones LDP primero se realiza el envio de mensajes *Hello* para el descubrimiento de vecinos usando el puerto del Protocolo de Control de Transmisión (*Transmission Control Protocol*, TCP) 646. Una vez se dectecta un vecino, se inicia el intercambio de información LDP con dicho vecino, como por ejemplo información de disponibilidad de etiquetas para su distribución, tiempo de sesión, capacidades que se pueden usar. Cuando se establece la conexión TCP, los vecinos LDP se envían mensajes de *keepalive*, para mantener la sesión en conexión, y permitir la transferencia de información de etiquetas entre los LSR, por medio de los mensajes *Label Request* y *Label Mapping* [13], como se observa en la figura 1.7.

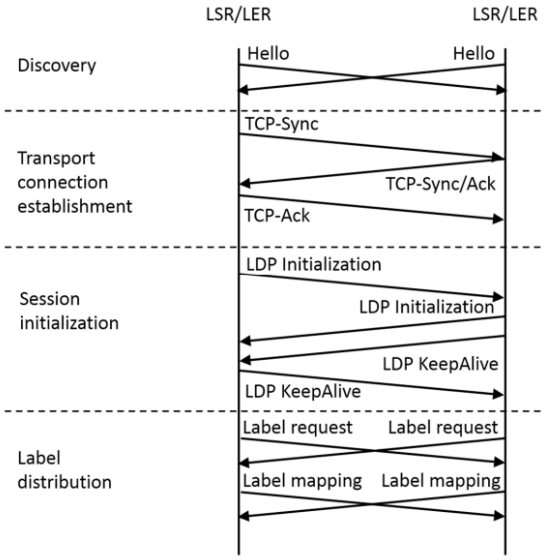


Figura 1.7 Establecimiento Sesión LDP [13]

1.2.3 Ventajas de MPLS

La tecnología MPLS brinda mayor seguridad en la implementación de Redes Privadas Virtuales (*Virtual Private Network*, VPN), Redes Enrutadas Privadas Virtuales (*Virtual Private Routed Network*, VPRN), y Servicios de LAN Privadas Virtuales (*Virtual Private LAN Services,* VPLS). MPLS proporciona una manera eficiente de establecer caminos virtuales dentro de una red, lo que la hace muy valiosa para redes de proveedores de servicios que necesitan administrar tráfico de múltiples clientes, además ofrece una mayor eficiencia de enrutamiento que al que se usa tradicionalmente, favorece la convergencia de servicios, pues en una sola infraestructura de red se puede tener servicios de voz, video y datos. Otra ventaja es que permite controlar los recursos priorizando el trafico y dando una mejor calidad en el servicio [12].

1.3 Red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de ISP

La red BGP/IPV6/VPN/MPLS (6*VPE over MPLS*) representa una combinación de tecnologías que brinda a sus usuarios eficiencia, seguridad, mejor desempeño en el enrutamiento, escalabilidad y funcionamiento en general de la red. Esta red se basa en cuatro componentes clave: el Protocolo de Conmutación de Etiquetas Multiprotocolo

(MPLS), las Redes Privadas Virtuales (VPN), el Protocolo de Pasarela de Borde (BGP) y el Protocolo de Internet versión 6 (IPv6). El Protocolo MPLS se encarga de introducir las etiquetas y de conmutarlas a medida que pasa por cada nodo de su red, esto permite la segmentación del tráfico y la optimización de rutas [14]. El Protocolo VPN crea un entorno privado y seguro dentro de una red pública, permitiendo la transmisión confidencial de datos [12]. El Protocolo BGP facilita el enrutamiento entre Sistemas Autónomos (*Autonomous System*, AS), asegurando la comunicación efectiva entre distintas redes [15]. IPv6 proporciona un protocolo de direccionamiento que resuelve el agotamiento de direcciones IPv4 y ofrece mejoras en la seguridad y enrutamiento [16].

Esta técnica es utilizada por los proveedores de servicios de red para facilitar la transición de las Redes de Área Amplia (*Wide Area Network*, WAN) hacia la conectividad IPv6 a través de VPNs, las cuales utilizan Enrutamiento Virtual y Reenvío (*Virtual Routing and Forwarding*, VRFs), permitiendo el uso de servicios tanto en IPv4 como en IPv6 sobre una infraestructura MPLS basada en IPv4 como se observa en la figura 1.8. Los equipos Enrutador Virtual de Borde de Proveedor (*Virtual Provider Edge,* VPE) desempeñan un papel esencial en la habilitación de esta funcionalidad, pues el enrutamiento de borde se realiza en dos direcciones [17]:

1. Entre los pares VPE se utiliza el Protocolo de Pasarela de Borde Interno (*Internal Border Gateway Protocol*, iBGP) para aprender las rutas tanto del Borde del Cliente (*Customer Edge*, CE), como entre los equipos VPE-CE.

2. Entre los pares VPE-CE se utilizan los Protocolos de Pasarela de Borde Externo (*External Border Gateway Protocol*, eBGP) y VRFs para compartir las rutas y establecer la comunicación con los clientes.

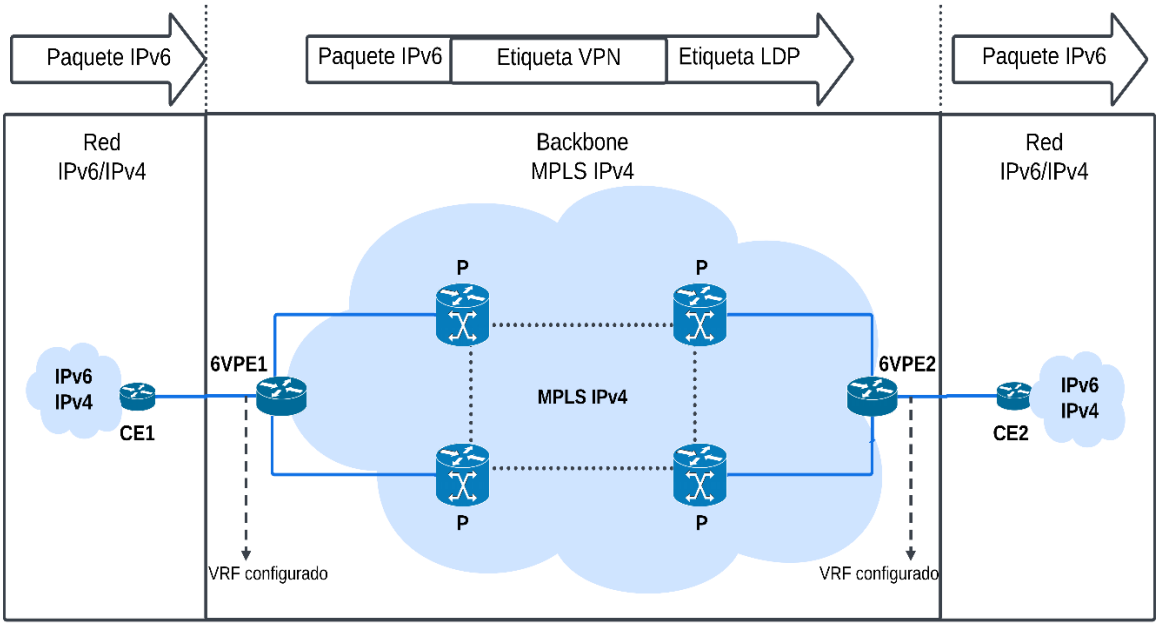


Figura 1.8 6VPE over MPLS [17]

Los dispositivos VPE son los *Routers* que están ubicados al borde de la red central y se conectan directamente con los *Routers* de los clientes. En estos dispositivos se implementan las VRFs, creando los túneles VPN que van a atravesar la red MPLS con

ayuda del protocolo LDP, que es el encargado de distribuir las etiquetas formando caminos LSP y así llegar al destino, según sea la cantidad de los clientes será la cantidad de las VRFs, para que este arreglo tenga un buen funcionamiento, es necesario tener en cuenta las capacidades y recursos de los dispositivos [17].

Para que un proveedor de servicios pueda implementar esta combinación, debe establecer una configuración de MPLS en su *CORE*, compuesto por los VPE y los *Router*s de Proveedor (*Provider*, P), por otra parte, se tienen los *Routers* CE que son externos al *backbone* y su contacto hacia el *CORE* se da por medio de los VPE [17].

La transición de IPv4 a IPv6 puede ser compleja, especialmente en entornos empresariales donde se utilizan VPNs para conectar sitios remotos de una manera segura y eficiente. Sin embargo, 6VPE resuelve este problema proporcionando una solución de enrutamiento que permite a las organizaciones utilizar sus redes VPNs existentes para transmitir tráfico IPv6. En lugar de actualizar por completo la infraestructura de red a IPv6, las organizaciones pueden mantener su infraestructura de IPv4 y utilizar 6VPE para enrutar el tráfico IPv6 a través de esta infraestructura existente [17].

6VPE utiliza la tecnología de enrutamiento MPLS, esto permite que los *Routers* de la red apliquen técnicas de enrutamiento eficientes y seguras a los paquetes IPv6, lo que simplifica la implementación de IPv6 en una Red Privada Virtual, donde en el núcleo no se revisa el paquete IPv6, sin embargo, realiza un analisis a las etiquetas para encaminar en paquete dentro de la red MPLS/IPv4 [18].

Los dispositivos P son los *Routers* que se encargan de hacer la conmutación de etiquetas asignadas por los VPE, la cual es una etiqueta externa, que indica cual es el VPE de salida, una vez el paquete llega al VPE de salida, este ultimo usa la etiqueta interna añadida por el VPE de entrada para indicar por cual interfaz debe salir el paquete para que llegar al destino requerido [19].

Los dispositivos CE, como se mencionó anteriormente, son los *Routers* de borde que pertenecen a la red del cliente, tienen conexión directa con los VPE y se encargan de distribuir las rutas hacia la red del cliente que fueron anunciadas por los VPE [19].

1.4 BGP

El Protocolo de Pasarela de Borde (*Border Gateway Protocol*, BGP) es el protocolo de enrutamiento fundamental en la arquitectura de Internet. Su función principal es facilitar el intercambio de información de enrutamiento entre diferentes Sistemas Autónomos, lo que permite a las redes comunicarse y enrutar tráfico a través de Internet [20].

BGP utiliza una serie de atributos para tomar decisiones de enrutamiento, como la longitud de Ruta de Sistema Autónomo (*Autonomous System Path*, AS-Path), la preferencia de ruta y otros criterios definidos. Uno de los atributos distintivos de BGP es el atributo AS-Path, que registra los AS que un paquete ha atravesado para llegar a su destino final. Esto permite a BGP evitar la formación de bucles de enrutamiento y seleccionar la ruta más eficiente. La comunicación se basa en una sesión TCP en el puerto 179 y debe mantenerse activa, ya que se actualiza periódicamente, para garantizar esa conectividad se envían mensajes periodicamente. En la fase inicial, cada *Router* comparte toda su información de enrutamiento con su vecino, y después solo se envían las nuevas rutas, actualizaciones o

eliminaciones de rutas previamente transmitidas. Si la conexión TCP se interrumpe, ambos extremos de la comunicación dejan de utilizar la información que obtuvieron del otro lado [20], la figura 1.9 muestra la estructura de paquete BGP.

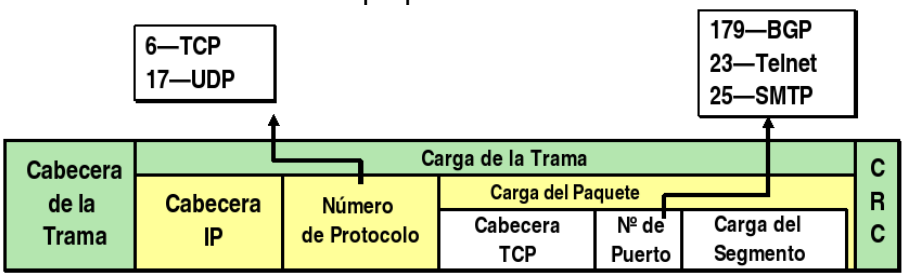


Figura 1.9 Estructura paquete BGP [21]

En BGP existen 4 tipos de mensajes que se utilizan para el intercambio de información y el establecimiento de conexiones entre *Routers* BGP, estos son [15]:

1. *Open*: Se utiliza para establecer una conexión BGP entre dos *Routers*.
2. *Keepalive*: Se utilizan para verificar que la conexión esté activa entre los *Routers* BGP. Se envían periódicamente para asegurarse de que la conexión TCP subyacente esté operativa.
3. *Update*: Se utilizan para anunciar cambios en las rutas o para informar sobre nuevas rutas.
4. *Notification*: Se envían cuando ocurre un error o cuando se termina una sesión BGP.

Cuando dos *Routers* están configurados con BGP, establecen una conexión TCP entre sí y se envían mensajes para iniciar y confirmar esta conexión. Estos *Routers* se conocen como "BGP peer *Router*" o "vecinos BGP". Una vez se establezca la conexión, los *Routers* intercambian sus tablas BGP completas. Debido a la confiabilidad de la conexión (gracias al uso del protocolo TCP), los *Routers* BGP solo necesitan actualizar los cambios en la red a partir de ese momento. A diferencia de otros protocolos que requieren mensajes de actualización periódicos, BGP envía mensajes "*Keepalive*" periódicamente para indicar que la conexión aún existe [15].

BGP mantiene una tabla que enumera a sus vecinos con los que ha establecido una conexión BGP. En esta tabla, el *Router* almacena toda la información que ha enviado y recibido a través del protocolo BGP, el propio *Router* es responsable de determinar la mejor ruta a través de la información en su tabla BGP y la proporciona a la dirección IP de destino. Además, es posible configurar el *Router* para que intercambie información de manera constante con sus vecinos BGP. Esto permite que el *Router* siempre esté al tanto de las mejores rutas disponibles y pueda tomar decisiones de enrutamiento más eficientes [16].

Una vez establecida la adyacencia, se hace un intercambio de rutas entre los vecinos BGP, cada *Router* almacena estas rutas en su tabla BGP, adquiriéndolas de los vecinos conectados, la ruta que un *Router* elige como la mejor opción debe cumplir con ciertos atributos, luego esta ruta se compara con las que el *Router* ya tiene en su propia tabla BGP. Después de un proceso de selección adicional, el *Router* decide qué ruta enviar a su tabla IP [20].

Este protocolo juega un papel crucial en la comunicación entre ASs en la red BGP/IPV6/VPN/MPLS. Su función esencial en el enrutamiento Inter dominio hace que sea necesario comprender su funcionamiento y cómo afecta el rendimiento de la red. La

implementación de los métodos *Route Reflector* y *Confederations BGP* en el entorno virtualizado se basa en la interacción y el comportamiento de BGP en una red de este tipo.

BGP es un protocolo de *Routing* y se utiliza como vector de ruta, creando la base para el intercambio de datos sobre la accesibilidad de los *Routers* disponibles y gestión de paquetes de datos. Es el protocolo principal de publicación de rutas más usado por las compañías más importantes de ISP para la comunicación entre diferentes ASs y dentro de ellos como se muestra en la figura 1.10, donde los ASs intercambian sus tablas de rutas, configurando y tomando decisiones basadas en políticas y reglas definidas por los administradores de la red para garantizar una mayor estabilidad en la comunicación interna BGP (iBGP) y externa BGP (eBGP) [20].

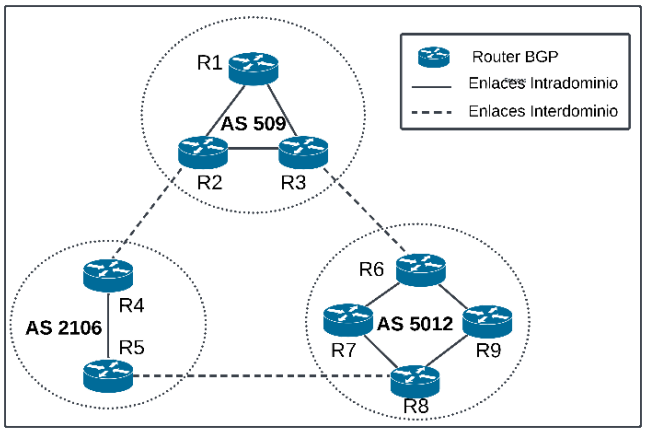


Figura 1.10 Red sencilla con varios Sistemas Autónomos [22]

El protocolo BGP posee sus propios atributos, los cuales tienen ciertas consideraciones para decidir que ruta resulta mejor para llegar a su destino [15] [20] [22]:

- *Origin*: Se encarga de informar a los AS como se introdujo el prefijo de red, es decir cómo se ha aprendido la ruta; se identifica con los siguientes valores:
  IGP: Indica que el prefijo fue originado en un IGP.
  EGP: Indica que el prefijo fue originado en un EGP.
  INCOMPLETE: Indica que el prefijo se ha aprendido de otra forma a las mencionadas anteriormente, como por ejemplo de manera estatica, o también puede ser producto de una redistribución incompleta de otro protocolo de ruteo.
- *Next-hop*: Indica la direccion IP de la interfaz del *Router* al cual dará el siguiente salto para alcanzar el destino.
- *As_Path:* Este atributo es una secuencia almacenada de AS que se deben atravesar para llegar al AS destino. Es un factor clave en el algoritmo de selección de rutas, pues es usado para indicar las distintas rutas que pueden llevar a un mismo destino, otra de sus grandes cualidades es prevenir bucles en los anuncios de las rutas, esto lo hace rechazando las rutas que se anuncien desde el mismo AS.
- *Multi-Exit Discriminator (MED):* Este atributo se usa en las sesiones eBGP. Indica a otro AS directamente conectado, la preferencia del trayecto del trafico de entrada, ya sea por dos o más enlaces, entre más bajo sea el valor del MED, más prioridad se le dará a dicha ruta.
- *Local Preference:* Este atributo se usa para indicar al AS cual es el punto de partida que tiene mayor preferencia para salir del AS que tiene distintas rutas hacia un destino de red determinado. Es usado en sesiones locales (iBGP), cuanto mayor sea el

número, mayor será la preferencia, por ende, BGP selecciona la ruta con el *Local Preference* de mayor valor.

- *Weight:* Este atributo es el principal, tiene la importancia mas alta sobre los otros atributos, siendo el primer criterio que es tomado en cuenta para elegir una ruta cuando se tienen diferentes caminos para llegar a un destino. En este caso la ruta con mayor valor será la elegida.

En una conexión iBGP se debe tener una topología de malla completa (*Full Mesh*) lo que obliga a mantener levantadas muchas sesiones TCP para transmitir la información de las rutas, causando una sobrecarga en la red. Para disminuir la cantidad de sesiones TCP, se propusieron dos arquitecturas muy utilizadas por los ISP: *Route Reflector* y *Confederations BGP* que minimizan los tiempos de convergencia [22].

1.4.1. Método *Route Reflector*

El método Reflexión de Rutas BGP (*Route Reflector*, RR) es definido por el IETF en la RFC 4456 [23]. Es un componente clave en redes que implementan el protocolo BGP, al ayudar a simplificar la administración de conexiones BGP en sistemas de redes complejas, eliminando la necesidad de una topología *Full Mesh.* Su función principal es reducir la carga de trabajo en los equipos VPE al minimizar la cantidad de sesiones BGP que deben mantener con otros VPE en la red. La figura 1.11 muestra como el *Router* RR refleja las rutas para el AS 15.138.

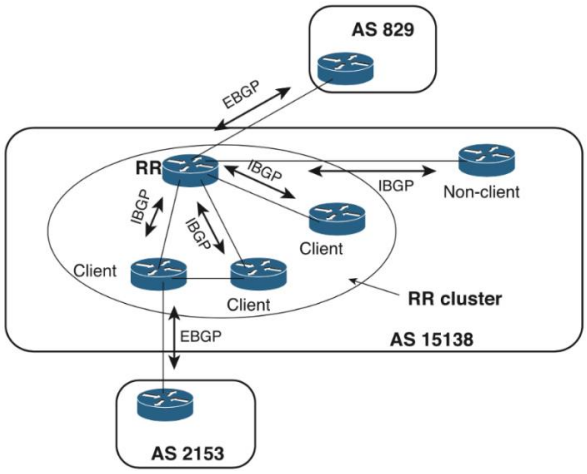


Figura 1.11 Método Route Reflector [24]

En lugar de establecer conexiones BGP con los demás VPE en la red, los VPE solo necesitan conectarse al Reflector de Rutas (*Route Reflector*, RR) y se convierten en Clientes del Reflector de Rutas (*Route Reflector Client, RRC)*. De esta manera, cada VPE mantiene un número constante de sesiones BGP, independientemente del número de VPE que se adicionen o elimen en la red.

Cuando un RR recibe un mensaje BGP, sigue las siguientes reglas [25]:

- Si el mensaje proviene de un vecino que no es un cliente, el RR refleja ese mensaje a todos sus clientes dentro de su *Cluster*, pero no a los que no son clientes del RR, como se observa en la figura 1.12.

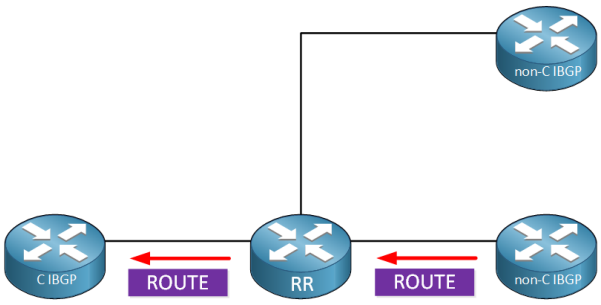


Figura 1.12 Regla 1 del método Route Reflector [25]

- Si el mensaje proviene de un cliente, el RR refleja el mensaje a todos los vecinos, tanto clientes como no clientes, como se observa en la figura 1.13.

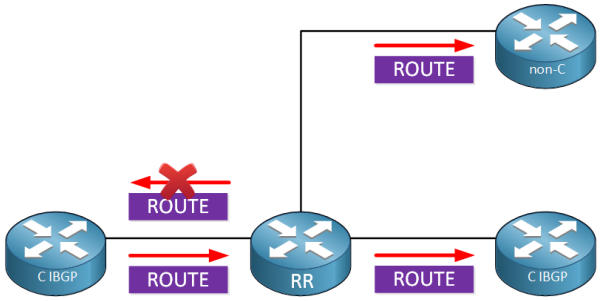


Figura 1.13 Regla 2 del método Route Reflector [25]

- Si el mensaje proviene de un vecino eBGP, el RR lo envía a todos los vecinos tanto clientes como no clients, como se observa en la figura 1.14.

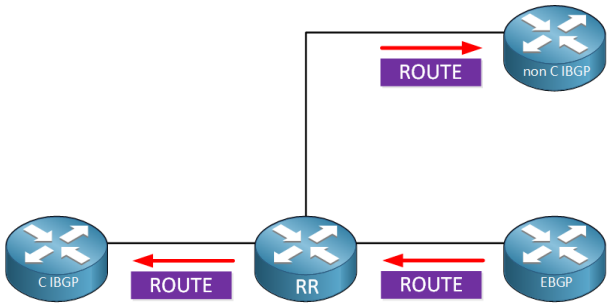


Figura 1.14 Regla 3 del método Route Reflector [25]

Una de las ventajas más notorias de esta configuración es que al adicionar un nuevo VPE a la red no requiere reconfigurar todas las conexiones BGP existentes. El RR permite agregar nuevos VPE sin afectar las configuraciones BGP existentes entre los demás VPE, simplificando enormemente la gestión de la red además de brindar escalabilidad [26].

Sin embargo, también existen algunas desventajas sobre los RR, pues estos tienen limitaciones que pueden afectar la eficiencia; al no realizar un buen diseño, como por ejemplo no dimensionar la capacidad de la red, falta de escalabilidad, reducción de opciones de encaminamiento y encaminamiento sub-óptimo. Puede perder robustez haciendolo vulnerable a fallos y pueden introducir retrasos en la convergencia debido a la necesidad de propagar y procesar actualizaciones de rutas a través de ellos [22].

1.4.2. Método *Confederations BGP*

Definido por el IETF en la RFC 5065 [27], es una solución para reducir una topología *Full Mesh*. Básicamente una confederación es un AS que se subdivide en subsistemas autónomos (subAS) más pequeños. Cada sub sistema se identifica de manera única dentro del AS mediante un número de subAS. Cada uno de estos subAS, se comporta como un AS independiente, manteniendo el requerimiento de malla completa de sesiones iBGP al interior de estos, es decir cada *Router* en ese subAS debe estar conectado a todos los demás *Routers* dentro del mismo subAS manteniendo conexiones BGP directas internamente para compartir rutas. Para conectarse con otros subAS se utiliza el protocolo BGP externo (eBGP), no enlaces iBGP a pesar de que estos pertenezcan a un AS global [28] como se muestra en la figura 1.15.

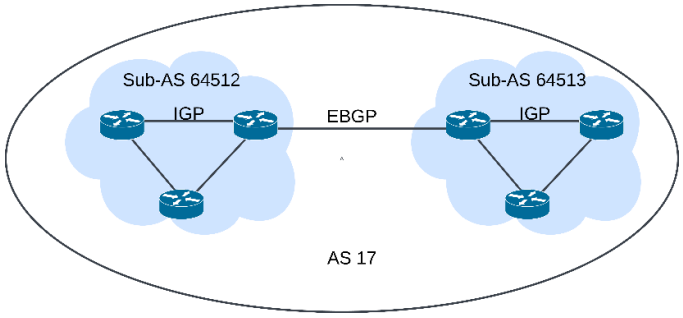


Figura 1.15 Método Confederations BGP [28]

Desde el punto de vista de otros AS en la misma confederación, el AS de la confederación parece ser un AS completo y único. Cuando las rutas se anuncian fuera del AS de la confederación, se eliminan los números de subAS asignados de forma privada. Esto significa que las rutas externas a la confederación solo ven el número de AS global asignado, sin conocer la estructura interna de subAS y secuencia de confederación [28].

1.5 TCP

El Protocolo de Control de Transmisión definido por el IETF en el RFC 9293 [29], es uno de los protocolos principales en la capa de transporte que tiene como objetivo principal proporcionar una comunicación orientada a la conexión, que garantice la entrega confiable y segura de datos entre dispositivos en redes IP [30]. En la figura 1.16 se puede observar la comunicación de dos procesos por TCP.

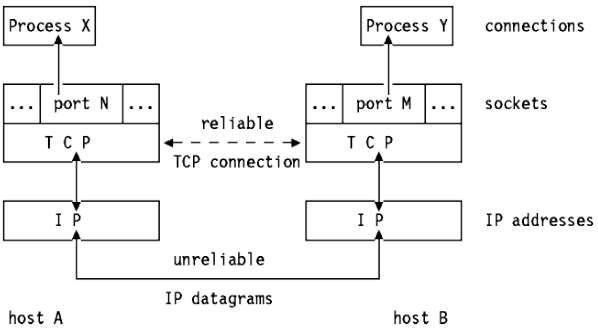


Figura 1.16 Conexión TCP entre procesos [31]

Desde la perspectiva de una aplicación, TCP se encarga de la transmisión fluida de datos a través de Internet. En este proceso, la aplicación no necesita preocuparse por dividir los datos en partes más pequeñas, como bloques o datagramas. En lugar de eso, TCP se encarga de organizar estos datos en lo que llamamos "segmentos TCP", que son entregados al módulo IP para su transmisión al destino. Una de las funciones principales de TCP es asegurar la integridad de los datos durante la transmisión. Para lograr esto, TCP implementa varios mecanismos, uno de ellos es asignar un número de secuencia único a cada byte de datos que se transmite, requiriendo que el receptor envíe un ACK (reconocimiento) para confirmar la recepción de los datos, si el emisor no recibe dicha respuesta durante cierto periodo de tiempo, retransmitirá los datos, garantizando que los datos lleguen a su destino. TCP puede utilizar estos números de secuencia para reorganizarlos en el orden correcto, incluso si llegan desordenados, además elimina los que llegan duplicados. Para abordar la posible corrupción de datos durante la transmisión, TCP agrega un campo de suma de control (*checksum*) a cada segmento de datos que se envía. El receptor calcula una suma de control similar y la compara con la recibida. Si no coinciden, se considera que los datos se han corrompido y se descartan [30] [31].

1.5.1. *Three-Way Handshake*

El protocolo de enlace de 3 vías es un procedimiento en las redes TCP/IP que establece una conexión entre un servidor y un cliente. Antes de que comience la comunicación de datos, tanto el cliente como el servidor deben llevar a cabo un proceso de intercambio de información para poder sincronizarse [32]. Este método de protocolo de enlace en tres pasos se utiliza para permitir que ambas partes que desean comunicarse establezcan y acuerden los parámetros de su conexión de red TCP simultáneamente antes de que comiencen a enviar datos. Si se ve desde un punto de vista general, es como si ambos extremos se dieran la mano antes de empezar a hablar, lo que facilita que múltiples conexiones de red TCP se configuren y utilicen en ambas direcciones al mismo tiempo [33].

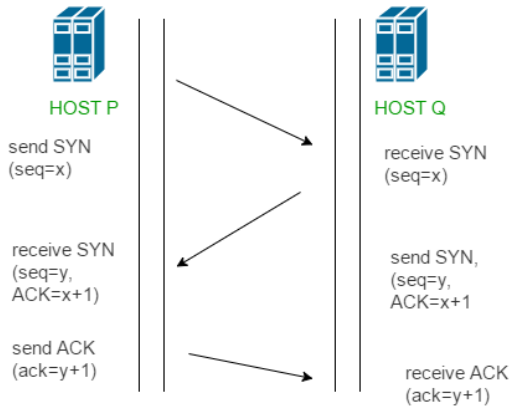


Figura 1.17 Three-Way Handshake [33]

En la figura 1.17 se observa que el cliente inicia el proceso enviando un paquete SYN (mensaje de sincronización) hacia el servidor, paquete que contiene un número de secuencia aleatorio. Al llegar al servidor, este devuelve un paquete SYN-ACK (*Synchronize-Acknowledgment*) con destino al cliente, esto para reconocer el número de secuencia del cliente y también informando que el servidor recibió la solicitud. Una vez el cliente recibe la confirmación por parte del servidor, envía un nuevo paquete ACK, donde reconoce el

número de secuencia del servidor y de esta manera se sincronizan ambos números de secuencia, estableciéndose una conexión formal [33].

1.5.2. *Round Trip Time* (RTT)

Se refiere al intervalo de tiempo medido en milisegundos (ms), que hay entre el envío de un paquete de datos por parte de un emisor, hasta obtener la respuesta de su confirmación por parte del receptor, como se observa en la figura 1.18. Este parámetro desempeña un papel fundamental en garantizar la confiabilidad de la entrega de datos en redes y protocolos de comunicación, pues cuando un administrador de red detecta que estos tiempos están fuera de lo común, se puede considerar que el paquete pudo ser desviado y sería una alerta para el administrador de la red, o simplemente puede ser indicador de que el paquete se perdió, generando el proceso de retransmisión teniendo en cuenta el tiempo de espera máximo a determinar para que se de esa retransmisión, además de detectar estas posibles fallas o ataques, permite evaluar la calidad de una conexión en una red, analizando los tiempos de repuesta [34].

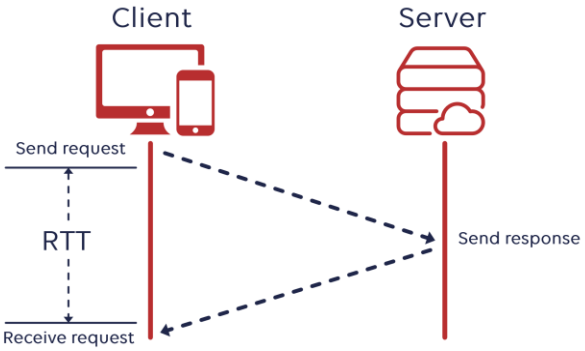


Figura 1.18 Round Trip Time [35]

El monitoreo continuo del RTT proporciona a un ISP valiosa información para comprender tanto la seguridad como el rendimiento del tráfico en su red [35].

1.5.3. *Frame Arrival Delay*

Se refiere a la diferencia en el tiempo que hay al enviar un paquete de datos entre el origen y la llegada de este a su destino. Es decir, al retraso experimentado por los paquetes de datos transmitidos a través de una red para llegar a su destino. Este retraso se produce debido a varios factores, incluidos los errores de transmisión, las retransmisiones de paquetes y la demora en la cola de transmisión en el búfer de la capa Control de Acceso al Medio, entre otros [36].

1.5.4. *Jitter*

El *Jitter* es una variación de las latencias de paquetes transmitidos de extremo a extremo en las redes, tiempo medido en milisegundos [37]. Esto se debe a que los paquetes experimentan retardos de puesta en cola variables en los *Routers* de la red. Debido a estas variaciones, el tiempo que toma un paquete en viajar desde su origen hasta su destino puede variar de un paquete a otro [38], como se observa en la figura 1.19.

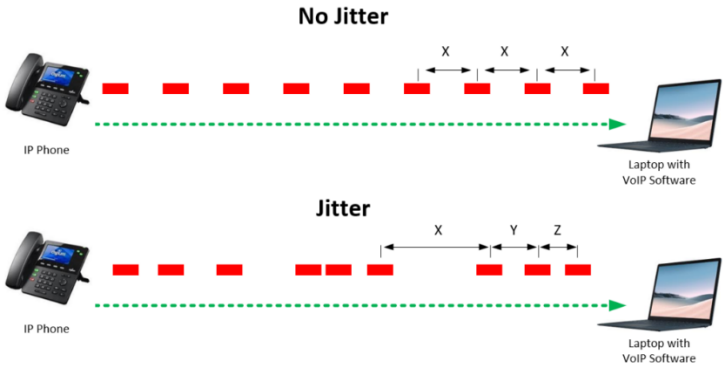


Figura 1.19 Jitter [38]

El *Jitter* es una medida estadística que cuantifica la variación promedio de las latencias en relación con la latencia media. No cabe duda de que supervisar métricas de conexión, como lo es el *Jitter*, resulta fundamental para obtener una mejor comprensión de la red, en cuanto detectar y gestionar puntos de falla o detectar dispositivos que no se comportan de la manera esperada cuando a rendimiento se refiere [38].

2. DISEÑO E IMPLEMENTACIÓN DE LA RED BGP/IPV6/VPN/MPLS (*6VPE OVER MPLS*) DE ISP EN UN ENTORNO DE PRUEBAS VIRTUALIZADO

En este capítulo se presenta la metodología para implementar una topología de red BGP/IPV6/VPN/MPLS de un Proveedor de Servicios de Internet en un entorno de pruebas virtualizado que simula una red real. Inicialmente se diseña la red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de ISP y se emula cada escenario de pruebas con los protocolos necesarios para el cumplimiento de los objetivos. En primera medida se realiza la simulación sin los métodos y posteriormente se aplican de forma independiente los métodos *Route Reflector*, *Confederations BGP* y la combinación de los mismos.

Para el proceso de implementación, se optó utilizar el modelo de desarrollo secuencial, el cual propone una estructura rígida que permite implementar y llevar seguimiento a un número determinado de actividades agrupadas en fases o ciclos [39]. De este modo, se menciona que, en este enfoque, cada una de las actividades que se describirán con posterioridad se llevaron a cabo de manera secuencial, es decir, una después de otra de acuerdo a las fases empleadas, como se muestra en la figura 2.1.

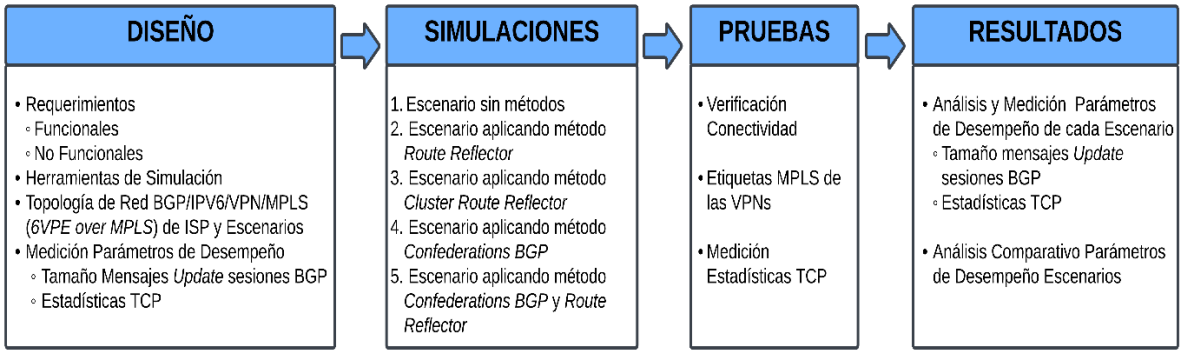


Figura 2.1 Fases de Desarrollo

2.1 Diseño

Se plantean los Requerimientos Funcionales y no Funcionales de la red BGP/IPv6/VPN/MPLS (*6VPE over MPLS*) de ISP, se analizan las diferentes herramientas de Simulación que permiten Virtualizar la Red, de las cuales se establece la más adecuada para este Trabajo de Grado. Se procede a diseñar la Topología de la red BGP/IPv6/VPN/MPLS (*6VPE over MPLS*) de ISP, que posteriormente se le aplicarán de forma independiente los métodos *Route Reflector*, *Confederations BGP* y la combinación de los mismos. Además, se definen los parámetros a medir en cada escenario para realizar la comparación del desempeño en los escenarios a los que se aplicaron métodos, con base al escenario sin métodos.

2.1.1 Requerimientos

Este trabajo, se centra en examinar estrategias para optimizar el rendimiento de redes de ISP. Los métodos *Route Reflector* y *Confederations BGP* serán aplicados para evaluar su impacto en cuanto al desempeño de la red. Para guiar este proceso, han sido definidos tanto los requerimientos funcionales, como los no funcionales, que establecen criterios para abordar las necesidades del sistema.

2.1.1.1 Requerimientos Funcionales

- Establecer en el núcleo de la red un solo Sistema Autónomo, con capacidad de soportar el protocolo MPLS utilizado para el etiquetado de paquetes, el direccionamiento de red debe ser IPv4 estático y aplicado sobre las *loopbacks* de los *Routers.*
- Implementar VRFs en los 6VPE del núcleo, permitiendo que la información enviada desde un cliente a otro, atraviese la red central mediante la implementación de Redes Privadas Virtuales, garantizando la seguridad y privacidad de los datos.
- Realizar enrutamiento estático IGP en el núcleo de la red, permitiendo la configuración de diversos escenarios de prueba.
- Configurar direccionamiento IPv6 en los clientes para abordar la carencia de direccionamiento.
- Permitir que la topología de red se adapte a la configuración de los diferentes escenarios de prueba, posibilitando la simulación y análisis efectivo de cada uno.
- Realizar enrutamiento eBGP entre los Sistemas Autónomos de los clientes y el sistema Autónomo del Proveedor de Servicios, asegurando la conectividad y la correcta interacción entre entidades autónomas.

2.1.1.2 Requerimientos no Funcionales

- Permitir compatibilidad con diferentes herramientas de simulación y rastreo, para permitir libre elección por parte del usuario.
- Facilitar la interoperabilidad del sistema permitiendo la conexión con dispositivos de distintas referencias o marcas, siempre y cuando se cumpla con las especificaciones técnicas requeridas.
- La red debe ser escalable, permitiendo la incorporación de nuevos dispositivos sin comprometer el rendimiento general del sistema.

2.1.2 Herramientas de Simulación

Los *Software* de Simulación y Emulación que permiten la virtualización de redes, más populares y robustos son, *Cisco Packet Tracer*, EVE-NG y GNS3, como se observa en la figura 2.2. De los cuales se opta por el Emulador Gráfico GNS3 que es un *Software* gratuito y de código abierto que permite emular varios dispositivos de red, los cuales facilitan la emulación de una red con diferentes características, al trabajar directamente con Imágenes de Sistema Operativo de Interconexión (*Internetwork Operating System*, IOS) de dispositivos reales, como lo son: *Routers*, *Switchs*, *Firewalls* y *Hosts*. Estos dispositivos deben soportar los protocolos IPv6, BGP, VPN y MPLS. Tiene todas las funciones de los dispositivos físicos al emularlos, comportándose lo más parecido a una red real, en comparación con el *Software Cisco Packet Tracer* que sólo simula los dispositivos. Por otra parte el emulador EVE-NG es una herramienta que brinda caracteristicas similares a GNS3,

sin embargo, se requiere de licencia para poder contar con gran cantidad de servicios, pues la versión gratuita es muy limitada, por ejemplo para poder integrar la herramienta con *Wireshark* se necesita la version de pago [40].

Figura 2.2 Herramientas de simulación [40]

El *Software* de emulación GNS3 permite emular *Routers* físicos por medio IOS, en un entorno virtual con un Servidor Remoto que agiliza el proceso de Virtualización. Se utiliza un equipo con procesador *Intel Core* i5 7200U, 16Gb de Memoria de Acceso Aleatorio (*Random Access Memory,* RAM) y Unidad de Estado Sólido (*Solid State Drive,* SSD) de 512Gb, cuyo *hardware* soporta la virtualización para realizar las simulaciones de todos los escenarios.

GNS3 ha sido optimizado para que trabaje de una forma liviana y eficiente con el Servidor en una máquina virtual (*GNS3 VM SERVER*), por lo tanto se procede a descargar el instalador desde la página oficial de GNS3 [41], donde se debe registrar para obtener acceso a la descarga del servidor *GNS3 VM y* se instala sobre el *Software* de Virtualización *VM Workstation Pro 16.* Se debe habilitar la tecnología de Virtualización que permite que varias cargas de trabajo compartan un conjunto común de recursos (*Hardware*), por lo que se habilita desde la configuración de la *Bios* del equipo, así como en el *Software* de Virtualización *VmWare*. Se muestra el *GNS3 VM Server* en la figura 2.3.

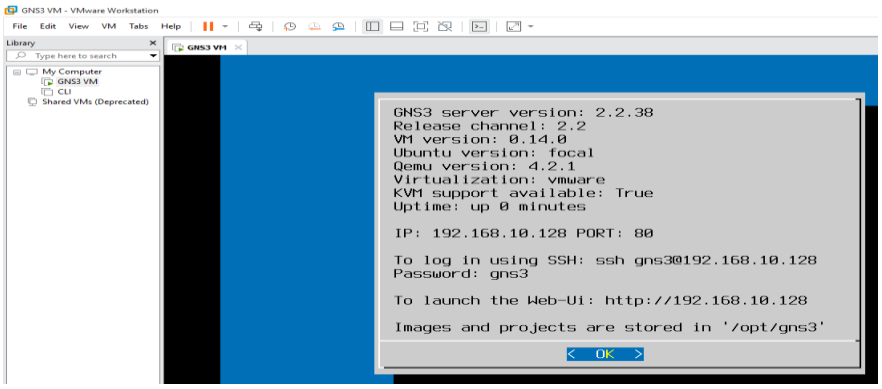


Figura 2.3 Server GNS3 VM en VM Workstation Pro 16

Desde el menú de GNS3 GUI de Windows, se accede a la pestaña *Edit,* se ingresa a *Preferences,* y se selecciona *Server,* para configurar la dirección IP 192.168.10.1 con la cuál se accede al Servidor en el campo seleccionable llamado *Host bindin.* En la opción de *GNS3 VM* se activa el Servidor y se configura como se muestra en la figura 2.4.

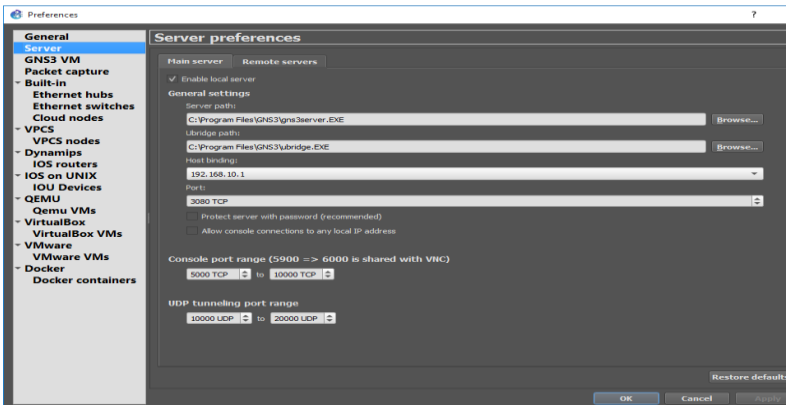


Figura 2.4 Configuración GNS3 VM SERVER

Se revisa en la interfaz Gráfica GUI de GNS3 que se haya conectado satisfactoriamente con el *Server*, en *Servers Summary* (parte inferior derecha) activándose en color verde la opción de *GNS3 VM*. Ahora se deben empezar a cargar las respectivas IOS de los *Routers* que se van a utilizar, utilizando imágenes de equipos Cisco. Se emulan las imágenes por medio de *Dynamips* que es el emulador de IOS que ejecuta imágenes de *Routers,* y *Dynagen* un front-end que se encarga de editar los archivos de texto. Se descarga la imagen del *Router* 7200 de Cisco y se accede en el menú principal en *Edit, Preferences, Dynamips,* IOS *Router*, para cargar la imagen. Este es a uno de los *Router* que se puede acceder a su IOS de manera gratuita y se pueden configurar los protocolos IPv6, BGP, VPN y MPLS que se necesitan para este proyecto; además de utilizar sus interfaces *Gigabit Ethernet* para realizar las diferentes conexiones. Al finalizar la carga de la imagen, aparece en las herramientas de simulación de los dispositivos que se tienen disponibles, como se observa en la figura 2.5.

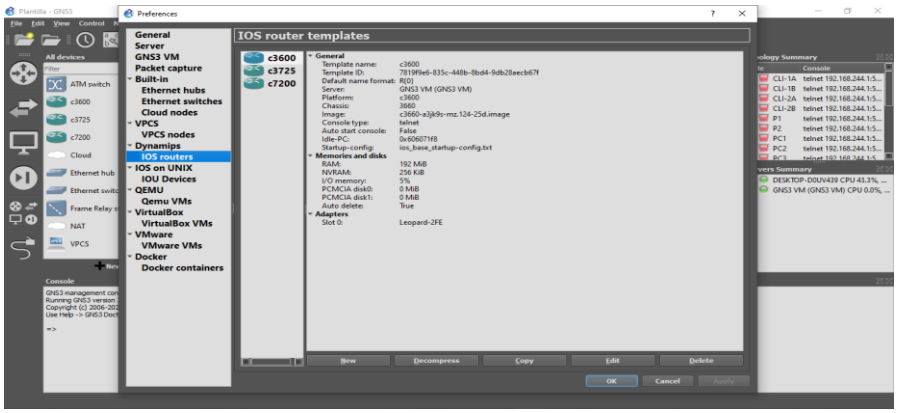


Figura 2.5 Conexión GNS3 VM SERVER y Carga de IOS

2.1.3 Topología de red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de ISP y Escenarios

Para este propósito se consideran los elementos esenciales que componen la red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de un ISP, que son los *Routers Provider* (P), *6VPN Provider Edge* (6VPE) y *Customer Edge* (CE). El *Backbone* está implementado sobre MPLS/IPv4 que es la arquitectura existente y está compuesto por los *Routers* P, 6VPE y a nivel de los clientes los *Routers* CE.

En la figura 2.6 se muestra el esquema de la Topología de la red BGP/IPv6/VPN/MPLS (*6VPE over MPLS*), donde se interconectan los Sistemas Autónomos de los clientes IPv6 con VPNs, a través del *CORE* de Proveedor con infraestructura existente MPLS/IPv4.

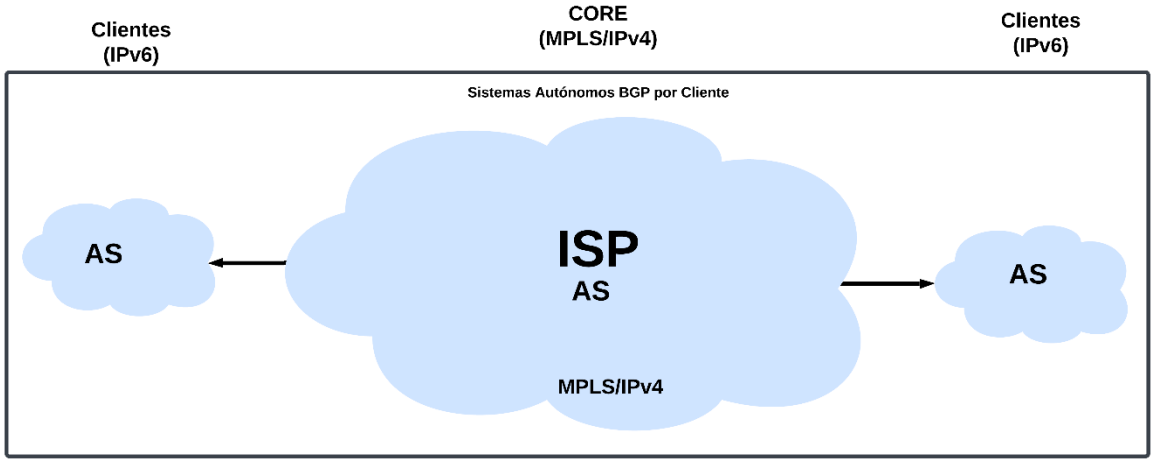


Figura 2.6 Esquema Topología de red BGP/IPV6/VPN/MPLS (6VPE over MPLS) ISP

En la topología base definida, se debe configurar inicialmente el Protocolo iBGP, del AS 65100 del ISP sin métodos en conexión *Full Mesh,* dando paso al escenario 1. Posteriormente a la topología base definida, se aplican individualmente los métodos *Route Reflector*, *Confederations BGP* y la combinación de los mismos, dando paso a los diferentes escenarios, como muestra la figura 2.7.

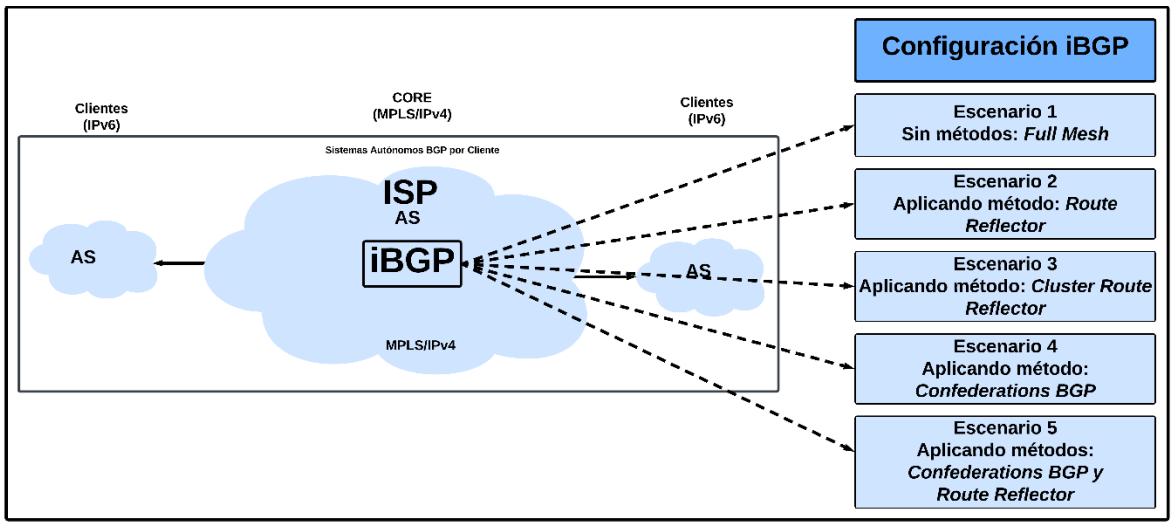


Figura 2.7 Escenarios de Simulación

2.1.4 Medición Parámetros de Desempeño

Se establecen los parámetros de medición en los Escenarios, los cuales son los parámetros de desempeño de red, tales como el tamaño de los mensajes *Update* de las sesiones BGP, y de las estadísticas TCP como el *Round Trip Time*, *Frame Arrival Delay* y *Jitter*. Estos valores deben ser medidos en todos los Escenarios, y comparados con base al Escenario 1 donde no se aplicaron métodos, como se observa en la figura 2.8.

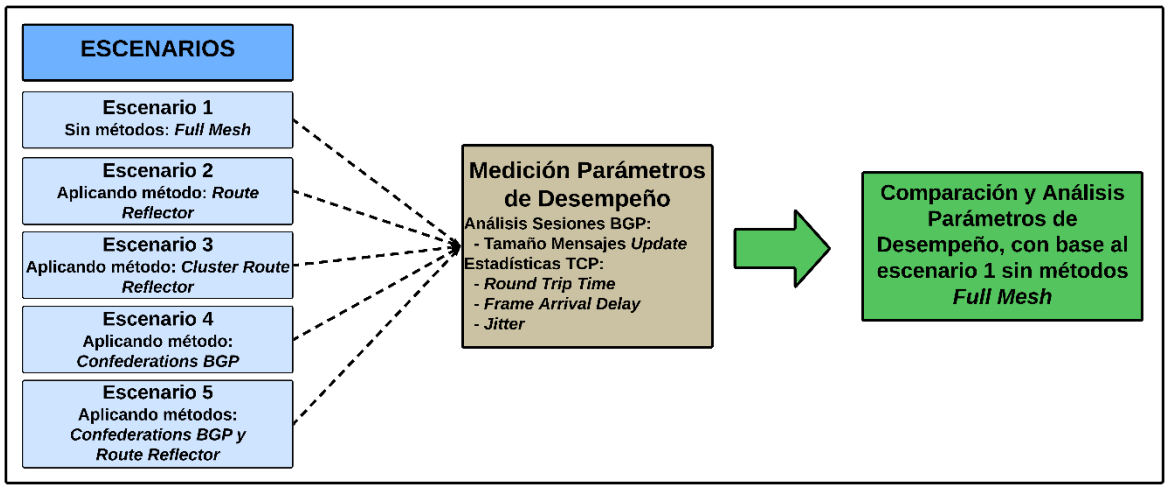


Figura 2.8 Medición Parámetros de Desempeño

2.2 Simulaciones

En este Trabajo de Grado, se diseña una red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de ISP como se observa en la figura 2.9.

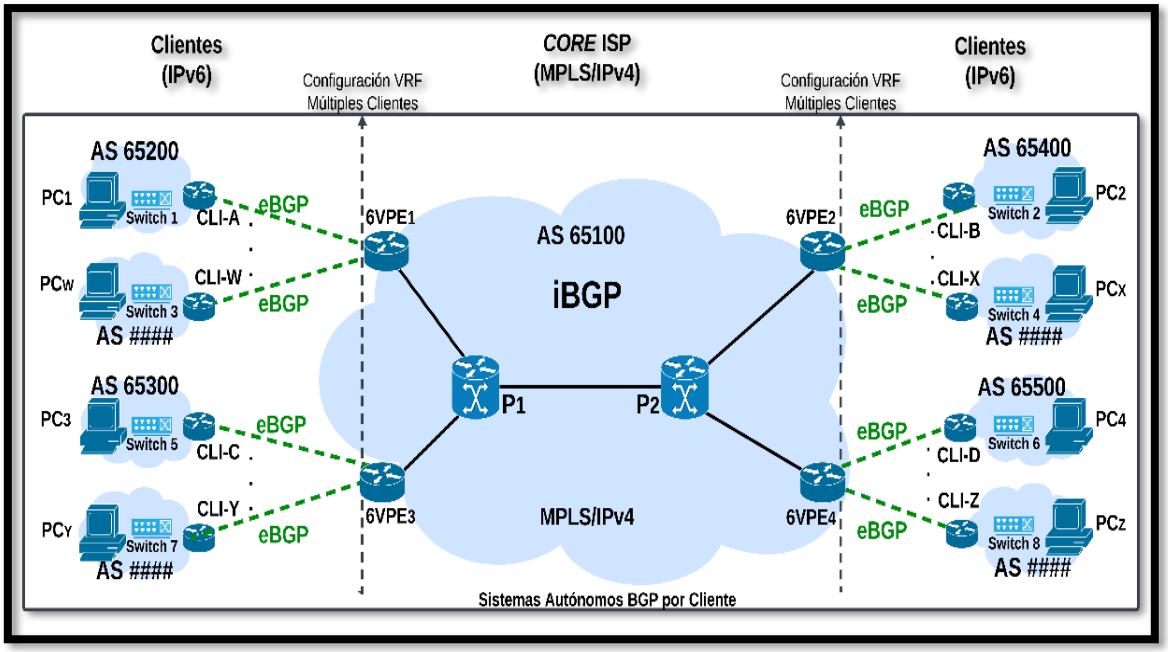


Figura 2.9 Red BGP/IPV6/VPN/MPLS (6VPE over MPLS) de ISP

Los elementos que componen el *CORE* son los *Routers* Provider (P) y los *Routers* 6VPE que son los *Routers* de Borde del Proveedor donde se crean las VPNs de los clientes en IPv6. Los *Routers* CE son los *Routers* de Borde de los Clientes que manejan Redes IPv6 y permiten el acceso al ISP, *Switchs* que permiten la interconexión de varios equipos en redes locales y los equipos de usuario finales *Hosts*, como se observa en la tabla 2.1.

Tabla 2.1 Elementos de Red

ELEMENTOS DE RED	NOMBRE
***Routers* de Borde de Cliente (CE)**	- CLI-A - CLI-B - CLI-C - CLI-D - CLI-W - CLI-X - CLI-Y - CLI-Z
***Routers* Virtuales de Borde de Proveedor (6VPE)**	- $6VPE_1$ - $6VPE_2$ - $6VPE_3$ - $6VPE_4$
***Routers* de Proveedor (P)**	- P_1 - P_2
Switchs	- *Switch*$_1$ - *Switch*$_2$ - *Switch*$_3$ - *Switch*$_4$ - *Switch*$_5$ - *Switch*$_6$ - *Switch*$_7$ - *Switch*$_8$
Hosts	- *PC-1A* - *PC-2A* - *PC-3* - *PC-4* - *PC-W* - *PC-X* - *PC-Y* - *PC-Z*

Se implementan VPNs para los clientes IPv6 en cada uno de los *Router* 6VPE utilizando las tablas VRF, que crean múltiples instancias de enrutamiento en un mismo *Router*, posibilitando que los clientes operen de manera simultánea y lógicamente independientes. Estos *Routers* hacen uso del Multi Protocolo BGP (*Multiprotocol Border Gateway Protocol*, MP-BGP) para propagar la información de las rutas de las VPNs, así como de las etiquetas MPLS dentro de la red. Cuando a un *Router* 6VPE llega un paquete de un cliente directamente conectado, busca en la tabla VRF encontrando una ruta VPNv6 hacia el destino que tiene una etiqueta MPLS asociada.

En esta Arquitectura de red los *Routers* 6VPE tienen contacto directo con la red de los clientes donde se crean las VPNs que agregan seguridad a la información transmitida. Tanto los *Routers* P como los 6VPE, trabajan en el proceso de conmutación de etiquetas con las cuales se construyen caminos LSP. La distribución de estas etiquetas MPLS se lleva a cabo mediante el protocolo LDP, el cual funciona a través del puerto 646 TCP. Cuando un *Router* 6VPE envía una dirección VPNv6 a través de la red MPLS, asigna dos etiquetas al tráfico del cliente, una etiqueta interna para identificación de las VPNs y otra etiqueta externa, que identifican los *Routers* P y van cambiando a medida que avanza y

llega hasta el penúltimo *Router* del *CORE*, es decir antes de llegar al *Router* 6VPE de destino. Cuando el paquete llega al penúltimo dispositivo antes de salir de la red central, el *Router* extrae la etiqueta externa y envía el paquete hacia el *Router* 6VPE de destino solo con la etiqueta interna, que determina la interfaz de salida de la VPNv6 al que el paquete debe ser direccionado para llegar a su destino. El *Router* 6VPE debe identificar que el paquete IP que ingresa pertenezca a la VRF, haciendo una búsqueda sobre la tabla de rutas IP asociada a la VRF. Para garantizar que los prefijos sean únicos, se asigna un Identificador de Rutas (*Route Distinguisher*, RD), creando la familia de direcciones para VPNv6 dentro de BGP.

Se configura MPLS/IPv4 con Interfaces Lógicas Internas de *Router* (*Loopbacks)* en el *CORE*, desde 1.1.1.1 hasta 6.6.6.6, donde se establece Enrutamiento Estático en IPv4 con el comando *ip route* como Protocolo de Pasarela Interior (*Interior Gateway Protocol*, IGP). Se asignan direcciones para los clientes VPNv6 asignando desde la red IPv6 2001:448:1024::1:0/112 hasta la red 2001:448:1024::13:0/112. Se muestra la configuración del direccionamiento de las interfaces en la tabla 2.2.

Tabla 2.2 Configuración de Direccionamiento

Router	**Interfaz**	**Dirección**
$6VPE_1$	Loopback 0	1.1.1.1
$6VPE_2$	Loopback 0	2.2.2.2
$6VPE_3$	Loopback 0	3.3.3.3
$6VPE_4$	Loopback 0	4.4.4.4
P_1	Loopback 0	5.5.5.5
P_2	Loopback 0	6.6.6.6
Cliente-A	Loopback 0	7::7
Cliente-B	Loopback 0	8::8
Cliente-C	Loopback 0	9::9
Cliente-D	Loopback 0	10::10
Cliente-A	g0/0	2001:448:1024::1:1
Cliente-A	g1/0	2001:448:1024::2:2
Cliente-B	g0/0	2001:448:1024::10:1
Cliente-B	g1/0	2001:448:1024::9:2
Cliente-C	g0/0	2001:448:1024::6:1
Cliente-C	g1/0	2001:448:1024::5:2
Cliente-D	g0/0	2001:448:1024::13:1
Cliente-D	g1/0	2001:448:1024::12:2
$6VPE_1$	g1/0	2001:448:1024::2:1
$6VPE_1$	g0/0	10.10.3.2
$6VPE_2$	g1/0	2001:448:1024::9:1
$6VPE_2$	g0/0	10.10.8.2
$6VPE_3$	g1/0	2001:448:1024::5:1
$6VPE_3$	g0/0	10.10.4.2
$6VPE_4$	g1/0	2001:448:1024::12:1
$6VPE_4$	g0/0	10.10.11.2
P_1	g0/0	10.10.3.1
P_1	g1/0	10.10.7.1
P_1	g2/0	10.10.4.1
P_2	g0/0	10.10.8.1

P_2	g1/0	10.10.7.2
P_2	g2/0	10.10.11.1

Se simula en el *Software* GNS3 una red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de un ISP, configurando primeramente la sesión iBGP *Full Mesh* sin aplicación de métodos y posteriormente aplicando independientemente los métodos *Route Reflector*, *Confederations BGP* y la combinación de ellos.

2.2.1 Escenario 1. Topología de Red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*): Sin aplicación de métodos – Conexión *Full Mesh*

Se configura iBGP en la red de *CORE,* levantando una conexión *Full Mesh* de sesiones virtuales iBGP (6 sesiones iBGP) entre los diferentes *Routers* 6VPE (6VPE$_1$, 6VPE$_2$, 6VPE$_3$ y 6VPE$_4$), la comunicación entre los *Routers* P y los *Routers* 6VPE se realiza internamente por MPLS/IPv4 asignando etiquetas, tal como se observa en la topología de la figura 2.10.

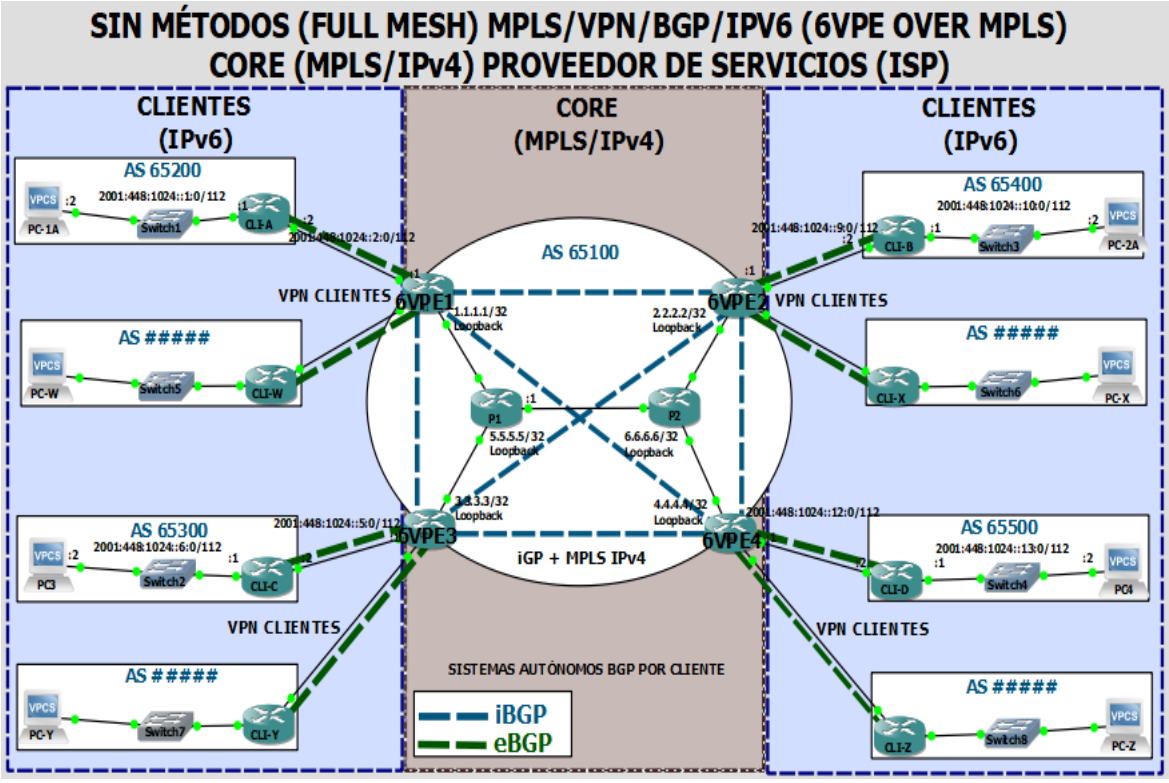


Figura 2.10 Escenario 1 - Full Mesh. Topología BGP/IPV6/VPN/MPLS

Se establecen VPNs en los *Routers* 6VPE, donde cada VPN tiene un único RD que junto a la Tarjeta de Ruta (*Route Target, RT)* permiten identificar las VPNs. El RD viene dado por el número de AS y seguido por el ID del cliente. Se crean las VRFs para las VPNs de los clientes en los diferentes *Routers* 6VPE, para efectos prácticos de conectividad y pruebas se permite el acceso de todas las VPNs en todos los *Routers* 6VPE, como se muestra en la tabla 2.3.

Tabla 2.3 Escenario 1. Creación de VRFs

6VPE$_1$	**6VPE$_2$**
conf t *vrf definition VRF1* *rd 65200:1* *address-family ipv6* *route-target export 65200:1* *route-target import 65300:1* *route-target import 65400:1* *route-target import 65500:1* *exit-add* *interf g1/0* *vrf forwarding VRF1* *ipv6 address 2001:448:1024::2:1/112* *no sh* *exit*	*conf t* *vrf definition VRF2* *rd 65400:1* *address-family ipv6* *route-target export 65400:1* *route-target import 65200:1* *route-target import 65300:1* *route-target import 65500:1* *exit-add* *interf g1/0* *vrf forwarding VRF2* *ipv6 address 2001:448:1024::9:1/112* *no sh* *exit*
6VPE$_3$	**6VPE$_4$**
conf t *vrf definition VRF3* *rd 65300:1* *address-family ipv6* *route-target export 65300:1* *route-target import 65200:1* *route-target import 65400:1* *route-target import 65500:1* *exit-add* *interf g1/0* *vrf forwarding VRF3* *ipv6 address 2001:448:1024::5:1/112*	*conf t* *vrf definition VRF4* *rd 65500:1* *address-family ipv6* *route-target export 65500:1* *route-target import 65200:1* *route-target import 65300:1* *route-target import 65400:1* *exit-add* *interf g1/0* *vrf forwarding VRF4* *ipv6 address 2001:448:1024::12:1/112*

Las VRFs hacen uso de direcciones IPv6 que van unidas al RD de las VPNs, estas son propagadas por iBGP desde los *Routers* 6VPE hacia los otros *Routers* 6VPE que contienen las VPNs, siendo el RT quien permite identificar rutas entrantes y salientes. La conexión entre clientes y 6VPE se realiza por medio de eBGP, porque pertenecen a diferentes AS.

Dentro del *Router* BGP hay varias familias de direcciones, por lo cual dentro de *address-family ipv6* en *route-target* se coloca el RD que se quiere exportar, que vendría siendo de la VPNv6 que tiene directamente conectada el 6VPE y el RD del *Peer* VPNv6 que se importa. Dentro del proceso BGP, en *address-family vpnv6* se activa el vecino 6VPE que contiene la VPN del cliente con el cual se envía y se recibe información; y en *address-family ipv6 vrf* VRF es donde se activa la interfaz de la VPN del cliente directamente conectado.

Las sesiones MPLS se realizan entre las interfaces que conectan directamente los *Router* P y 6VPE con LDP a través de las direcciones de *Loopback* 6VPE$_1$ (L0: 1.1.1.1), 6VPE$_2$ (L0: 2.2.2.2), 6VPE$_3$ (L0: 3.3.3.3), 6VPE$_4$ (L0: 4.4.4.4), P$_1$ (L0: 5.5.5.5), P$_2$ (L0: 6.6.6.6), activándose los *neighbor* LDP. Se habilita MPLS por LDP en las interfaces directamente conectadas entre los *Routers* P$_1$, P$_2$ y los 6VPE con los comandos *mpls ip y mpls label protocol ldp.*

Las sesiones iBGP *Full Mesh* entre los 4 *Routers* 6VPE se realiza por medio de sus direcciones *Loopback* 0, $6VPE_1$ (L0: 1.1.1.1), $6VPE_2$ (L0: 2.2.2.2), $6VPE_3$ (L0: 3.3.3.3), $6VPE_4$ (L0: 4.4.4.4), tal como se muestra en la tabla 2.4.

Tabla 2.4 Escenario 1. Sesiones BGP

$6VPE_1$	**$6VPE_2$**
router bgp 65100 *bgp router-id 1.1.1.1* *no bgp default ipv4-unicast* *neighbor 2.2.2.2 remote-as 65100* *neighbor 2.2.2.2 update-source Loopback0* *neighbor 3.3.3.3 remote-as 65100* *neighbor 3.3.3.3 update-source Loopback0* *neighbor 4.4.4.4 remote-as 65100* *neighbor 4.4.4.4 update-source Loopback0* *neighbor 2001:448:1024::2:2 remote-as 65200*	*router bgp 65100* *bgp router-id 2.2.2.2* *no bgp default ipv4-unicast* *neighbor 1.1.1.1 remote-as 65100* *neighbor 1.1.1.1 update-source Loopback0* *neighbor 3.3.3.3 remote-as 65100* *neighbor 3.3.3.3 update-source Loopback0* *neighbor 4.4.4.4 remote-as 65100* *neighbor 4.4.4.4 update-source Loopback0* *neighbor 2001:448:1024::9:2 remote-as 65400*
address-family vpnv6 *neig 2.2.2.2 activate* *neig 2.2.2.2 send community both* *neig 3.3.3.3 activate* *neig 3.3.3.3 send community both* *neig 4.4.4.4 activate* *neig 4.4.4.4 send community both* *exit-add*	*address-family vpnv6* *neig 1.1.1.1 activate* *neig 1.1.1.1 send community both* *neig 3.3.3.3 activate* *neig 3.3.3.3 send community both* *neig 4.4.4.4 activate* *neig 4.4.4.4 send community both* *exit-add*
address-family ipv6 vrf VRF1 *neig 2001:448:1024::2:2 activate* *redistribute connected*	*address-family ipv6 vrf VRF2* *neig 2001:448:1024::9:2 activate* *redistribute connected*
$6VPE_3$	**$6VPE_4$**
router bgp 65100 *bgp router-id 3.3.3.3* *no bgp default ipv4-unicast* *neighbor 1.1.1.1 remote-as 65100* *neighbor 1.1.1.1 update-source Loopback0* *neighbor 2.2.2.2 remote-as 65100* *neighbor 2.2.2.2 update-source Loopback0* *neighbor 4.4.4.4 remote-as 65100* *neighbor 4.4.4.4 update-source Loopback0* *neighbor 2001:448:1024::5:2 remote-as 65300*	*router bgp 65100* *bgp router-id 4.4.4.4* *no bgp default ipv4-unicast* *neighbor 1.1.1.1 remote-as 65100* *neighbor 1.1.1.1 update-source Loopback0* *neighbor 2.2.2.2 remote-as 65100* *neighbor 2.2.2.2 update-source Loopback0* *neighbor 3.3.3.3 remote-as 65100* *neighbor 3.3.3.3 update-source Loopback0* *neighbor 2001:448:1024::12:2 remote-as 65500*
address-family vpnv6 *neig 1.1.1.1 activate* *neig 1.1.1.1 activate send community both* *neig 2.2.2.2 activate* *neig 2.2.2.2 activate send community both* *neig 4.4.4.4 activate* *neig 4.4.4.4 activate send community both*	*address-family vpnv6* *neig 1.1.1.1 activate* *neig 1.1.1.1 send community both* *neig 2.2.2.2 activate* *neig 2.2.2.2 send community both* *neig 3.3.3.3 activate* *neig 3.3.3.3 send community both*
address-family ipv6 vrf VRF3 *neig 2001:448:1024::5:2 activate* *redistribute connected*	*address-family ipv6 vrf VRF4* *neig 2001:448:1024::12:2 activate* *redistribute connected*

Cuando se activan estos *Neighbor* en el proceso iBGP dentro de *address-family vpnv6*, se crea la conexión lógica directa entre todos los *Routers* 6VPE, pero internamente el *CORE* se estará comunicando por medio de etiquetas entre las interfaces de los *Routers* P y 6VPE directamente conectados.

2.2.2 Escenario 2. Topología de Red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*): Con aplicación de método *Route Reflector*

Para evitar levantar tantas sesiones iBGP se procede aplicar el método *Route Reflector* a la topología de la figura 2.9 con el direccionamiento de la tabla 2.2, cambiando solamente su configuración iBGP. Se convierte el *Router* P_1 en el *Router Reflector 1* (RR_1) y los demás *Routers* en sus clientes; es decir que las sesiones iBGP de los *Routers* 6VPE se realiza con la *Loopback0* del *Router Reflector 1* (RR_1), quedando sus sesiones Virtuales iBGP como se muestra en la figura 2.11.

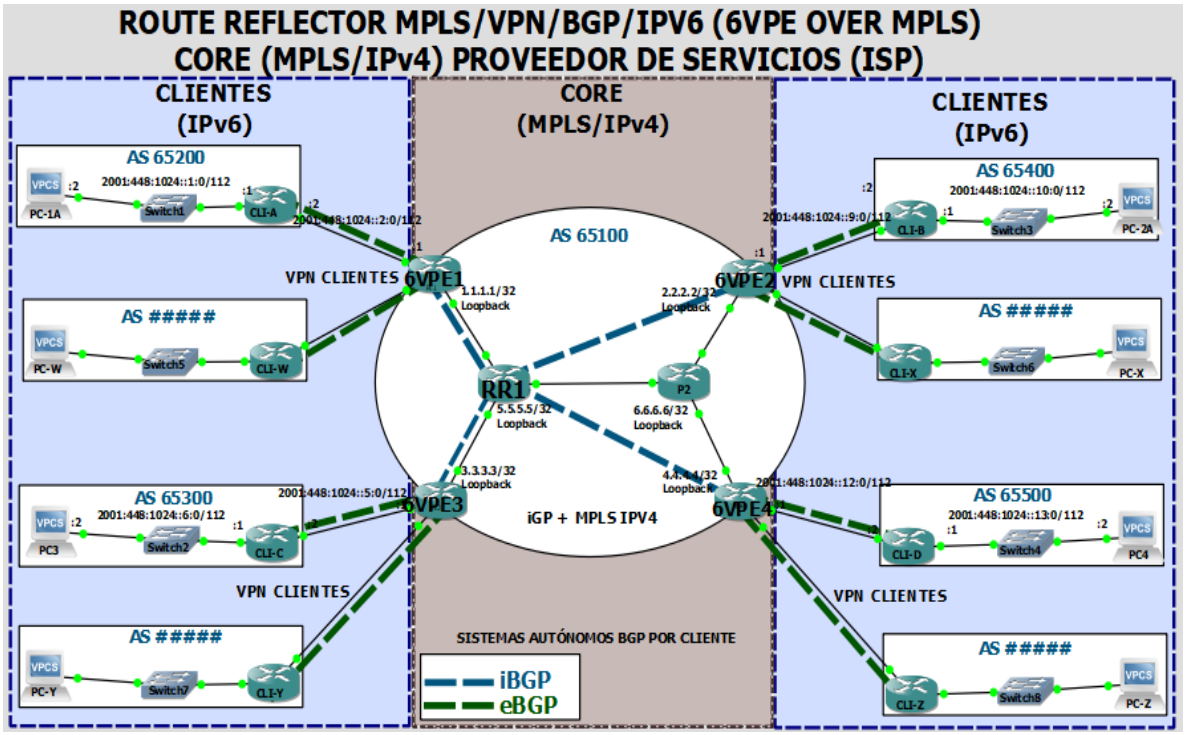


Figura 2.11 Escenario 2 - Route Reflector. Topología BGP/IPV6/VPN/MPLS

Como se puede observar las sesiones iBGP disminuyen (4 sesiones iBGP), ya que todos los *Routers* 6VPE sólo envían las rutas directamente al *Router Reflector* y este se encarga de anunciarlas hacia los otros *Routers* 6VPE, donde P_1 es usado como *Router Reflector* para las direcciones VPNv6 que reenvía las rutas para los sitios de las VPNv6.

En esta configuración se activan las 2 comunidades, *Extended y Standard*, con el comando *send-comunity both* para que los *Routers* 6VPE tomen las rutas que son insertadas en las VRFs y las anuncie como VPNv6 con sus RD y RT, quedando la configuración iBGP de los *Routers* 6VPE como se muestra en la tabla 2.5.

Tabla 2.5 Escenario 2. Sesiones iBGP Routers 6VPE

router bgp 65100 *neig 5.5.5.5 remote-as 65100* *neig 5.5.5.5 update-source loopback 0*
address-family vpnv6 *neig 5.5.5.5 activate* *neig 5.5.5.5 send-community both*

El *Router* P_1 que es asignado como *Router Reflector 1* (RR_1) crea un *Peer Group* para levantar sesiones con los diferentes *Routers* 6VPE, que simplifica la configuración y mejora la actualización de mensajes *Update BGP*. Se observa la configuración del *Router Reflector 1* (RR_1) en la tabla 2.6.

Tabla 2.6 Escenario 2. Sesiones iBGP Router Reflector RR_1

conf t *router bgp 65100* *neighbor RR1 peer-group* *neighbor RR1 remote-as 65100* *neighbor 1.1.1.1 peer-group RR1* *neighbor 2.2.2.2 peer-group RR1* *neighbor 3.3.3.3 peer-group RR1* *neighbor 4.4.4.4 peer-group RR1*
address-family vpnv6 *neighbor RR1 send-community both* *neighbor RR1 route-reflector-client* *neighbor 1.1.1.1 activate* *neighbor 2.2.2.2 activate* *neighbor 3.3.3.3 activate* *neighbor 4.4.4.4 activate*

Dentro de la familia VPNv6 se utiliza el comando *route-reflector-client* para establecer cuáles son los *Router* vecinos que van a ser tratados como clientes para anunciar las rutas aprendidas.

2.2.3 Escenario 3. Topología de Red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*): Con aplicación de método *Cluster Route Reflector*

Se procede aplicar el método *Cluster Route Reflector* a la topología de la figura 2.9 con el direccionamiento de la tabla 2.2, cambiando su configuración iBGP (8 sesiones iBGP) al convertir los *Router* del Proveedor (P_1, P_2) en los *Routers Reflectors* (RR_1, RR_2) respectivamente y los demás *Routers* en sus clientes. Las sesiones iBGP de los *Router* 6VPE se realiza con las *Loopback0* de los 2 *Routers Reflectors*, quedando sus sesiones virtuales iBGP como se muestra en la figura 2.12.

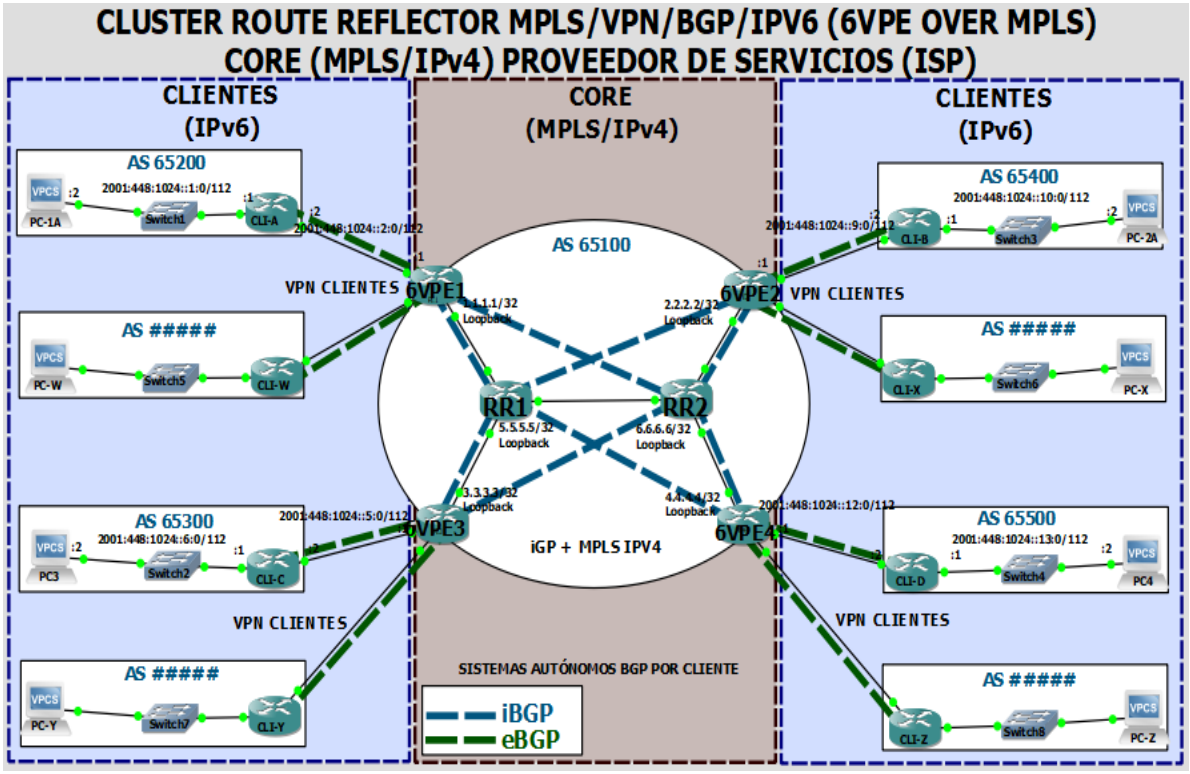


Figura 2.12 Escenario 3 - Cluster Route Reflector. Topología BGP/IPV6/VPN/MPLS

Los *Routers* 6VPE envían las rutas directamente a los *Routers Reflectors* (RR_1, RR_2) y estos se encargan de anunciarlas hacia los demás *Router* 6VPE. RR_1 y RR_2 son usados como *Routers Reflectors* para las direcciones VPNv6, los cuales reenvían las rutas para los sitios de las VPNs, quedando la configuración iBGP de los *Router* 6VPE como se muestra en la tabla 2.7.

Tabla 2.7 Escenario 3. Sesiones iBGP Routers 6VPE

router bgp 65100 *neighbor 5.5.5.5 remote-as 65100* *neighbor 5.5.5.5 update-source loopback 0* *neighbor 6.6.6.6 remote-as 65100* *neighbor 6.6.6.6 update-source loopback 0*
address-family vpnv6 *neighbor 5.5.5.5 activate* *neighbor 5.5.5.5 send-community both* *neighbor 6.6.6.6 activate* *neighbor 6.6.6.6 send-community both* *exit-address-family*

Los *Routers Reflectors* crean cada uno su propio *Peer Group* para levantar sesiones con los diferentes *Router* 6VPE, dentro de la familia VPNv6 de RR_1 y RR_2 se utiliza el comando *route-reflector-client* para establecer cuales son los *Router* vecinos que van a ser tratados como clientes, siendo los *Routers* con los que se van a anunciar las rutas aprendidas. Se observa la configuración de los *Routers Reflectors* RR_1 y RR_2 en la tabla 2.8.

Tabla 2.8 Escenario 3. Sesiones iBGP Routers Reflectors RR_1 y RR_2

RR_1	RR_2
router bgp 65100 *neighbor RR1 peer-group* *neighbor RR1 remote-as 65100* *neighbor 1.1.1.1 peer-group RR1* *neighbor 2.2.2.2 peer-group RR1* *neighbor 3.3.3.3 peer-group RR1* *neighbor 4.4.4.4 peer-group RR1*	*router bgp 65100* *neighbor RR2 peer-group* *neighbor RR2 remote-as 65100* *neighbor 1.1.1.1 peer-group RR2* *neighbor 2.2.2.2 peer-group RR2* *neighbor 3.3.3.3 peer-group RR2* *neighbor 4.4.4.4 peer-group RR2*
address-family vpnv6 *neighbor RR1 send-community both* *neighbor RR1 route-reflector-client* *neighbor 1.1.1.1 activate* *neighbor 2.2.2.2 activate* *neighbor 3.3.3.3 activate* *neighbor 4.4.4.4 activate*	*address-family vpnv6* *neighbor RR2 send-community both* *neighbor RR2 route-reflector-client* *neighbor 1.1.1.1 activate* *neighbor 2.2.2.2 activate* *neighbor 3.3.3.3 activate* *neighbor 4.4.4.4 activate*

2.2.4 Escenario 4. Topología de Red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*): Con aplicación de método *Confederations BGP*

Se implementa el método de *Confederations BGP* a la topología de la figura 2.9 con el direccionamiento de la tabla 2.2, separando el AS 65100 en 2 subAS (65101 y 65102), donde los *Routers* 6VPE1, P1 y 6VP3 pertenecen al subAS 65101 y los *Routers* 6VPE2, P2 y 6VP4 pertenecen al subAS 65102.

Los *Routers* 6VPE1 y 6VPE2 son los *Routers* de Borde de la Confederación nombrados $ASBR_1$ y $ASBR_2$ respectivamente, quienes funcionan como vecinos. Estos deben levantar una sesión eBGP entre los 2 subsistemas y se encargan de anunciar las rutas hacia los demás *Routers* 6VPE por iBGP dentro de cada subsistema (2 sesiones iBGP y 1 sesión eBGP), como se muestra en la topología de la figura 2.13.

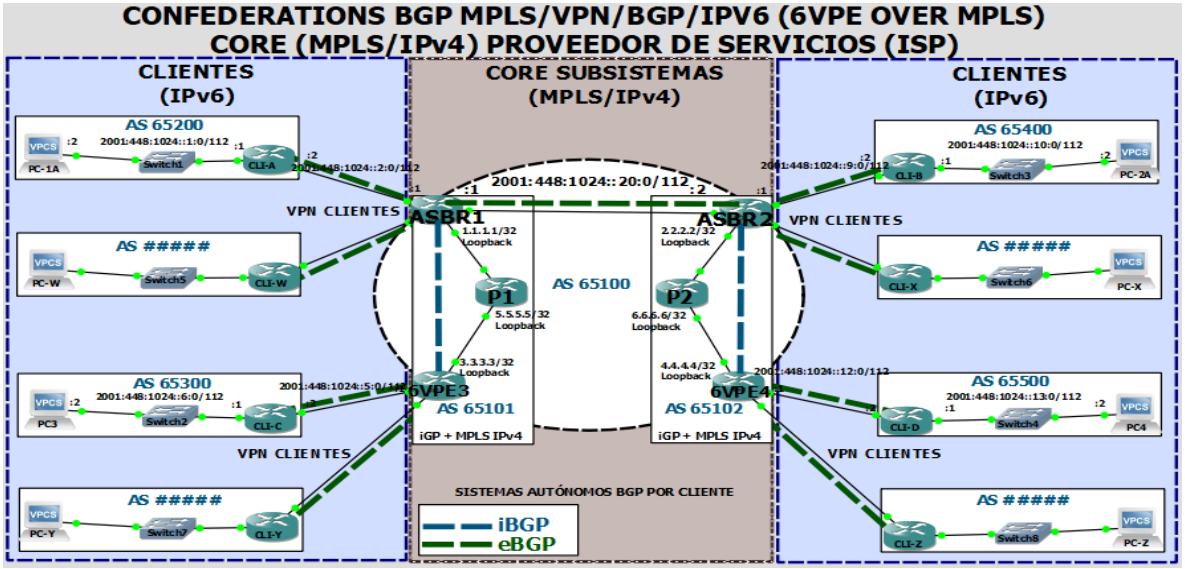


Figura 2.13 Escenario 4 - Confederations BGP.Topología BGP/IPV6/VPN/MPLS

Se asigna una conexión física entre los 2 *Routers* $ASBR_1$ y $ASBR_2$ que comunican por eBGP los 2 subsistemas, asignando la dirección IPv6 2001:448:1024::20:1 y 2001:448:1024::20:2 respectivamente. Cada subsistema internamente levanta sesiones iBGP, siendo entonces los *Peer* $ASBR_1$ con $6VPE_3$ y $ASBR_2$ con $6VPE_4$, como se muestra en la tabla 2.9.

Tabla 2.9 Escenario 4. Sesiones iBGP Routers $6VPE_3$ y $6VPE_4$

$6VPE_3$	**$6VPE_4$**
conf t *router bgp 65101* *bgp confederation identifier 65100* *bgp router-id 3.3.3.3* *neighbor 1.1.1.1 remote-as 65101* *neighbor 1.1.1.1 update-source Loopback 0*	*conf t* *router bgp 65102* *bgp confederation identifier 65100* *bgp router-id 4.4.4.4* *neighbor 2.2.2.2 remote-as 65102* *neighbor 2.2.2.2 update-source Loopback 0*
address-family vpnv6 *neighbor 1.1.1.1 activate* *neighbor 1.1.1.1 send-community both*	*address-family vpnv6* *neighbor 2.2.2.2 activate* *neighbor 2.2.2.2 send-community both*

Por medio de los comandos *router bgp 65101 y router bgp 65102* se define los 2 subAS, y con los comandos *bgp confederation identifier 65100 se* configuran los subsistemas como pertenecientes a la misma *Confederation BGP 65100.*

Se configuran las sesiones BGP (iBGP y eBGP) para los *Routers* $6VPE_1$ y $6VPE_2$, como se muestra en la tabla 2.10.

Tabla 2.10 Escenario 4. Sesiones BGP Routers $6VPE_1$ y $6VPE_2$

$6VPE_1$	**$6VPE_2$**
router bgp 65101 *bgp confederation identifier 65100* *bgp router-id 1.1.1.1* *no bgp default route-target filter* *bgp confederation peers 65102* *neighbor 3.3.3.3 remote-as 65101* *neighbor 3.3.3.3 update-source Loopback0* *neighbor 2001:448:1024::20:2 remote-as 65102* *neighbor 2001:448:1024::20:2 next-hop-self*	*router bgp 65102* *bgp router-id 2.2.2.2* *no bgp default route-target filter* *bgp confederation identifier 65100* *bgp confederation peers 65101* *neighbor 4.4.4.4 remote-as 65102* *neighbor 4.4.4.4 update-source Loopback0* *neighbor 2001:448:1024::20:1 remote-as 65101* *neighbor 2001:448:1024::20:1 next-hop-self*
address-family vpnv6 *neighbor 3.3.3.3 activate* *neighbor 3.3.3.3 send-community both* *neighbor 3.3.3.3 next-hop-self* *neighbor 2001:448:1024::20:2 activate* *neighbor 2001:448:1024::20:2 send-community both* *neighbor 2001:448:1024::20:2 next-hop-self*	*address-family vpnv6* *neighbor 4.4.4.4 activate* *neighbor 4.4.4.4 send-community both* *neighbor 4.4.4.4 next-hop-self* *neighbor 2001:448:1024::20:1 activate* *neighbor 2001:448:1024::20:1 send-community both* *neighbor 2001:448:1024::20:1 next-hop-self*

Los *Routers* vecinos internos de cada subsistema deben conocer sus direcciones por iBGP para que cuando se haga la conexión por eBGP entre los subsistemas, estos puedan conocer las direcciones de las VPNs de los clientes de cada subsistema de la *Confederation*. Por medio del comando *address-family-vpnv6* dentro de la configuración de *Router BGP* se redistribuyen las rutas aprendidas por BGP hacia las VRFs de los clientes.

2.2.5 Escenario 5. Topología de Red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*): Con aplicación de métodos *Confederations BGP y Route Reflector*

Se implementa el método de *Confederations BGP* junto con el método *Route Reflector*, donde se mantiene la *Confederation* del anterior Escenario subdividiendo el AS 65100 en SubAS (65101, 65102) y se utilizarán los *Routers* P_1 y P_2 como *Routers Reflectors* (RR_1 y RR_2) respectivamente dentro de cada subsistema, como muestra la topología de la figura 2.14.

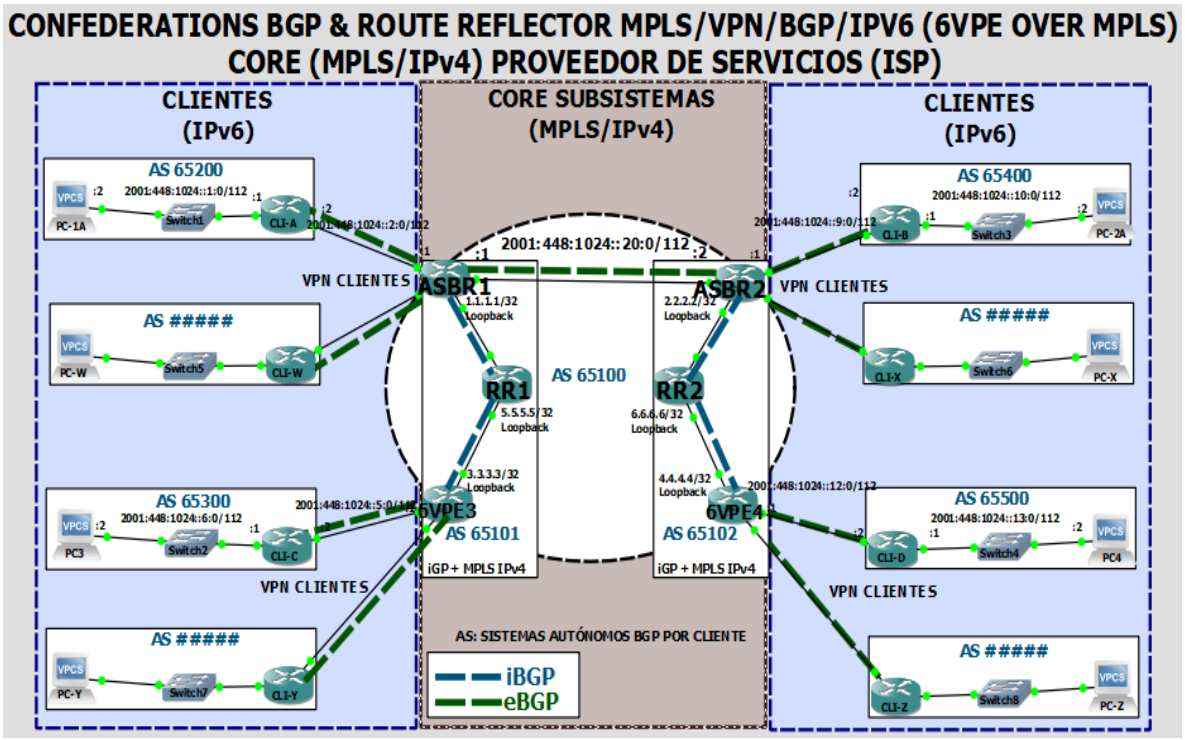


Figura 2.14 Escenario 5 - Confederations BGP y Route Reflector. Topología BGP/IPV6/VPN/MPLS

Los *Routers* de Borde de la *Confederation* nombrados $ASBR_1$ ($6VPE_1$) y $ASBR_2$ ($6VPE_2$) levantan una sesión por eBGP entre los 2 subsistemas, que se encargan de anunciar las rutas hacia el *Router Reflector* de cada subsistema por iBGP y este se encarga del reenvío hacia los *Routers* 6VPE (4 sesiones iBGP y 1 sesión eBGP), redistribuyendo las rutas aprendidas por BGP hacia las VRFs de los clientes.

La configuración de los *Routers* $6VPE_3$ y $6VPE_4$ de este Escenario se muestran en la tabla 2.11.

Tabla 2.11 Escenario 5. Sesiones iBGP Routers 6VPE$_3$ y 6VPE$_4$

6VPE$_3$	**6VPE$_4$**
router bgp 65101 *bgp router-id 3.3.3.3* *bgp confederation identifier 65100* *neighbor 5.5.5.5 remote-as 65101* *neighbor 5.5.5.5 update-source Loopback 0*	*router bgp 65102* *bgp router-id 4.4.4.4* *bgp confederation identifier 65100* *neighbor 6.6.6.6 remote-as 65102* *neighbor 6.6.6.6 update-source Loopback 0*
address-family vpnv6 *neighbor 5.5.5.5 activate* *neighbor 5.5.5.5 send-community both* *neighbor 5.5.5.5 next-hop-self*	*address-family vpnv6* *neighbor 6.6.6.6 activate* *neighbor 6.6.6.6 send-community both* *neighbor 6.6.6.6 next-hop-self*

Se levanta la sesión iBGP entre los *Routers* ASBR$_1$, RR$_1$, 6VPE$_3$ y entre los *Routers* ASBR$_2$, RR$_2$, 6VPE$_4$. Para cada subsistema de la *Confederation* se activa MPLS/IPv4 internamente en las interfaces correspondientes de los *Routers.* Para los *Routers Reflectors* RR$_1$ y RR$_2$ se asignan los comandos *bgp confederation identifier 65100* indicando que pertenecen a la *Confederation* 65100, se crean los grupos de pares (*peer-group)* para RR$_1$ y RR$_2$ optimizando la configuración con el comando *neighbor RR peer-group y se* configuran los clientes con el comando *neighbor RR route-reflector-client;* siendo los clientes de RR$_1$ los *Routers* ASBR$_1$ y 6VPE$_3$, y para RR$_2$ los *Routers* ASBR$_2$ y 6VPE$_4$. Las configuraciones de los *Routers Reflectors* RR$_1$ y RR$_2$ se muestran en la tabla 2.12.

Tabla 2.12 Escenario 5. Sesiones iBGP Routers Reflectors RR$_1$ y RR$_2$

RR$_1$	**RR$_2$**
router bgp 65101 *bgp router-id 5.5.5.5* *bgp confederation identifier 65100* *neighbor RR1 peer-group* *neighbor RR1 remote-as 65101* *neighbor 1.1.1.1 peer-group RR1* *neighbor 3.3.3.3 peer-group RR1*	*router bgp 65102* *bgp router-id 6.6.6.6* *bgp confederation identifier 65100* *neighbor RR2 peer-group* *neighbor RR2 remote-as 65102* *neighbor 2.2.2.2 peer-group RR2* *neighbor 4.4.4.4 peer-group RR2*
address-family vpnv6 *neighbor RR1 send-community both* *neighbor RR1 route-reflector-client* *neighbor 1.1.1.1 activate* *neighbor 3.3.3.3 activate*	*address-family vpnv6* *neighbor RR2 send-community both* *neighbor RR2 route-reflector-client* *neighbor 2.2.2.2 activate* *neighbor 4.4.4.4 activate*

Para los *Routers* de Borde de la *Confederation 65100 (*ASBR$_1$ y ASBR$_2$) se levanta la sesión eBGP al ser pertenecientes a los subsistemas 65101 y 65102 respectivamente, y se levanta la sesión iBGP como Clientes del *Router Reflector* (ASBR$_1$ con RR$_1$ y ASBR$_2$ con RR$_2$). Se observa la configuración de los *Routers* ASBR$_1$ y ASBR$_2$ en la tabla 2.13.

Tabla 2.13 Escenario 5. Sesiones BGP Routers ASBR$_1$ Y ASBR$_2$

ASBR$_1$	**ASBR$_2$**
router bgp 65101 *bgp confederation identifier 65100* *bgp router-id 1.1.1.1* *no bgp default route-target filter*	*router bgp 65102* *bgp router-id 2.2.2.2* *no bgp default route-target filter* *bgp confederation identifier 65100*

bgp confederation peers 65102 *neighbor 5.5.5.5 remote-as 65101* *neighbor 5.5.5.5 update-source Loopback0* *neighbor 2001:448:1024::20:2 remote-as 65102* *neighbor 2001:448:1024::20:2 next-hop-self*	*bgp confederation peers 65101* *neighbor 6.6.6.6 remote-as 65102* *neighbor 6.6.6.6 update-source Loopback0* *neighbor 2001:448:1024::20:1 remote-as 65101* *neighbor 2001:448:1024::20:1 next-hop-self*
address-family vpnv6 *neighbor 5.5.5.5 activate* *neighbor 5.5.5.5 send-community both* *neighbor 5.5.5.5 next-hop-self* *neighbor 2001:448:1024::20:2 activate* *neighbor 2001:448:1024::20:2 send-community both* *neighbor 2001:448:1024::20:2 next-hop-self*	*address-family vpnv6* *neighbor 6.6.6.6 activate* *neighbor 6.6.6.6 send-community both* *neighbor 6.6.6.6 next-hop-self* *neighbor 2001:448:1024::20:1 activate* *neighbor 2001:448:1024::20:1 send-community both* *neighbor 2001:448:1024::20:1 next-hop-self*

3. PRUEBAS DE FUNCIONALIDAD DE RED BGP/IPV6/VPN/MPLS (*6VPE OVER MPLS*) SIN MÉTODOS – CONEXIÓN *FULL MESH* (ESCENARIO 1)

Los Escenarios fueron implementados en un entorno de pruebas virtualizado por medio del *Software* GNS3 y su funcionamiento se validó con la herramienta de análisis de protocolos y rastreo *Wireshark* que viene implementado en el *Software*. Se capturan los paquetes para el análisis de protocolos, donde se procede a dar *click* derecho a la interfaz que se desea analizar y *start-capture,* como se muestra en la figura 3.1.

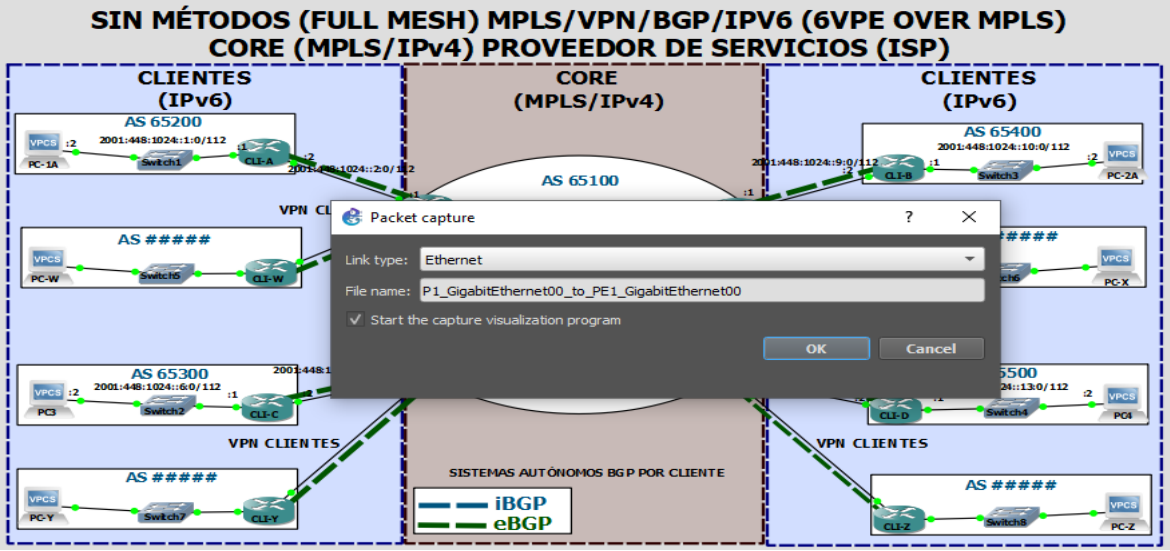


Figura 3.1 Captura de Paquetes con Wireshark

3.1 Verificación de Conectividad entre Clientes: Red BGP/IPV6/VPN/MPLS sin métodos – Conexión *Full Mesh*

Se utiliza el comando *ping* desde el *Router* CLI-A como fuente de los mensajes hasta el *Router* CLI-B como destino, y se comprueba la conexión como se muestra en la figura 3.2.

```
CLI-A#ping  2001:448:1024::9:2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:448:1024::9:2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/46/56 ms
```

Figura 3.2 Ping Router CLI-A a CLI-B

Se capturan las tramas en una interfaz dentro del *CORE,* y *se* comprueba conectividad entre los clientes por medio de mensajes ICMPv6 (*Internet Control Message Protocol version 6*) entre los *Routers* CLI-A y CLI-B, como se muestra en la figura 3.3.

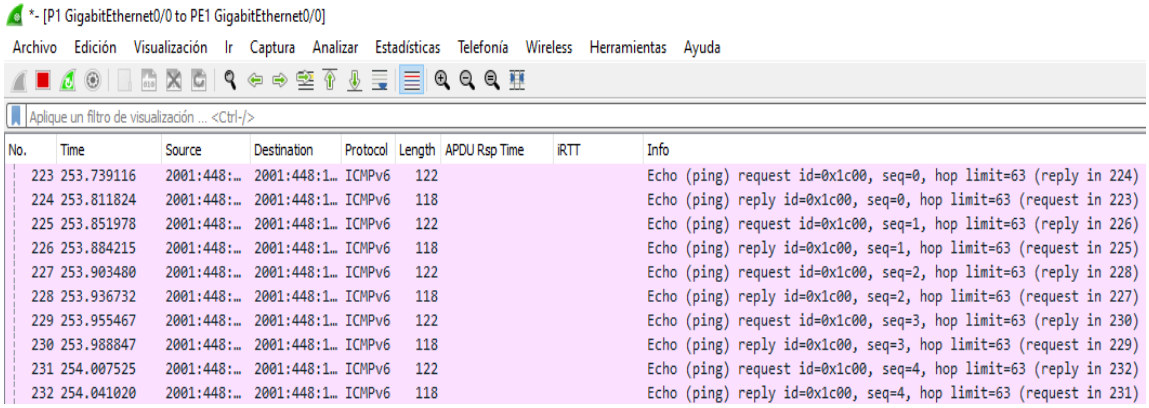


Figura 3.3 Mensajes ICMPv6

Se ejecuta el comando *traceroute* con el fin de analizar los saltos que toma el paquete para llegar a su destino dentro de la red y se pueden observar las etiquetas del proceso MPLS en la figura 3.4.

```
CLI-A#traceroute 2001:448:1024::9:2
Type escape sequence to abort.
Tracing the route to 2001:448:1024::9:2

  1 2001:448:1024::2:1 8 msec 12 msec 20 msec
  2 ::FFFF:10.10.3.1 [MPLS: Labels 17/26 Exp 0] 28 msec 52 msec 56 msec
  3 ::FFFF:10.10.7.2 [MPLS: Labels 21/26 Exp 0] 56 msec 60 msec 52 msec
  4 2001:448:1024::9:1 [AS 65100] 52 msec 68 msec 52 msec
  5 2001:448:1024::9:2 [AS 65100] 52 msec 56 msec 72 msec
```

Figura 3.4 Traceroute Router CLI-A a CLI-B

Analizando la figura 3.4, se observan los 5 saltos que toma el paquete en llegar al destino desde el *Router* CLI-A hasta el *Router* CLI-B, tal como se representa en la trayectoria oscura de la figura 3.5.

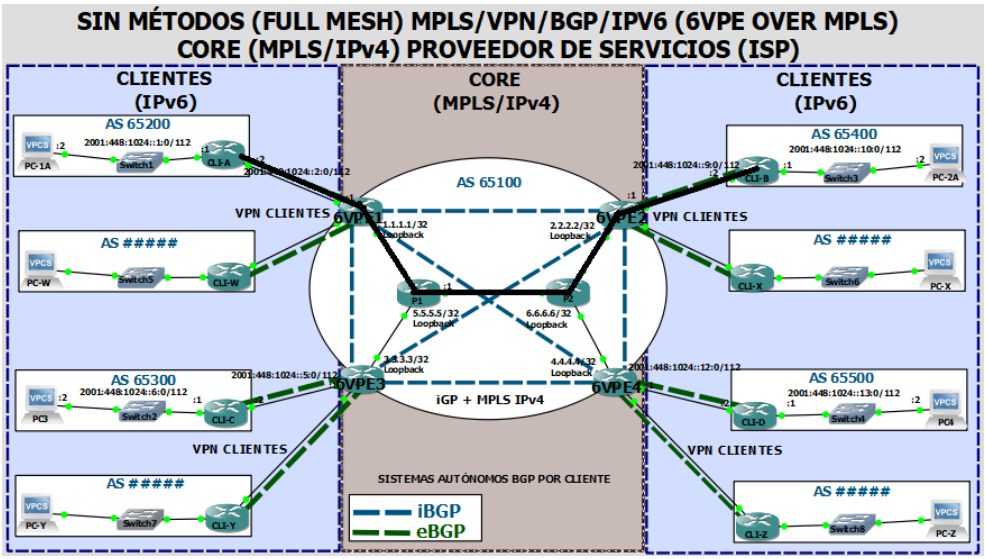


Figura 3.5 Traceroute Router CLI-A a CLI-B en Topología Full Mesh

Nota: La anterior comprobación de conexión se realizó para todos los *Routers* de los clientes, de forma general.

Para el Escenario 1 sin aplicación de métodos se establece una conexión *Full Mesh* entre los 6VPE, donde se tienen 6 sesiones iBGP levantadas entre las direcciones de *Loopback0* de estos *Routers*, los cuales también levantan sesiones de *Neighbor LDP* por MPLS con los *Routers* P_1 y P_2, y se asignan interfaces de *Loopback0* a los *Routers* CLI. Se observan las direcciones de *Loopback0* de los *Routers* P, 6VPE y CLI en la tabla 3.1.

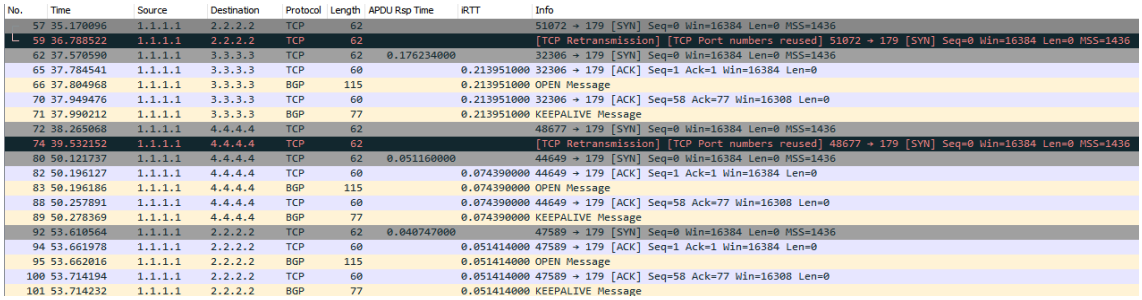
Tabla 3.1 Direcciones Loopback 0

Router	**Dirección**
$6VPE_1$	1.1.1.1
$6VPE_2$	2.2.2.2
$6VPE_3$	3.3.3.3
$6VPE_4$	4.4.4.4
P_1	5.5.5.5
P_2	6.6.6.6
CLI-A	7::7
CLI-B	8::8
CLI-C	9::9
CLI-D	10::10

Para el Escenario 1, se procede a realizar el análisis de protocolos con la herramienta *Wireshark* para comprobar los parámetros de las sesiones BGP a nivel de mensajes *UPDATE*, verificación de asignación de etiquetas MPLS por los *Routers* 6VPE, visualización de puertos TCP al levantar sesiones BGP y medidas de rendimiento como *RTT*, *Frame Arrival Delay* y *Jitter*.

3.2 Etiquetas MPLS de las VPNs Red BGP/IPV6/VPN/MPLS sin métodos – Conexión *Full Mesh*

Se verifican las etiquetas MPLS asignadas usando el filtro: *tcp and mpls,* en la herramienta *Wireshark* entre todas las interfaces internas del *CORE* para posteriormente desplegar la información de cada conexión. Se muestra la interfaz entre $6VPE_1$ y P_1 en la figura 3.6*,* pero se realiza en todas las interfaces.



No.	Time	Source	Destination	Protocol	Length	APDU Rsp Time	IRTT	Info
57	35.170096	1.1.1.1	2.2.2.2	TCP	62			51072 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
59	36.788522	1.1.1.1	2.2.2.2	TCP	62			[TCP Retransmission] [TCP Port numbers reused] 51072 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
62	37.570590	1.1.1.1	3.3.3.3	TCP	62	0.176234000		32306 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
65	37.784541	1.1.1.1	3.3.3.3	TCP	60		0.213951000	32306 → 179 [ACK] Seq=1 Ack=1 Win=16384 Len=0
66	37.804968	1.1.1.1	3.3.3.3	BGP	115		0.213951000	OPEN Message
70	37.949476	1.1.1.1	3.3.3.3	TCP	60		0.213951000	32306 → 179 [ACK] Seq=58 Ack=77 Win=16308 Len=0
71	37.990212	1.1.1.1	3.3.3.3	BGP	77		0.213951000	KEEPALIVE Message
72	38.265068	1.1.1.1	4.4.4.4	TCP	62			48677 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
74	39.532152	1.1.1.1	4.4.4.4	TCP	62			[TCP Retransmission] [TCP Port numbers reused] 48677 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
80	50.121737	1.1.1.1	4.4.4.4	TCP	62	0.051160000		44649 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
82	50.196127	1.1.1.1	4.4.4.4	TCP	60		0.074390000	44649 → 179 [ACK] Seq=1 Ack=1 Win=16384 Len=0
83	50.196186	1.1.1.1	4.4.4.4	BGP	115		0.074390000	OPEN Message
88	50.257891	1.1.1.1	4.4.4.4	TCP	60		0.074390000	44649 → 179 [ACK] Seq=58 Ack=77 Win=16308 Len=0
89	50.278369	1.1.1.1	4.4.4.4	BGP	77		0.074390000	KEEPALIVE Message
92	53.610564	1.1.1.1	2.2.2.2	TCP	62	0.040747000		47589 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
94	53.661978	1.1.1.1	2.2.2.2	TCP	60		0.051414000	47589 → 179 [ACK] Seq=1 Ack=1 Win=16384 Len=0
95	53.662016	1.1.1.1	2.2.2.2	BGP	115		0.051414000	OPEN Message
100	53.714194	1.1.1.1	2.2.2.2	TCP	60		0.051414000	47589 → 179 [ACK] Seq=58 Ack=77 Win=16308 Len=0
101	53.714232	1.1.1.1	2.2.2.2	BGP	77		0.051414000	KEEPALIVE Message

Figura 3.6 Sesiones TCP (iBGP - $6VPE_1$)

Se analizan las sesiones TCP de todas las interfaces y se observan los puertos TCP de las 6 sesiones iBGP del Escenario 1, donde al darle *Click* a cada conexión obtenemos información en los puertos TCP (*Src Port*) y las etiquetas MPLS (*MPLS Label*) asignadas en el *CORE*, como se observa en la figura 3.7, figura 3.8 y figura 3.9.

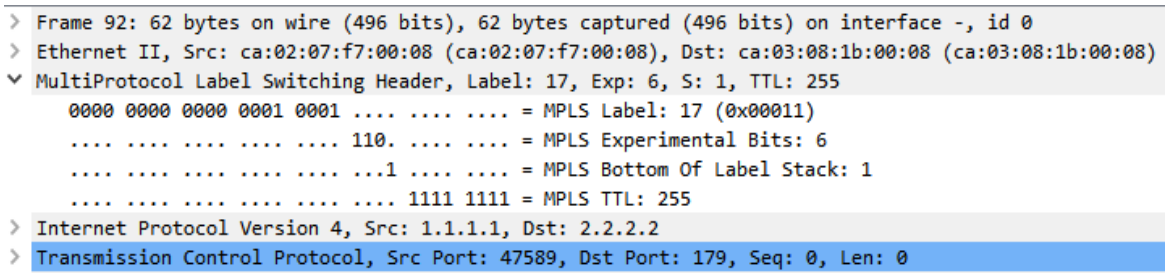


Figura 3.7 Puerto TCP y Etiqueta MPLS (6VPE$_1$ - 6VPE$_2$)

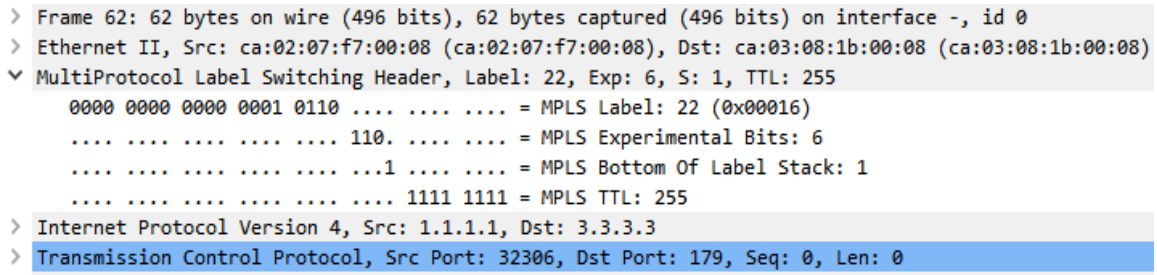


Figura 3.8 Puerto TCP y Etiqueta MPLS (6VPE$_1$ - 6VPE$_3$)

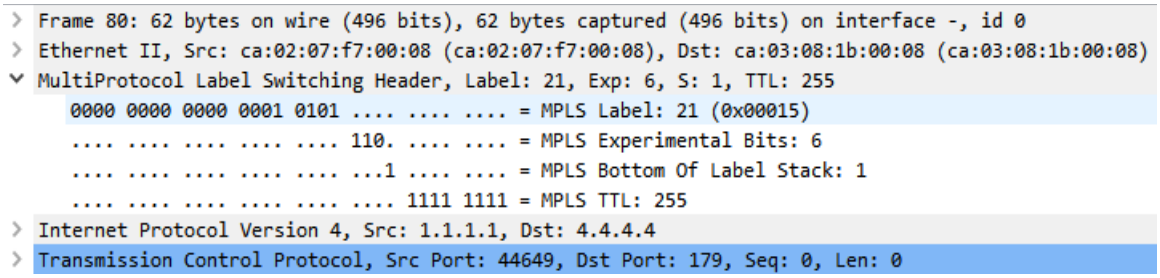


Figura 3.9 Puerto TCP y Etiqueta MPLS (6VPE$_1$ - 6VPE$_4$)

Con la información obtenida de *Wireshark* se genera la tabla 3.2, donde se muestran los Puertos TCP asignados a las sesiones BGP y etiquetas MPLS en el Escenario 1.

Tabla 3.2 Puertos TCP de sesiones iBGP y Etiquetas MPLS

Conexión	Puerto TCP	Etiqueta
6VPE$_1$ - 6VPE$_2$	47589	17
6VPE$_1$ - 6VPE$_3$	32306	22
6VPE$_1$ - 6VPE$_4$	44649	21
6VPE$_2$ - 6VPE$_3$	25835	23
6VPE$_2$ - 6VPE$_4$	13383	25
6VPE$_3$ - 6VPE$_4$	30776	21

También se verifica la sesión LDP, para la propagación de las rutas de los clientes VPNv6, como se observa en la figura 3.10.

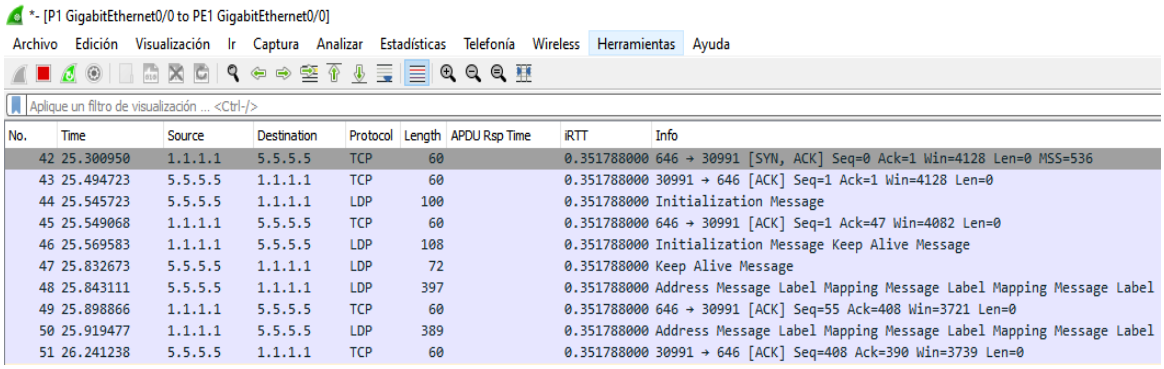


Figura 3.10 Trama LDP

Para el Escenario 1 se comprueban las sesiones iBGP que establecen los *Routers* 6VPE con los demás *Routers* 6VPE, las sesiones LDP que se establecen entre los *Routers* P_1, P_2, 6VPE y de las sesiones VRF de las VPNv6 de los clientes por eBGP conectados a sus interfaces, como se muestran en la figura 3.11, figura 3.12, figura 3.13, figura 3.14, figura 3.15 y figura 3.16.

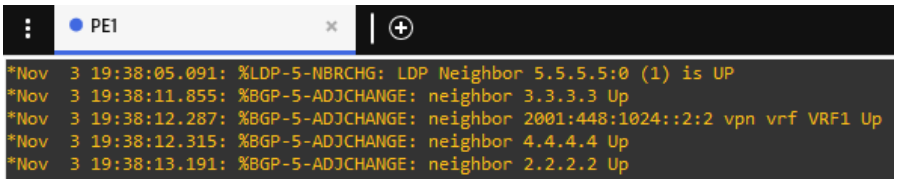


Figura 3.11 Establecimiento Neighbors iBGP, VRF y LDP. Router 6VPE$_1$

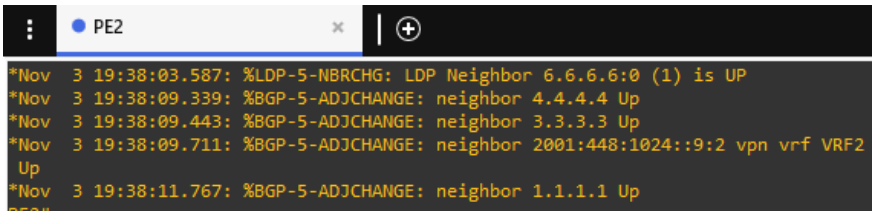


Figura 3.12 Establecimiento Neighbors iBGP, VRF y LDP. Router 6VPE$_2$

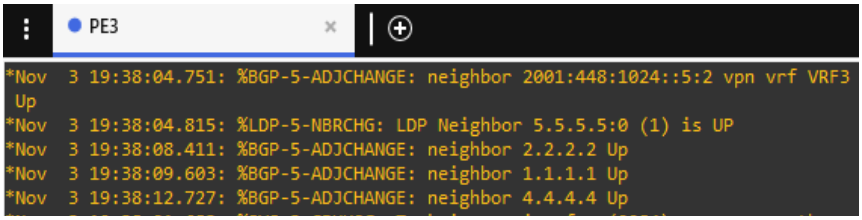


Figura 3.13 Establecimiento Neighbors iBGP, VRF y LDP. Router 6VPE$_3$

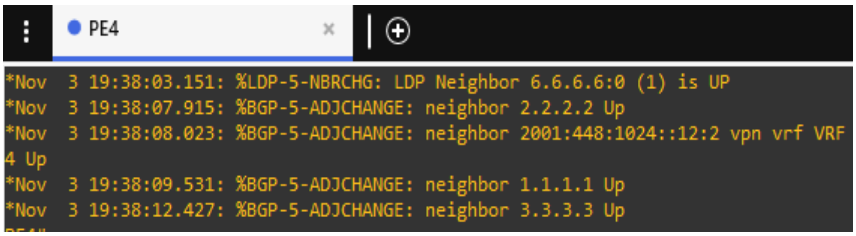


Figura 3.14 Establecimiento Neighbors iBGP, VRF y LDP. Router 6VPE$_4$

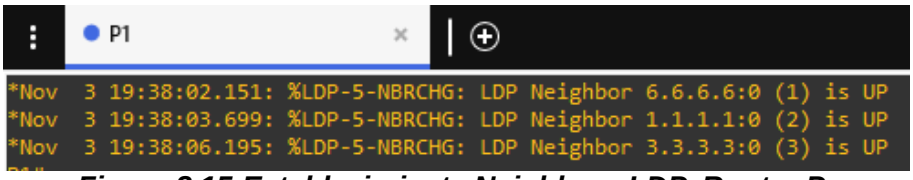


Figura 3.15 Establecimiento Neighbors LDP. Router P_1

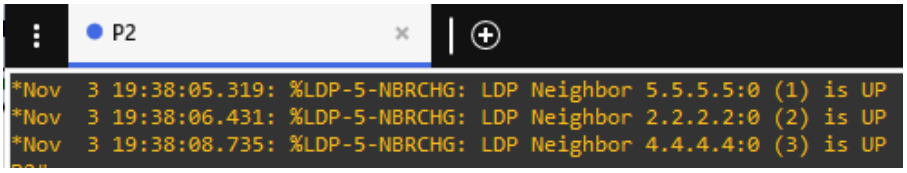


Figura 3.16 Establecimiento Neighbors LDP. Router P_2

3.3 Mediciones Estadísticas TCP Red BGP/IPV6/VPN/MPLS sin métodos – Conexión *Full Mesh*

Para la respectiva medición de estadísticas se activa el Protocolo *TRANSUM* que extiende las capacidades de análisis de rendimiento de la red y mediciones de *Wireshark,* permitiendo analizar los tiempos de respuesta de servicios e identificar fallas que causan tiempos de convergencia lentos.

Wireshark actualmente trae implementado *TRANSUM,* pero no viene activado, por lo cual en el menú principal de *Wireshark* en la pestaña *Analizar*, Protocolos *activados,* se busca el Protocolo *TRANSUM* y se activa. Con esto y algunos filtros permite analizar las estadísticas de la trama TCP, como se observa en la figura 3.17.

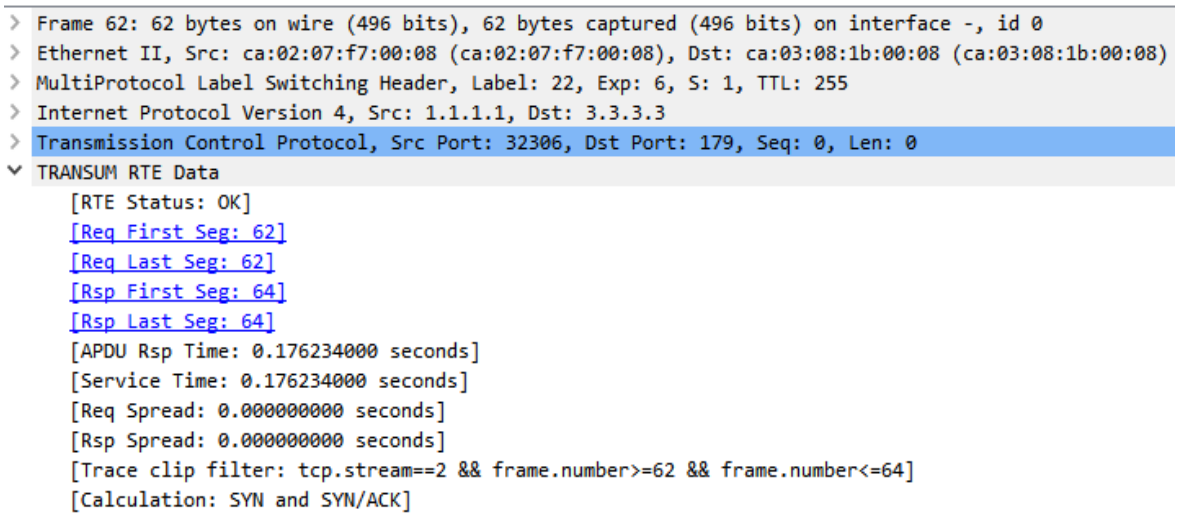


Figura 3.17 Captura Transum

Se utiliza el filtro: *tcp.flags.syn==1.* Permite visualizar los paquetes que inician las sesiones BGP entre los *Routers* 6VPE (*TCP SYN - ACK*), así como de las sesiones LDP entre los *Routers* 6VPE y P, esto se muestra en la figura 3.18, figura 3.19, figura 3.20 y figura 3.21.

No.	Time	Source	Destination	Protoc	Length	iRTT	Info
40	25.142935	5.5.5.5	1.1.1.1	TCP	60		30991 → 646 [SYN] Seq=0 Win=4128 Len=0 MSS=536
42	25.300950	1.1.1.1	5.5.5.5	TCP	60	0.351788000	646 → 30991 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
57	35.170096	1.1.1.1	2.2.2.2	TCP	62		51072 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
59	36.788522	1.1.1.1	2.2.2.2	TCP	62		[TCP Retransmission] [TCP Port numbers reused] 51072 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
62	37.570590	1.1.1.1	3.3.3.3	TCP	62		32306 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
64	37.746824	3.3.3.3	1.1.1.1	TCP	58	0.213951000	179 → 32306 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
72	38.265068	1.1.1.1	4.4.4.4	TCP	62		48677 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
74	39.532152	1.1.1.1	4.4.4.4	TCP	62		[TCP Retransmission] [TCP Port numbers reused] 48677 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
80	50.121737	1.1.1.1	4.4.4.4	TCP	62		44649 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
81	50.172897	4.4.4.4	1.1.1.1	TCP	58	0.074390000	179 → 44649 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
92	53.610564	1.1.1.1	2.2.2.2	TCP	62		47589 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
93	53.651311	2.2.2.2	1.1.1.1	TCP	58	0.051414000	179 → 47589 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436

Figura 3.18 Captura iRTT Interfaz 6VPE₁ - P₁

No.	Time	Source	Destination	Protoc	Length	iRTT	Info
35	8.859346	6.6.6.6	2.2.2.2	TCP	60		30499 → 646 [SYN] Seq=0 Win=4128 Len=0 MSS=536
36	8.900775	2.2.2.2	6.6.6.6	TCP	60	0.155539000	646 → 30499 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
38	9.035357	3.3.3.3	2.2.2.2	TCP	60		25835 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
40	9.046400	2.2.2.2	3.3.3.3	TCP	60	0.006620000	179 → 25835 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
53	10.952842	3.3.3.3	2.2.2.2	TCP	58	0.006620000	[TCP Retransmission] [TCP Port numbers reused] 25835 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
56	12.377726	2.2.2.2	3.3.3.3	TCP	62	0.006620000	[TCP Retransmission] 179 → 25835 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
68	17.011270	4.4.4.4	2.2.2.2	TCP	58		13383 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
69	17.025604	2.2.2.2	4.4.4.4	TCP	62	0.052586000	179 → 13383 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
79	17.808401	1.1.1.1	2.2.2.2	TCP	58		47589 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
80	17.821273	2.2.2.2	1.1.1.1	TCP	62	0.051823000	179 → 47589 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436

Figura 3.19 Captura iRTT Interfaz 6VPE₃ - P₂

No.	Time	Source	Destination	Protoc	Length	iRTT	Info
36	24.855157	5.5.5.5	3.3.3.3	TCP	60		13429 → 646 [SYN] Seq=0 Win=4128 Len=0 MSS=536
38	24.922324	3.3.3.3	5.5.5.5	TCP	60	0.320322000	646 → 13429 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
56	37.305531	1.1.1.1	3.3.3.3	TCP	58		32306 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
57	37.474670	3.3.3.3	1.1.1.1	TCP	62	0.215204000	179 → 32306 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
68	41.731376	3.3.3.3	4.4.4.4	TCP	62		63504 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
69	43.456335	3.3.3.3	4.4.4.4	TCP	62		[TCP Retransmission] [TCP Port numbers reused] 63504 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
70	44.499360	3.3.3.3	2.2.2.2	TCP	62		25835 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
72	46.491244	3.3.3.3	2.2.2.2	TCP	62	0.021598000	[TCP Retransmission] [TCP Port numbers reused] 25835 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
76	47.938682	2.2.2.2	3.3.3.3	TCP	58	0.021598000	[TCP Port numbers reused] 179 → 25835 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
87	52.568506	4.4.4.4	3.3.3.3	TCP	58		30776 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
88	52.652589	3.3.3.3	4.4.4.4	TCP	62	0.113307000	179 → 30776 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436

Figura 3.20 Captura iRTT Interfaz 6VPE₂ - P₁

No.	Time	Source	Destination	Protoc	Length	iRTT	Info
35	10.194436	6.6.6.6	4.4.4.4	TCP	60		22113 → 646 [SYN] Seq=0 Win=4128 Len=0 MSS=536
37	10.231277	4.4.4.4	6.6.6.6	TCP	60	0.095366000	646 → 22113 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
51	15.417756	1.1.1.1	4.4.4.4	TCP	58		44649 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
52	15.438673	4.4.4.4	1.1.1.1	TCP	62	0.072114000	179 → 44649 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
63	18.103053	4.4.4.4	2.2.2.2	TCP	62		13383 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
64	18.103218	4.4.4.4	3.3.3.3	TCP	62		30776 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1436
65	18.129967	2.2.2.2	4.4.4.4	TCP	58	0.050732000	179 → 13383 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436
74	18.212548	3.3.3.3	4.4.4.4	TCP	58	0.113514000	179 → 30776 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1436

Figura 3.21 Captura iRTT Interfaz 6VPE₄ - P₂

Se puede observar en la figura 3.22 el valor para tiempo de ida y vuelta inicial iRTT que es la latencia inicial de los mensajes TCP cuando se realiza el establecimiento de la conexión *Three-way Hanshake* entre los *Routers* 6VPE1 y 6VPE2, para lo cual se toman los dos primeros mensajes (SYN y SYN-ACK) respecto a cada segmento como iRTT para cada sesión iBGP, estos mensajes se pueden analizar en la sección *TCP, SEQ/ACK análisis.*

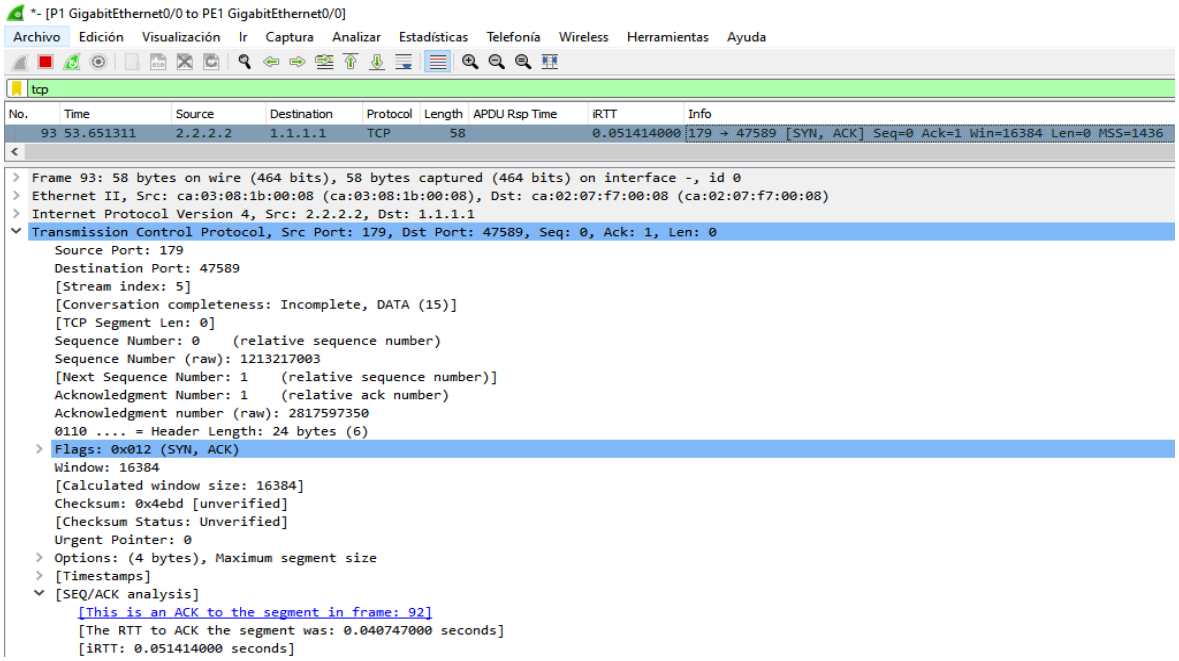


Figura 3.22 Initial Round Trip Time

Para observar gráficamente el RTT de las sesiones TCP en *Wireshark*, se accede desde el menú principal, *Estadísticas, Gráficas de secuencia TCP, Round Trip Time* como se observa en la figura 3.23.

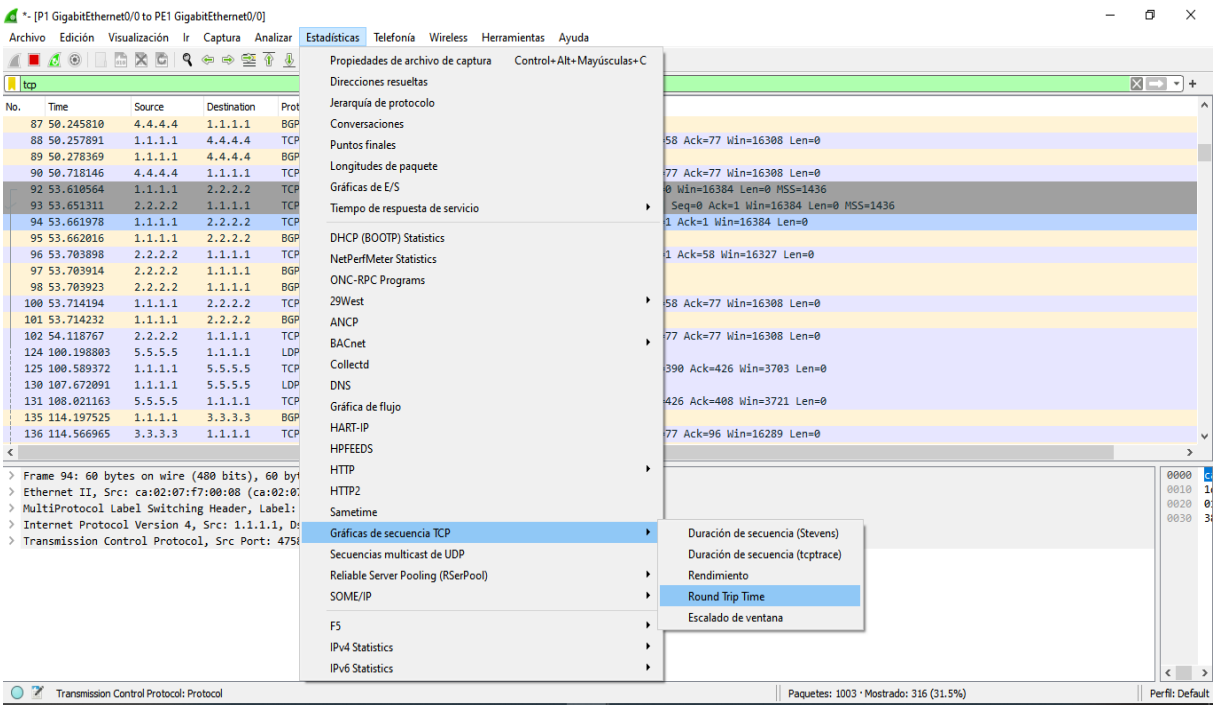


Figura 3.23 Generación Gráficas de Secuencia TCP

Por lo tanto, se genera la gráfica de RTT para la sesión iBGP entre 6VPE$_1$ y 6VPE$_2$, tomando la secuencia TCP en *Wireshark*, como se observa en la figura 3.24.

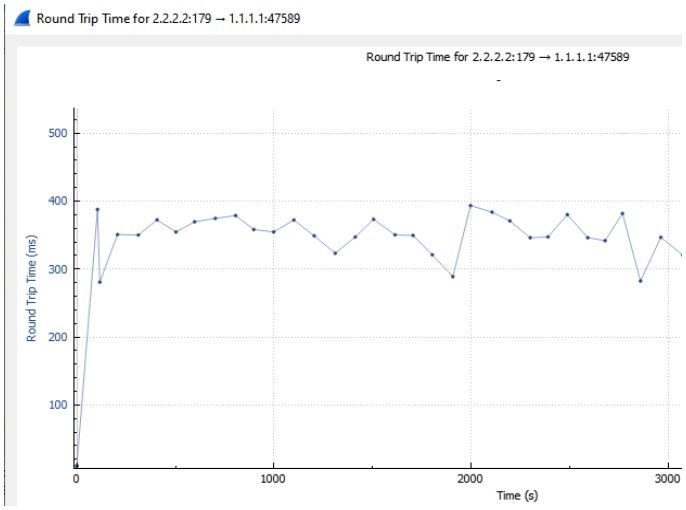


Figura 3.24 Round Trip Time Sesión TCP 6VPE$_1$ - 6VPE$_2$

Se aplica el filtro: *tcp and bgp and ip.src,* donde se selecciona un paquete BGP y en la sección de *Frame*, se agrega *Time delta from previos captured frame* como columna para poder visualizar la diferencia del tiempo de la última trama capturada, como se muestra en la figura 3.25.

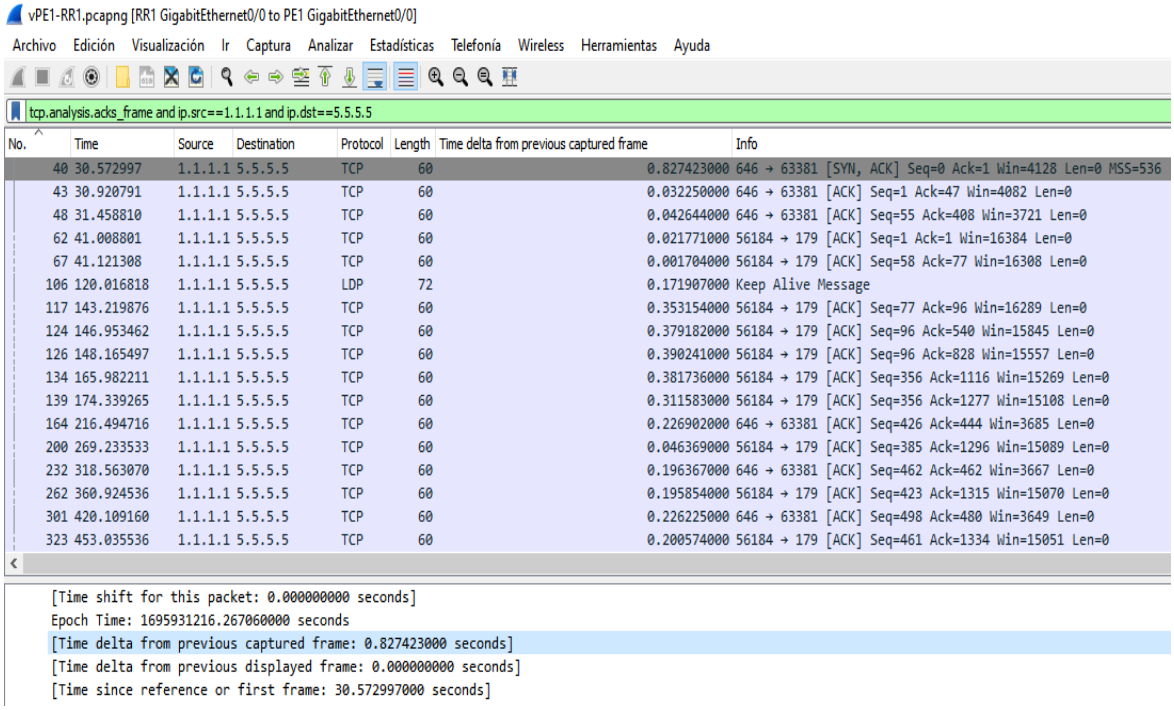


Figura 3.25 Time Delta from Previous Captured Frame

Para visualizar gráficamente el *Frame Arrival Delay* se apoya en las herramientas de *Wireshark*, donde en el menú principal se accede a *Estadísticas* y *Gráficas de E/S*. Se configura como filtro: *tcp and bgp,* en Y Axis se configura el valor máximo *MAX (Y Field)*, y en *Y Field* se usa el valor *frame.time_delta* que es el tiempo de retraso de cada trama, como se muestra en la figura 3.26.

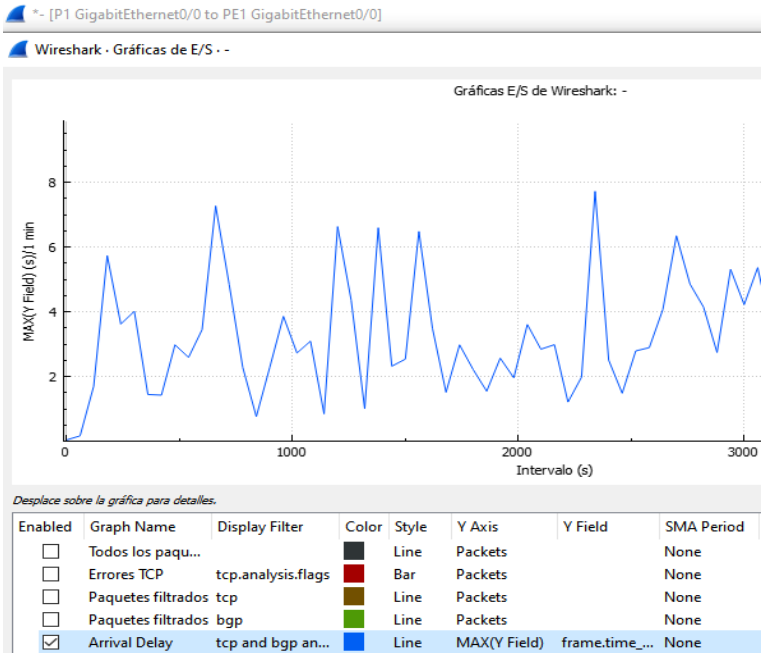


Figura 3.26 Frame Arrival Delay 6VPE₁ - P₁

Para medir el delay entre tramas también se apoya en la herramienta de *Wireshark, Gráficas de E/S* se mide el tiempo entre cada trama apoyándose del filtro: *tcp.stream eq* , y en el campo *YField* el filtro: *frame.time_delta_displayed,* como se muestra en la figura 3.27.

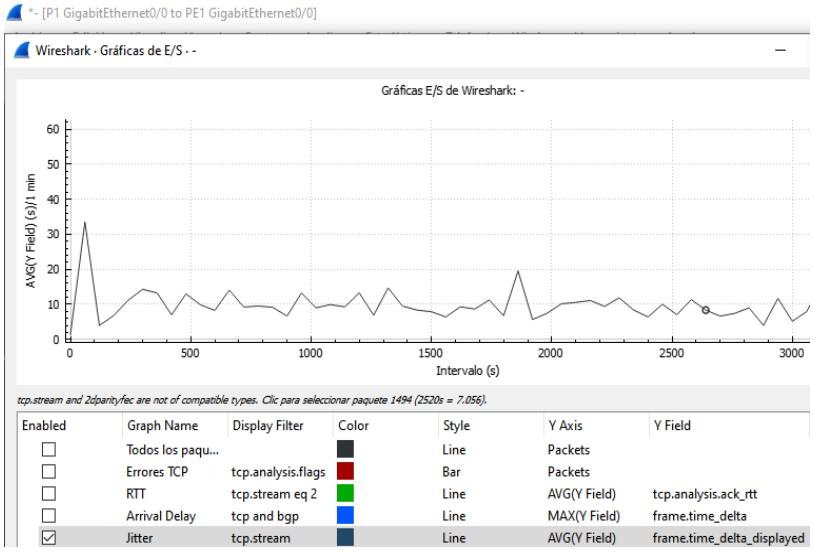


Figura 3.27 Jitter 6VPE₁ - P₁

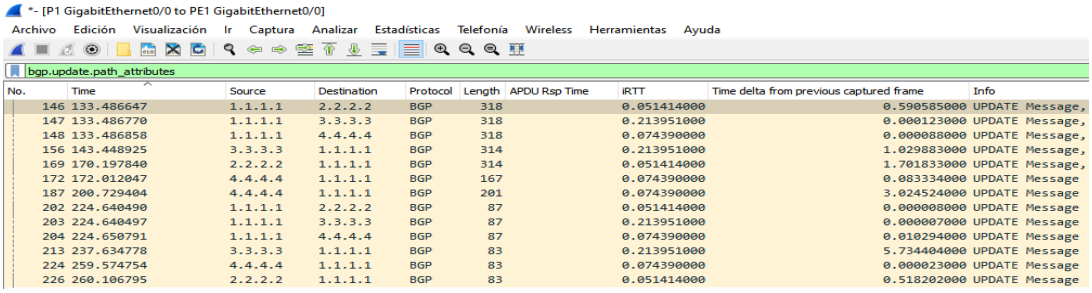
4. MEDICIONES Y ANÁLISIS DE RESULTADOS EN LOS ESCENARIOS

4.1 Escenario 1: Topología sin métodos

Como se pudo observar en la simulación del Escenario 1 sin métodos, se crea una conexión *Full Mesh* entre los *Routers* 6VPE, estableciendo 6 sesiones iBGP, donde según lo planteado se realizan pruebas, mediciones, análisis de mensajes *Update* y sesiones TCP durante un tiempo prolongado de 50 minutos al igual que para todos los Escenarios. Se garantiza la captura de suficientes datos para analizar el comportamiento de la red, puesto que en las pruebas se evidencia que la red converge durante los primeros minutos y tiende a estabilizarse.

4.1.1 Análisis iBGP - *Full Mesh*

Se realiza la captura de los mensajes *Update* en los diferentes *Routers* 6VPE para el Escenario 1 *Full Mesh* sin métodos, con el filtro: *bgp.update.path_attributes,* se obtiene el tamaño de los mensajes *Update (Frame Length)* de las sesiones BGP como se muestra en la figura 4.1, y se generan las gráficas de flujo.



No.	Time	Source	Destination	Protocol	Length	APDU Rsp Time	iRTT	Time delta from previous captured frame	Info
146	133.486647	1.1.1.1	2.2.2.2	BGP	318		0.051414000	0.590585000	UPDATE Message,
147	133.486770	1.1.1.1	3.3.3.3	BGP	318		0.213951000	0.000123000	UPDATE Message,
148	133.486858	1.1.1.1	4.4.4.4	BGP	318		0.074390000	0.000088000	UPDATE Message,
156	143.448925	3.3.3.3	1.1.1.1	BGP	314		0.213951000	1.029883000	UPDATE Message,
169	170.197840	2.2.2.2	1.1.1.1	BGP	314		0.051414000	1.701833000	UPDATE Message,
172	172.012047	4.4.4.4	1.1.1.1	BGP	167		0.074390000	0.083334000	UPDATE Message
187	200.729404	4.4.4.4	1.1.1.1	BGP	201		0.074390000	3.024524000	UPDATE Message
202	224.640490	1.1.1.1	2.2.2.2	BGP	87		0.051414000	0.000008000	UPDATE Message
203	224.640497	1.1.1.1	3.3.3.3	BGP	87		0.213951000	0.000007000	UPDATE Message
204	224.650791	1.1.1.1	4.4.4.4	BGP	87		0.074390000	0.010294000	UPDATE Message
213	237.634778	3.3.3.3	1.1.1.1	BGP	83		0.213951000	5.734404000	UPDATE Message
224	259.574754	4.4.4.4	1.1.1.1	BGP	83		0.074390000	0.000023000	UPDATE Message
226	260.106795	2.2.2.2	1.1.1.1	BGP	83		0.051414000	0.518202000	UPDATE Message

Figura 4.1 Escenario 1. Mensajes Update Router 6VPE$_1$

4.1.1.1 iBGP *Full Mesh* - *Router* 6VPE$_1$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_1$ con los demás *Routers* 6VPE en la figura 4.2.

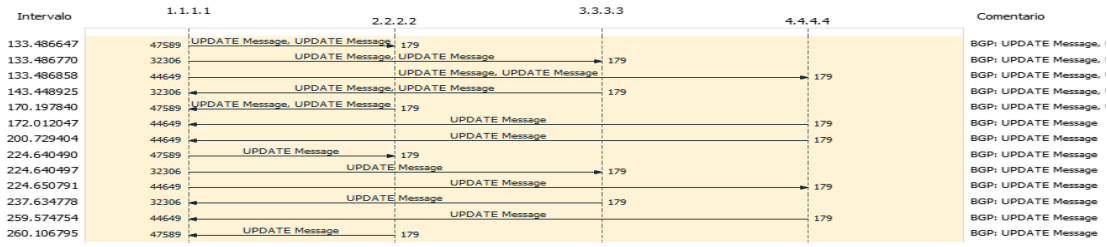


Figura 4.2 Escenario 1. Flujo Mensajes Update Router 6VPE$_1$

Frame Length 6VPE$_1$: 2460 *bytes*.

4.1.1.2 iBGP *Full Mesh* - *Router* 6VPE$_2$

Se obtienen los mensajes *Update* del *Router* 6VPE$_2$ con los demás *Routers* 6VPE en la figura 4.3.

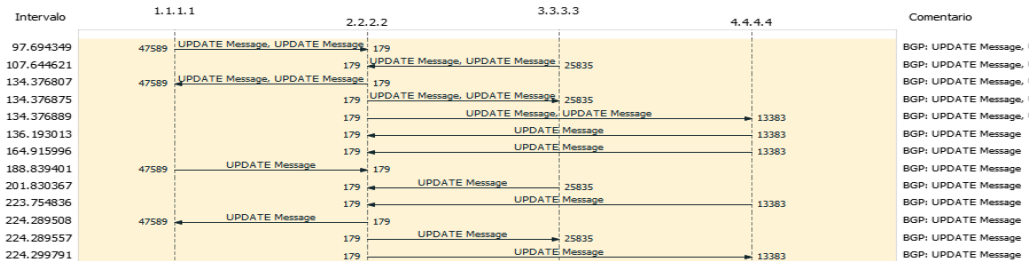


Figura 4.3 Escenario 1. Flujo Mensajes Update Router 6VPE$_2$

Frame Length 6VPE$_2$: 2460 *bytes*.

4.1.1.3 iBGP *Full Mesh* - *Router* 6VPE$_3$

Se obtienen los mensajes *Update* del *Router* 6VPE$_3$ con los demás *Routers* 6VPE en la figura 4.4.

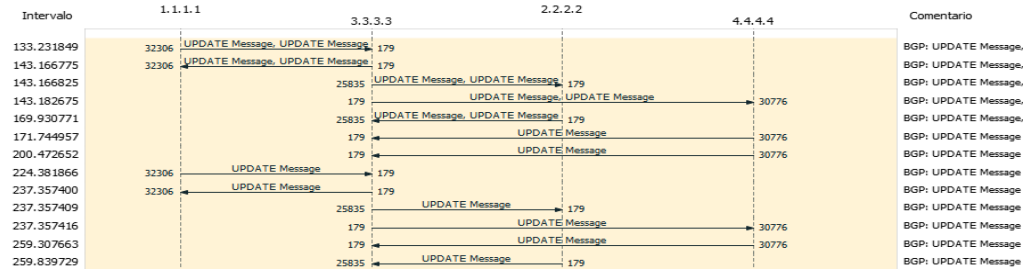


Figura 4.4 Escenario 1. Flujo Mensajes Update Router 6VPE$_3$

Frame Length 6VPE$_3$: 2460 *bytes*.

4.1.1.4 iBGP *Full Mesh* - *Router* 6VPE$_4$

Se obtienen los mensajes *Update* del *Router* 6VPE$_4$ con los demás *Routers* 6VPE en la figura 4.5.

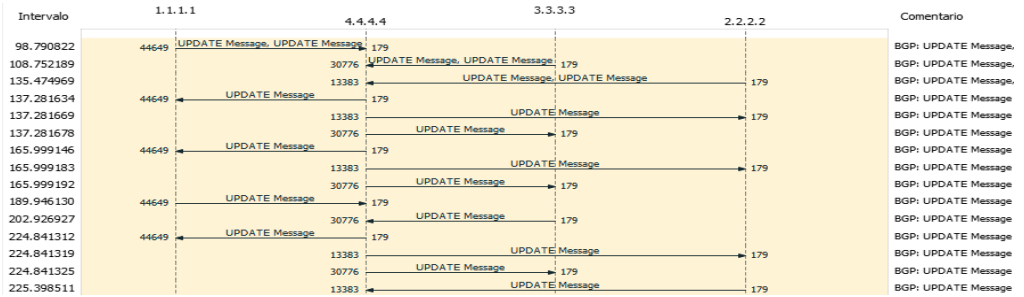


Figura 4.5 Escenario 1. Flujo Mensajes Update Router 6VPE$_4$

Frame Length 6VPE$_4$: 2460 *bytes*.

4.1.2 Mediciones Estadísticas TCP

4.1.2.1 *Round Trip Time*

Se obtienen valores para iRTT de los mensajes TCP que es el tiempo de ida y vuelta inicial o latencia inicial cuando se realiza el establecimiento de la conexión *Three-way Hanshake* entre los *Routers* 6VPE en el Escenario 1 (6 sesiones iBGP), para lo cual se toman los dos primeros mensajes tiempo de ida y vuelta (SYN y SYN-ACK) respecto a cada segmento y también se calcula el RTT promedio de las sesiones BGP durante toda la conexión.

- **RTT Sesión iBGP 6VPE$_1$ - 6VPE$_2$:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==2.2.2.2.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_1$ y 6VPE$_2$ es:
  iRTT: 51,414 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==1.1.1.1 and ip.dst==2.2.2.2.*
  Se observa el RTT de toda la captura en la figura 4.6, y su valor promedio es:
  RTT: 331,201 ms.

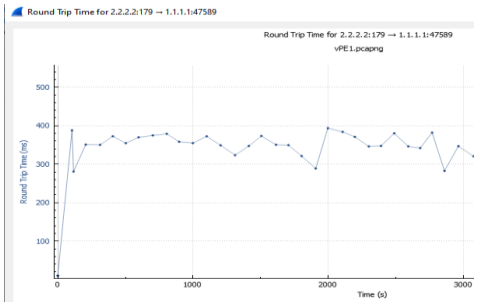


Figura 4.6 Escenario 1. RTT 6VPE$_1$ - 6VPE$_2$

- **RTT Sesión iBGP 6VPE$_1$ - 6VPE$_3$:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==3.3.3.3.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_1$ y 6VPE$_3$ es:
  iRTT: 213,951 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==1.1.1.1 and ip.dst==3.3.3.3.*
  Se observa el RTT de toda la captura en la figura 4.7, y su valor promedio es:
  RTT: 346,497 ms.

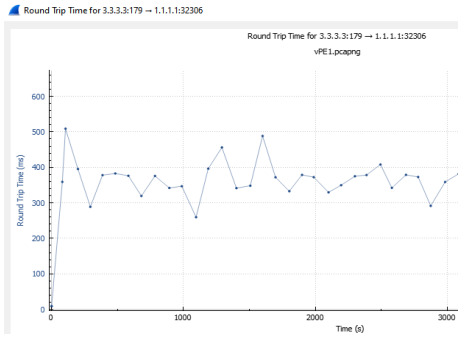


Figura 4.7 Escenario 1. RTT 6VPE$_1$ - 6VPE$_3$

- **RTT Sesión iBGP 6VPE$_1$ - 6VPE$_4$:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==4.4.4.4.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_1$ y 6VPE$_4$ es:
  iRTT: 74,39 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==1.1.1.1 and ip.dst==4.4.4.4.*
  Se observa el RTT de toda la captura en la figura 4.8, y su valor promedio es:
  RTT: 325,036 ms.

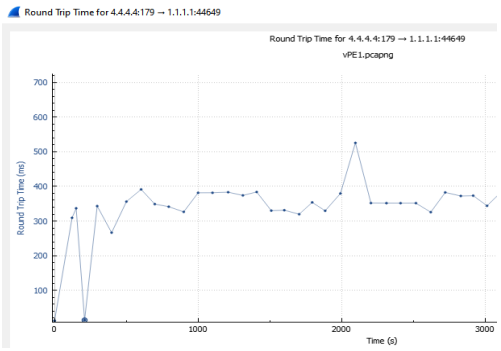


Figura 4.8 Escenario 1. RTT 6VPE$_1$ - 6VPE$_4$

- **RTT Sesión iBGP 6VPE$_2$ - 6VPE$_3$:**

- Filtro: *tcp and bgp and ip.src==2.2.2.2 and ip.dst==3.3.3.3.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_2$ y 6VPE$_3$ es:
  iRTT: 6,620 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==2.2.2.2 and ip.dst==3.3.3.3.*
  Se observa el RTT de toda la captura en la figura 4.9, y su valor promedio es:
  RTT: 326,357 ms.

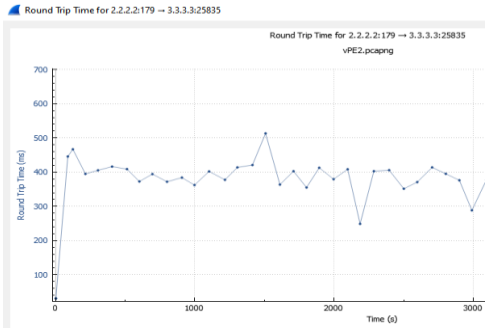


Figura 4.9 Escenario 1. RTT 6VPE$_2$ - 6VPE$_3$

- **RTT Sesión iBGP 6VPE$_2$ - 6VPE$_4$:**

- Filtro: *tcp and bgp and ip.src==2.2.2.2 and ip.dst==4.4.4.4.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_2$ y 6VPE$_4$ es:
  iRTT: 52,586 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==2.2.2.2 and ip.dst==4.4.4.4.*
  Se observa el RTT de toda la captura en la figura 4.10, y su valor promedio es:
  RTT: 326,580 ms.

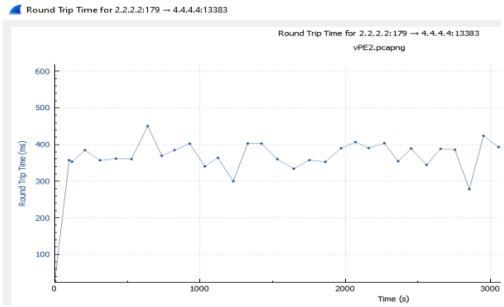


Figura 4.10 Escenario 1. RTT 6VPE$_2$ - 6VPE$_4$

- **RTT Sesión iBGP 6VPE$_3$ - 6VPE$_4$:**

- Filtro: *tcp and bgp and ip.src==3.3.3.3 and ip.dst==4.4.4.4.*
  El (iRTT) de la sesión iBGP entre los *Routers* 6VPE$_3$ y 6VPE$_4$ es:
  iRTT: 113,307 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==3.3.3.3 and ip.dst==4.4.4.4.*
  Se observa el RTT de toda la captura en la figura 4.11, y su valor promedio es:
  RTT: 334,130 ms.

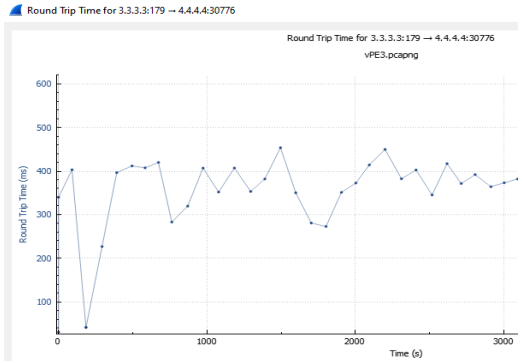


Figura 4.11 Escenario 1. RTT 6VPE$_3$ - 6VPE$_4$

Con los resultados obtenidos para el Escenario 1 *Full Mesh* sin aplicación de métodos se genera la tabla 4.1 de Análisis Estadístico para los valores de iRTT y RTT.

Tabla 4.1 Escenario 1. Análisis Estadístico RTT

	iRTT (ms)	**RTT (ms)**
Promedio	85,378	331,634
Desviación Estándar	71,921	8,053
Coeficiente de variación	842,388	24,283

4.1.2.2 *Frame Arrival Delay* y *Jitter*

Es la diferencia de tiempo entre la trama que llega y la última trama recibida en los *Routers* 6VPE para las 6 sesiones iBGP del Escenario 1 *Full Mesh*. Se aplica el filtro: *tcp.analysis.ack_rtt and ip.src==x.x.x.x and ip.dst==x.x.x.x* y el *Jitter* que es la variación del *Delay*, como se muestra en la tabla 4.2 con los valores promedios obtenidos para cada una de las sesiones iBGP.

Tabla 4.2 Escenario 1. Frame Arrival Delay y Jitter Sesiones iBGP

Conexión	*Frame Arrival Delay* (ms)	*Jitter* (ms)
$6VPE_1$ - $6VPE_2$	308,845	83,636
$6VPE_1$ - $6VPE_3$	310,771	118,058
$6VPE_1$ - $6VPE_4$	295,659	122,57
$6VPE_2$ - $6VPE_3$	274,535	119,249
$6VPE_2$ - $6VPE_4$	317,671	70,898
$6VPE_3$ - $6VPE_4$	307,317	87,262

Con los resultados obtenidos para el Escenario 1 sin métodos *Full Mesh* se genera la tabla 4.3 de Análisis estadístico para los valores de *Frame Arrival Delay* y *Jitter*.

Tabla 4.3 Escenario 1. Análisis Estadístico Frame Arrival Delay y Jitter

	Frame Arrival Delay (ms)	*Jitter* (ms)
Promedio	302,4	100,279
Desviación Estándar	15,4	22,282
Coeficiente de variación	50,93	222,204

4.2 Escenario 2: Topología con método *Route Reflector*

En la simulación del Escenario 2 aplicando el método *Route Reflector*, se configura un *Router Reflector* que se encarga de reflejar las rutas VPNv6 con los *Routers* 6VPE. Se establecen 4 sesiones iBGP, donde según lo planteado se realizan pruebas, mediciones y análisis de mensajes *Update* y sesiones TCP para un tiempo de 50 minutos, al igual que para todos los Escenarios.

4.2.1 Análisis iBGP *Route Reflector*

Se realiza la captura de los mensajes *Update* entre el *Router Reflector* RR_1 y los diferentes *Routers* 6VPE para el Escenario 2 con el filtro: *bgp.update.path_attributes,* se obtiene el tamaño de mensajes *Update (Frame Length)* de las sesiones BGP como se muestra en la figura 4.12, y se generan las gráficas de flujo.

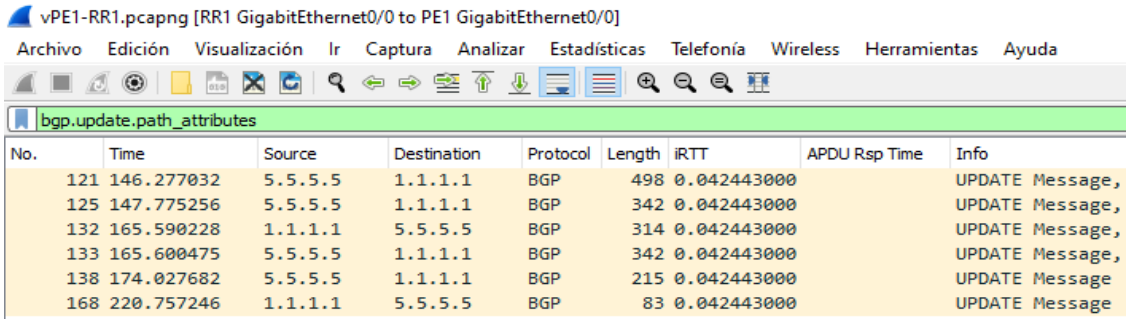


Figura 4.12 Escenario 2. Mensajes Update Router $6VPE_1$

4.2.1.1 iBGP *Route Reflector* - *Router* 6VPE$_1$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_1$ con el *Router Reflector* RR$_1$ en la figura 4.13.

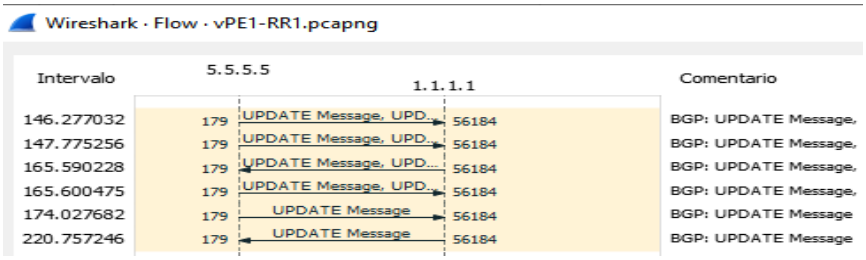


Figura 4.13 Escenario 2. Flujo Mensajes Update Router 6VPE$_1$

Frame Length 6VPE$_1$: 1794 *bytes*.

4.2.1.2 iBGP *Route Reflector* - *Router* 6VPE$_2$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_2$ con el *Router Reflector* RR$_1$ en la figura 4.14.

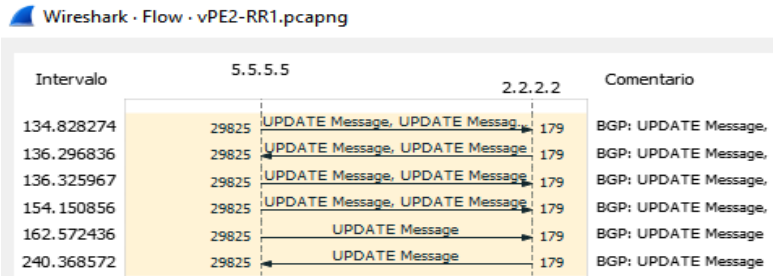


Figura 4.14 Escenario 2. Flujo Mensajes Update Router 6VPE$_2$

Frame Length 6VPE$_2$: 1802 *bytes*.

4.2.1.3 iBGP *Route Reflector* - *Router* 6VPE$_3$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_3$ con el *Router Reflector* RR$_1$ en la figura 4.15.

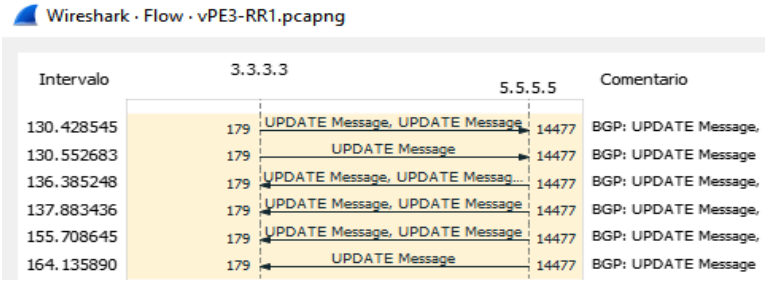


Figura 4.15 Escenario 2. Flujo Mensajes Update Router 6VPE$_3$

Frame Length 6VPE$_3$: 1794 *bytes*.

4.2.1.4 iBGP *Route Reflector* - *Router* 6VPE$_4$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_4$ con el *Router Reflector* RR$_1$ en la figura 4.16.

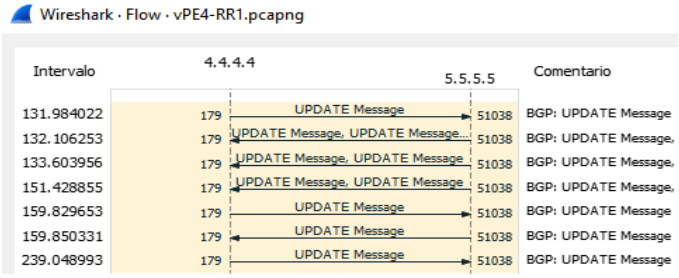


Figura 4.16 Escenario 2. Flujo Mensajes Update Router 6VPE$_4$

Frame Length 6VPE$_4$: 1860 *bytes*.

4.2.2 Mediciones Estadísticas TCP

4.2.2.1 *Round Trip Time*

Se obtienen valores para iRTT de los mensajes TCP que es el tiempo de ida y vuelta inicial o latencia inicial cuando se realiza el establecimiento de la conexión *Three-way Hanshake* entre los *Routers* 6VPE y el *Router Reflector* RR$_1$ en el Escenario 2 (4 sesiones iBGP), para lo cual se toman los dos primeros mensajes tiempo de ida y vuelta (SYN y SYN-ACK) respecto a cada segmento y también se calcula el RTT promedio de las sesiones BGP durante toda la conexión.

- **RTT Sesión iBGP 6VPE$_1$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_1$ y RR$_1$ es:
  iRTT: 42,443 ms.

- Filtro: *tcp.analysis.ack_rtt and ip.src==1.1.1.1 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.17, y su valor promedio es:
  RTT: 216,935 ms.

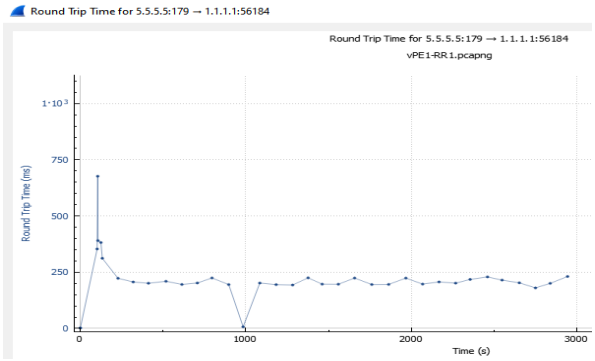


Figura 4.17 Escenario 2. RTT 6VPE$_1$ - RR$_1$

- **RTT Sesión iBGP 6VPE$_2$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==2.2.2.2 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_2$ y RR$_1$ es:
  iRTT: 53,673 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==2.2.2.2 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.18, y su valor promedio es:
  RTT: 324,363 ms.

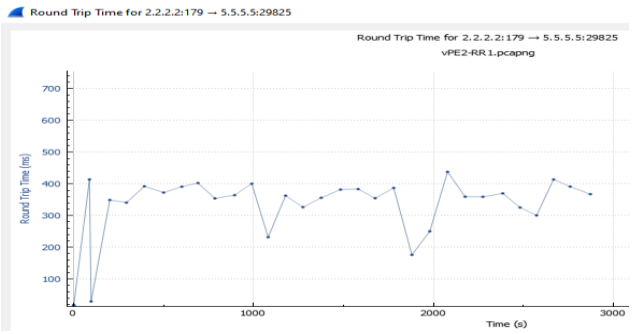


Figura 4.18 Escenario 2. RTT 6VPE$_2$ - RR$_1$

- **RTT Sesión iBGP 6VPE$_3$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==3.3.3.3 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_3$ y RR$_1$ es:
  iRTT: 57,854 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==3.3.3.3 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.19, y su valor promedio es:
  RTT: 335,561 ms.

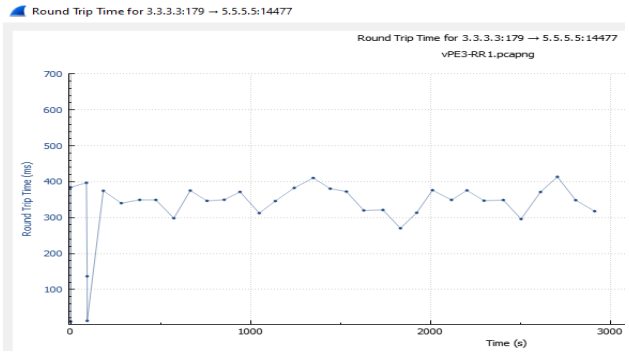


Figura 4.19 Escenario 2. RTT 6VPE$_3$ - RR$_1$

- **RTT Sesión iBGP 6VPE$_4$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==4.4.4.4 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_4$ y RR$_1$ es:
  iRTT: 31,092 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==4.4.4.4 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.20, y su valor promedio es:
  RTT: 321,506 ms.

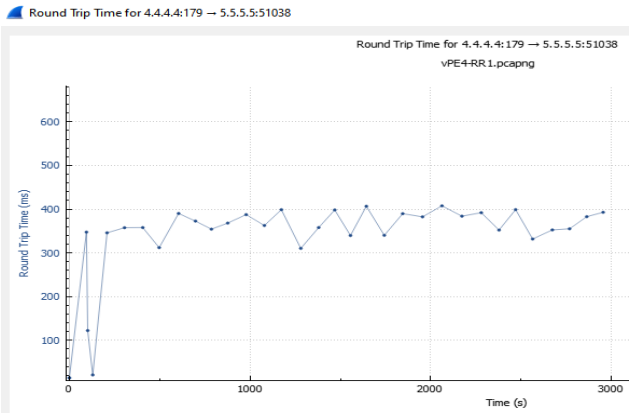


Figura 4.20 Escenario 2. RTT 6VPE$_4$ - RR$_1$

Con los resultados obtenidos para el Escenario 2 con método *Route Reflector* se genera la tabla 4.4 de Análisis Estadístico para los valores de iRTT y RTT.

Tabla 4.4 Escenario 2. Análisis Estadístico RTT

	iRTT(ms)	**RTT(ms)**
Promedio	46,216	299,591
Desviación Estándar	12,049	55,437
Coeficiente de variación	260,719	185,041

4.2.2.2 *Frame Arrival Delay* y *Jitter*

Es la diferencia de tiempo entre la trama que llega y la última trama recibida en los *Routers* 6VPE para las 4 sesiones iBGP del Escenario 2 *Route Reflector*, se aplica el filtro: *tcp and bgp and ip.src==x.x.x.x and ip.dst==x.x.x.x* y el *Jitter* que es la variación del *Delay*, como se muestra en la tabla 4.5 con los valores promedios obtenidos para cada una de las sesiones iBGP.

Tabla 4.5 Escenario 2. Frame Arrival Delay y Jitter Sesiones iBGP

Conexión	***Frame Arrival Delay* (ms)**	***Jitter* (ms)**
6VPE$_1$ - RR$_1$	201,992	61
6VPE$_2$ - RR$_1$	304,037	90,582
6VPE$_3$ - RR$_1$	315,171	96,29
6VPE$_4$ - RR$_1$	278,842	145,894

Con los resultados obtenidos para el Escenario 2 con método *Route Reflector* se genera la tabla 4.6 de Análisis Estadístico para los valores de *Frame Arrival Delay* y *Jitter*.

Tabla 4.6 Escenario 2. Análisis Estadístico Frame Arrival Delay y Jitter

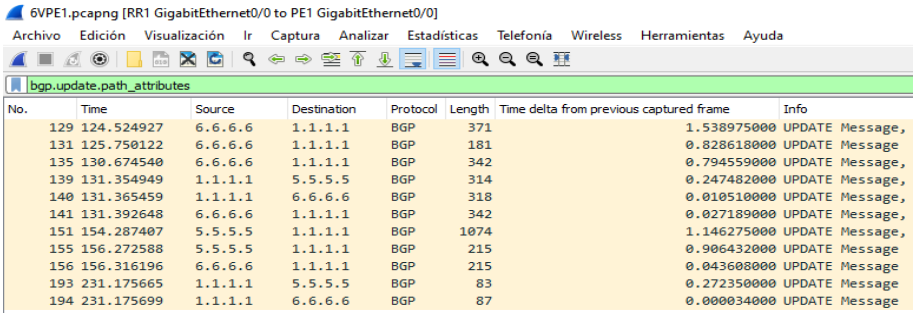
	***Frame Arrival Delay* (ms)**	***Jitter* (ms)**
Promedio	275,01	98,442
Desviación Estándar	50,996	35,213
Coeficiente de variación	185,432	357,709

4.3 Escenario 3: Topología con método *Cluster Route Reflector*

En la simulación del Escenario 3 aplicando el método de *Cluster Route Reflector*, con dos *Routers Reflector* que se encargan de reflejar las rutas VPNv6 con los *Routers* 6VPE, se establecen 8 sesiones iBGP, donde según lo planteado se realizan pruebas, mediciones y análisis de mensajes *Update* y sesiones TCP para un tiempo de 50 minutos al igual que para todos los Escenarios.

4.3.1 Análisis iBGP *Cluster Route Reflector*

Se realiza la captura de los mensajes *Update* entre los *Routers Reflectors* RR_1 y RR_2 con los diferentes *Routers* 6VPE para el Escenario 3 con el filtro: *bgp.update.path_attributes,* se obtiene el tamaño de los mensajes *Update* (*Frame Length*) de las sesiones BGP como se muestra en la figura 4.21, y se generan las gráficas de flujo.



No.	Time	Source	Destination	Protocol	Length	Time delta from previous captured frame	Info
129	124.524927	6.6.6.6	1.1.1.1	BGP	371	1.538975000	UPDATE Message,
131	125.750122	6.6.6.6	1.1.1.1	BGP	181	0.828618000	UPDATE Message
135	130.674540	6.6.6.6	1.1.1.1	BGP	342	0.794559000	UPDATE Message,
139	131.354949	1.1.1.1	5.5.5.5	BGP	314	0.247482000	UPDATE Message,
140	131.365459	1.1.1.1	6.6.6.6	BGP	318	0.010510000	UPDATE Message,
141	131.392648	6.6.6.6	1.1.1.1	BGP	342	0.027189000	UPDATE Message,
151	154.287407	5.5.5.5	1.1.1.1	BGP	1074	1.146275000	UPDATE Message,
155	156.272588	5.5.5.5	1.1.1.1	BGP	215	0.906432000	UPDATE Message
156	156.316196	6.6.6.6	1.1.1.1	BGP	215	0.043608000	UPDATE Message
193	231.175665	1.1.1.1	5.5.5.5	BGP	83	0.272350000	UPDATE Message
194	231.175699	1.1.1.1	6.6.6.6	BGP	87	0.000034000	UPDATE Message

Figura 4.21 Escenario 3. Mensajes Update Router 6VPE$_1$

4.3.1.1 iBGP *Cluster Route Reflector* - *Router* 6VPE$_1$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* $6VPE_1$ con los *Routers Reflector* RR_1 y RR_2 en la figura 4.22.

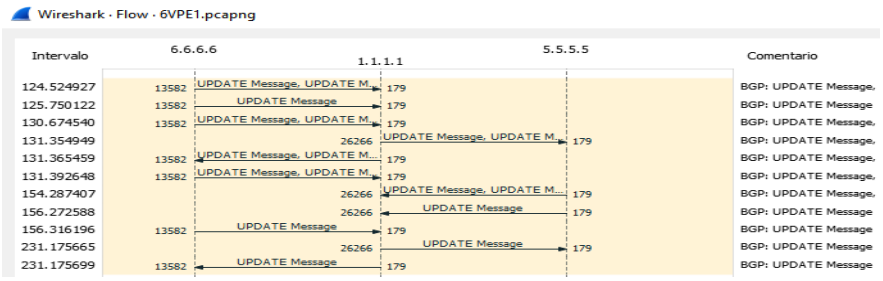


Figura 4.22 Escenario 3. Flujo Mensajes Update Router 6VPE$_1$

Frame Length $6VPE_1$: 3542 *bytes*.

4.3.1.2 iBGP *Cluster Route Reflector* - *Router* 6VPE$_2$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* $6VPE_2$ con los *Routers Reflector* RR_1 y RR_2 en la figura 4.23.

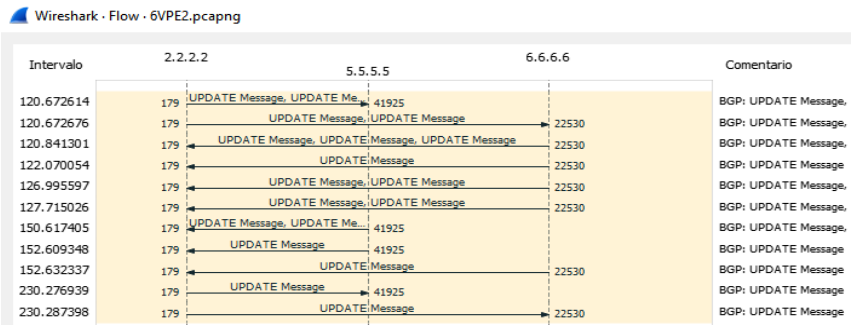


Figura 4.23 Escenario 3. Flujo Mensajes Update Router 6VPE$_2$

Frame Length 6VPE$_2$: 3542 *bytes*.

4.3.1.3 iBGP *Cluster Route Reflector - Router* 6VPE$_3$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_3$ con los *Routers Reflector* RR$_1$ y RR$_2$ en la figura 4.24.

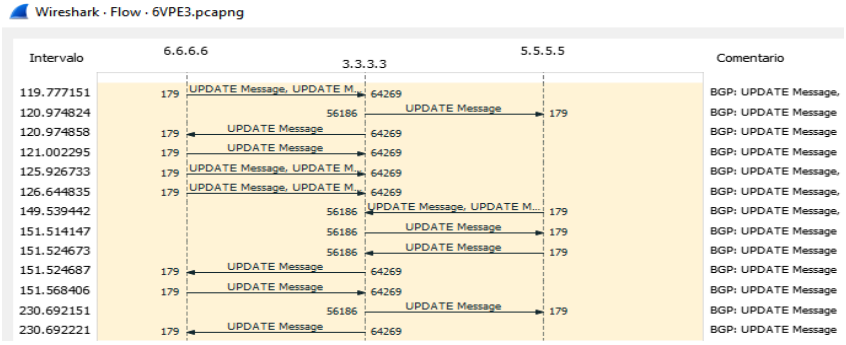


Figura 4.24 Escenario 3. Flujo Mensajes Update Router 6VPE$_3$

Frame Length 6VPE$_3$: 3654 *bytes*.

4.3.1.4 iBGP *Cluster Route Reflector - Router* 6VPE$_4$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_4$ con los *Routers Reflector* RR$_1$ y RR$_2$ en la figura 4.25.

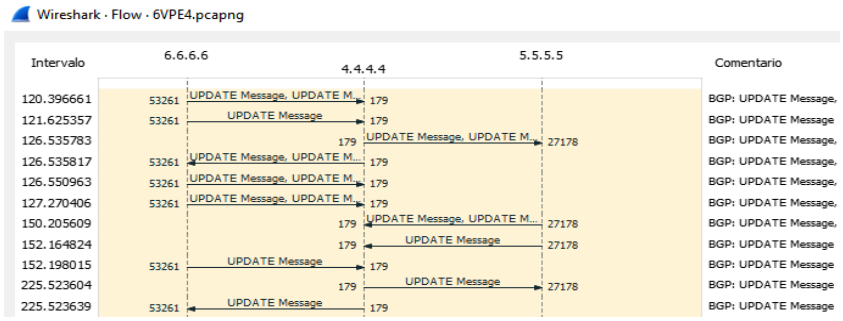


Figura 4.25 Escenario 3. Flujo Mensajes Update Router 6VPE$_4$

Frame Length 6VPE$_4$: 3542 *bytes*.

4.3.2 Mediciones Estadísticas TCP

4.3.2.1 *Round Trip Time*

Se obtienen valores para iRTT de los mensajes TCP que es el tiempo de ida y vuelta inicial o latencia inicial cuando se realiza el establecimiento de la conexión *Three-way Hanshake* entre los *Routers* 6VPE y los *Routers* RR_1 y RR_2 en el Escenario 3 (8 sesiones iBGP), para lo cual se toman los dos primeros mensajes tiempo de ida y vuelta (SYN y SYN-ACK) respecto a cada segmento y también se calcula el RTT promedio de las sesiones BGP durante toda la conexión.

- **RTT Sesión iBGP $6VPE_1$ - RR_1:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* $6VPE_1$ y RR_1 es:
  iRTT: 41,746 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==1.1.1.1 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.26, y su valor promedio es:
  RTT: 310,475 ms.

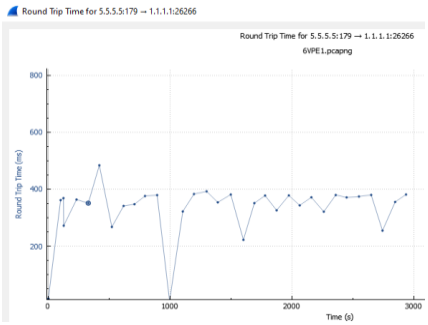


Figura 4.26 Escenario 3. RTT $6VPE_1$ - RR_1

- **RTT Sesión iBGP $6VPE_1$ - RR_2:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==6.6.6.6.*
  El iRTT de la sesión iBGP entre los *Routers* $6VPE_1$ y RR_2 es:
  iRTT: 31,503 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==1.1.1.1 and ip.dst==6.6.6.6.*
  Se observa el RTT de toda la captura en la figura 4.27, y su valor promedio es:
  RTT: 325,950 ms.

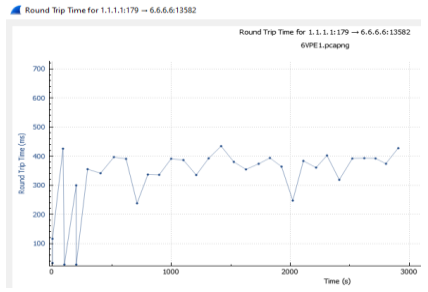


Figura 4.27 Escenario 3. RTT $6VPE_1$ - RR_2

- **RTT Sesión iBGP 6VPE$_2$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==2.2.2.2 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_2$ y RR$_1$ es:
  iRTT: 91,756 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==2.2.2.2 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.28, y su valor promedio es:
  RTT: 310,668 ms.

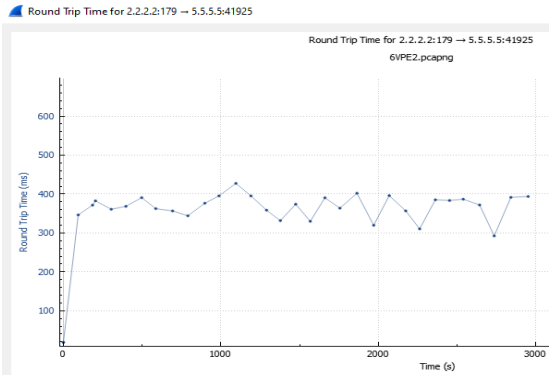


Figura 4.28 Escenario 3. RTT 6VPE$_2$ - RR$_1$

- **RTT Sesión iBGP 6VPE$_2$ - RR$_2$:**

- Filtro: *tcp and bgp and ip.src==2.2.2.2 and ip.dst==6.6.6.6.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_2$ y RR$_2$ es:
  iRTT: 52,3 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==2.2.2.2 and ip.dst==6.6.6.6.*
  Se observa el RTT de toda la captura en la figura 4.29, y su valor promedio es:
  RTT: 317,933 ms.

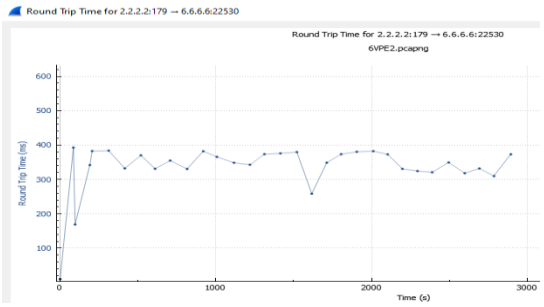


Figura 4.29 Escenario 3. RTT 6VPE$_2$ - RR$_2$

- **RTT Sesión iBGP 6VPE$_3$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==3.3.3.3 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_3$ y RR$_1$ es:
  iRTT: 31,521 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==3.3.3.3 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.30, y su valor promedio es:
  RTT: 322,241 ms.

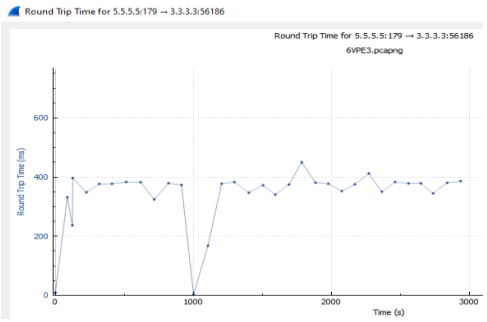


Figura 4.30 Escenario 3. RTT 6VPE$_3$ - RR$_1$

- **RTT Sesión iBGP 6VPE$_3$ - RR$_2$:**

- Filtro: *tcp and bgp and ip.src==3.3.3.3 and ip.dst==6.6.6.6.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_3$ y RR$_2$ es:
  iRTT: 51,903 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==3.3.3.3 and ip.dst==6.6.6.6.*
  Se observa el RTT de toda la captura en la figura 4.31, y su valor promedio es:
  RTT: 319,657 ms.

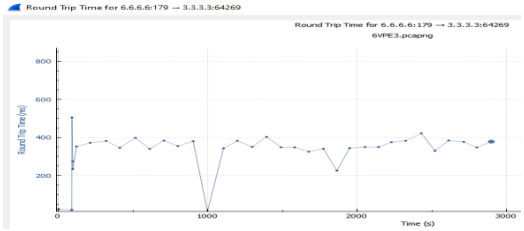


Figura 4.31 Escenario 3. RTT 6VPE$_3$ - RR$_2$

- **RTT Sesión iBGP 6VPE$_4$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==4.4.4.4 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_4$ y RR$_1$ es:
  iRTT: 41,699 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==4.4.4.4 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.32, y su valor promedio es:
  RTT: 332,975 ms.

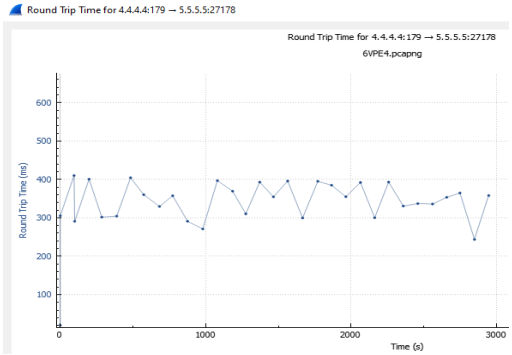


Figura 4.32 Escenario 3. RTT 6VPE$_4$ - RR$_1$

- **RTT Sesión iBGP 6VPE$_4$ - RR$_2$:**

- Filtro: *tcp and bgp and ip.src==4.4.4.4 and ip.dst==6.6.6.6.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_4$ y RR$_2$ es:
  iRTT: 62,063 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==4.4.4.4 and ip.dst==6.6.6.6.*
  Se observa el RTT de toda la captura en la figura 4.33, y su valor promedio es:
  RTT: 314,882 ms.

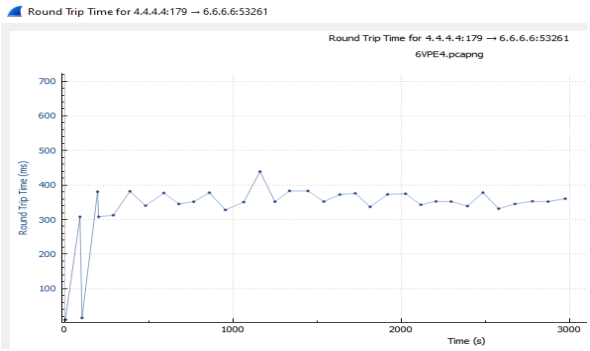


Figura 4.33 Escenario 3. RTT 6VPE$_4$ - RR$_2$

Con los resultados obtenidos para el Escenario 3 con método *Cluster Route Reflector* se genera la tabla 4.7 de análisis estadístico para los valores de iRTT y RTT.

Tabla 4.7 Escenario 3. Análisis Estadístico RTT

	iRTT (ms)	RTT (ms)
Promedio	50,561	319,348
Desviación Estándar	19,698	7,688
Coeficiente de variación	389,589	24,074

4.3.2.2 *Frame Arrival Delay* y *Jitter*

Es la diferencia de tiempo entre la trama que llega y la última trama recibida en los *Routers* 6VPE para las 8 sesiones iBGP del Escenario 3 *Cluster Route Reflector*, se aplica el filtro: *tcp and bgp and ip.src==x.x.x.x and ip.dst==x.x.x.x* y el *Jitter* que es la variación del *Delay*, como se muestra en la tabla 4.8 con los valores promedios obtenidos para cada una de las sesiones iBGP.

Tabla 4.8 Escenario 3. Frame Arrival Delay y Jitter Sesiones iBGP

Conexión	*Frame Arrival Delay* (ms)	*Jitter* (ms)
6VPE$_1$ - RR$_1$	280,499	114,193
6VPE$_1$ - RR$_2$	281,161	125,58
6VPE$_2$ - RR$_1$	300,164	94,619
6VPE$_2$ - RR$_2$	285,214	108,1852
6VPE$_3$ - RR$_1$	314,534	77,454
6VPE$_3$ - RR$_2$	287,569	111,171
6VPE$_4$ - RR$_1$	309,141	99,563
6VPE$_4$ - RR$_2$	300,232	94,431

Con los resultados obtenidos para el Escenario 3 con método *Cluster Route Reflector* se genera la tabla 4.9 de Análisis Estadístico para los valores de *Frame Arrival Delay* y *Jitter*.

Tabla 4.9 Escenario 3. Análisis Estadístico Frame Arrival Delay y Jitter

	***Frame Arrival Delay* (ms)**	***Jitter* (ms)**
Promedio	294,814	103,15
Desviación Estándar	13,028	14,816
Coeficiente de variación	44,19	143,64

4.4 Escenario 4: Topología con método *Confederations BGP*

En la simulación del Escenario 4 aplicando el método de *Confederations BGP*, se subdivide el AS 65100 en dos subsistemas (AS 65101 y AS 65102). Se establecen 3 sesiones BGP, 2 sesiones iBGP entre 6VPE$_1$ - 6VPE$_3$, 6VPE$_2$ - 6VPE$_4$ y una sesión eBGP entre 6VPE$_1$ - 6VPE$_2$ que refleja las rutas VPNv6 entre los subsistemas. Según lo planteado se realizan pruebas, mediciones y análisis de mensajes *Update* y sesiones TCP para un tiempo de 50 minutos al igual que para todos los Escenarios.

4.4.1 Análisis sesiones BGP en *Confederations BGP*

Se realiza la captura de mensajes *Update* de las 2 sesiones iBGP entre 6VPE$_1$ - 6VPE$_3$, 6VPE$_2$ - 6VPE$_4$ y la sesión eBGP entre 6VPE$_1$ - 6VPE$_2$ para el Escenario 4 con el filtro: *bgp.update.path_attributes,* se obtiene el tamaño de los mensajes *Update* (*Frame Length*) de las sesiones BGP como se muestra en la figura 4.34, y se generan las gráficas de flujo.

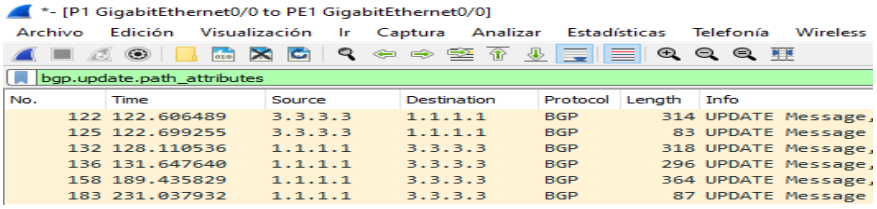


Figura 4.34 Escenario 4. Mensajes Update Router 6VPE$_1$

4.4.1.1 iBGP *Confederations BGP* - *Router* 6VPE$_1$ - 6VPE$_3$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_1$ con el 6VPE$_3$ en la figura 4.35.

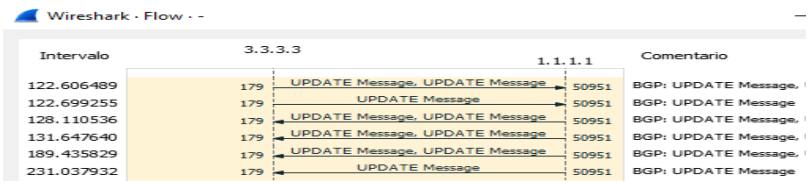


Figura 4.35 Escenario 4. Flujo Mensajes Update Router 6VPE$_1$ - 6VPE$_3$

Frame Length 6VPE$_1$ - 6VPE$_3$: 1462 *bytes*.

4.4.1.2 eBGP *Confederations BGP* - *Router* 6VPE$_1$ - 6VPE$_2$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_1$ con el 6VPE$_2$ en la figura 4.36.

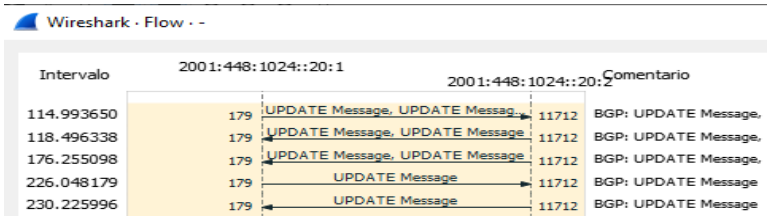


Figura 4.36 Escenario 4. Flujo Mensajes Update Router 6VPE$_1$ - 6VPE$_2$

Frame Length 6VPE$_1$ - 6VPE$_2$: 1708 *bytes*.

4.4.1.3 iBGP *Confederations BGP* - *Router* 6VPE$_2$ - 6VPE$_4$

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE$_2$ con el 6VPE$_4$ en la figura 4.37.

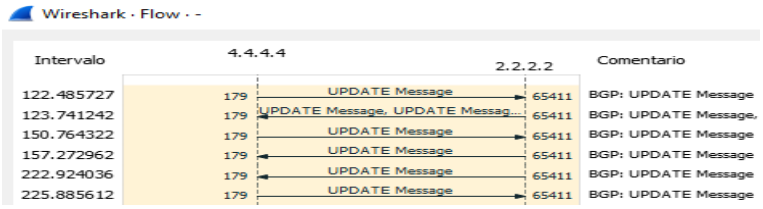


Figura 4.37 Escenario 4. Flujo Mensajes Update Router 6VPE$_2$ - 6VPE$_4$

Frame Length 6VPE$_2$ - 6VPE$_4$: 1458 *bytes*.

4.4.2 Mediciones Estadísticas TCP

4.4.2.1 *Round Trip Time*

Se obtienen valores para iRTT de los mensajes TCP que es el tiempo de ida y vuelta inicial o latencia inicial cuando se realiza el establecimiento de la conexión *Three-way Hanshake* entre los *Routers* 6VPE en el Escenario 4 (2 sesiones iBGP y 1 sesión eBGP), para lo cual se toman los dos primeros mensajes tiempo de ida y vuelta (SYN y SYN-ACK) respecto a cada segmento y también se calcula el RTT promedio de las sesiones BGP durante toda la conexión.

- **RTT Sesión iBGP 6VPE$_1$ - 6VPE$_3$:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==3.3.3.3.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_1$ y RR$_1$ es:
  iRTT: 61,87 ms.
- Filtro: *tcp.analysis.acks_frame and ip.src==1.1.1.1 and ip.dst==3.3.3.3.*
  Se observa el RTT de toda la captura en la figura 4.38, y su valor promedio es:
  RTT: 316,367 ms.

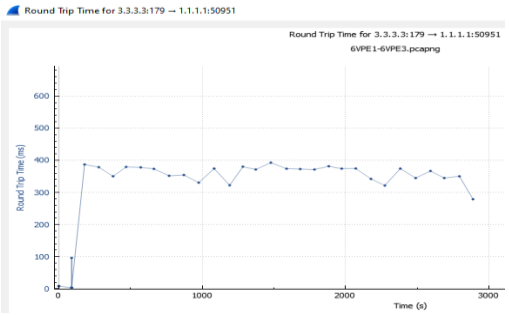


Figura 4.38 Escenario 4. RTT 6VPE$_1$ - 6VPE$_3$

- **RTT Sesión eBGP 6VPE$_1$ - 6VPE$_2$:**

- Filtro: *tcp and bgp and ipv6.src==2001:448:1024::20:1 and ipv6.dst==2001:448:1024::20:2.*
  El iRTT de la sesión eBGP entre los *Routers* 6VPE$_1$ y 6VPE$_2$ es:
  iRTT: 32,195 ms.
- Filtro: *tcp.analysis.ack_rtt and ipv6.src==2001:448:1024::20:1 and ipv6.dst==2001:448:1024::20:2.*
  Se observa el RTT de toda la captura en la figura 4.39, y su valor promedio es:
  RTT: 328,314 ms.

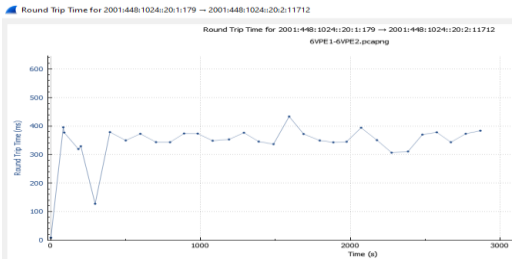


Figura 4.39 Escenario 4. RTT 6VPE$_1$ - 6VPE$_2$

- **RTT Sesión iBGP 6VPE$_2$ - 6VPE$_4$:**

- Filtro: *tcp and bgp and ip.src==2.2.2.2 and ip.dst==4.4.4.4.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_2$ y 6VPE$_4$ es:
  iRTT: 81,107 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==2.2.2.2 and ip.dst==4.4.4.4.*
  Se observa el RTT de toda la captura en la figura 4.40, y su valor promedio es:
  RTT: 312,886 ms.

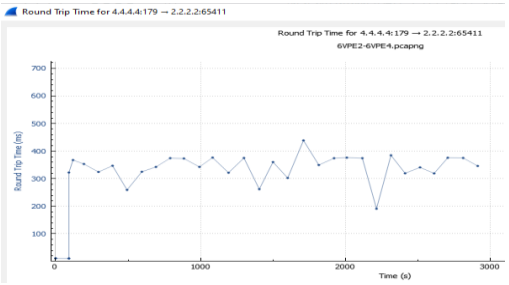


Figura 4.40 Escenario 4. RTT 6VPE$_2$ - 6VPE$_4$

Con los resultados obtenidos para el Escenario 4 con método *Confederations BGP* se genera la tabla 4.10 de Análisis Estadístico para los valores de iRTT y RTT.

Tabla 4.10 Escenario 4. Análisis Estadístico RTT

	iRTT (ms)	**RTT (ms)**
Promedio	58,39	319,189
Desviación Estándar	24,641	8,092
Coeficiente de variación	422,001	25,351

4.4.2.2 *Frame Arrival Delay* y *Jitter*

Es la diferencia de tiempo entre la trama que llega y la última trama recibida en los *Routers* 6VPE para las sesiones BGP del Escenario 4 *Confederations BGP*, se aplica el filtro: *tcp and bgp and ip.src==x.x.x.x and ip.dst==x.x.x.x* y el *Jitter* que es la variación del *Delay*, como se muestra en la tabla 4.11 con los valores promedios obtenidos para cada una de las sesiones BGP.

Tabla 4.11 Escenario 4. Frame Arrival Delay y Jitter Sesiones BGP

Conexión	***Frame Arrival Delay* (ms)**	***Jitter* (ms)**
$6VPE_1$ - $6VPE_3$	294,14	70,41
$6VPE_1$ - $6VPE_2$	319,487	61,536
$6VPE_2$ - $6VPE_4$	289,042	77,538

Con los resultados obtenidos para el Escenario 4 con método *Confederations BGP* se genera la tabla 4.12 de Análisis Estadístico para los valores de *Frame Arrival Delay* y *Jitter*.

Tabla 4.12 Escenario 4. Análisis Estadístico Frame Arrival Delay y Jitter

	***Frame Arrival Delay* (ms)**	***Jitter* (ms)**
Promedio	300,889	69,828
Desviación Estándar	16,306	8,017
Coeficiente de variación	54,192	114,806

4.5 Escenario 5: Topología con métodos *Confederations BGP* y *Route Reflector*

En la simulación del Escenario 5 aplicando los métodos *Confederations BGP y Route Reflector*, se subdivide el AS 65100 en dos subsistemas (AS 65101 y AS 65102) igual que en el Escenario 4, pero se adiciona el método *Route Reflector* en los dos subsistemas donde se configuran el *Router Reflector* RR_1 que reenvía las rutas VPNv6 para el subsistema 65101 y el *Router Reflector* RR_2 que reenvía las rutas VPNv6 para el subsistema 65102. Se levanta una sesión eBGP entre $6VPE_1$ con $6VPE_2$ que refleja las rutas VPNv6 entre los subsistemas, 4 sesiones iBGP entre RR_1 con $6VPE_1$, RR_1 con $6VPE_3$, RR_2 con $6VPE_2$ y RR_2 con $6VPE_4$. Según lo planteado se realizan pruebas, mediciones y análisis de mensajes *Update* y sesiones TCP para un tiempo de 50 minutos al igual que para todos los Escenarios.

4.5.1 Análisis Sesiones BGP en *Confederations BGP* y *Route Reflector*

Se realiza la captura de mensajes *Update* de las 4 sesiones iBGP entre RR_1 con $6VPE_1$, RR_1 con $6VPE_3$, RR_2 con $6VPE_2$, RR_2 con $6VPE_4$ y la sesión eBGP entre $6VPE_1$ con $6VPE_2$, para el Escenario 5 con el filtro: *bgp.update.path_attributes,* se obtiene el tamaño de los mensajes *Update* (*Frame Length*) de las sesiones BGP como se muestra en la figura 4.41, y se generan las gráficas de flujo.

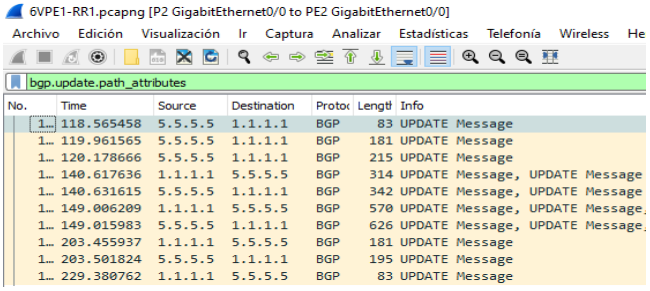


Figura 4.41 Escenario 5. Mensajes Update Router $6VPE_1$

4.5.1.1 iBGP *Confederations BGP* - *Router* $6VPE_1$ - RR_1

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* $6VPE_1$ con el RR_1 en la figura 4.42.

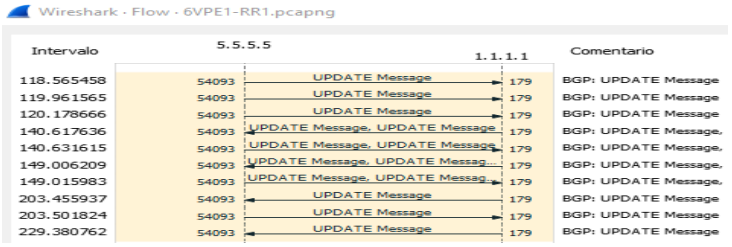


Figura 4.42 Escenario 5. Flujo Mensajes Update Router $6VPE_1$ - RR_1

Frame Length $6VPE_1$ - RR_1: 2790 *bytes*.

4.5.1.2 iBGP *Confederations BGP* - *Router* $6VPE_3$ - RR_1

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* $6VPE_3$ con el RR_1 en la figura 4.43.

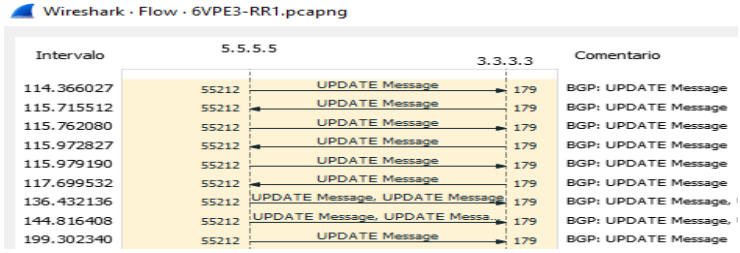


Figura 4.43 Escenario 5. Flujo Mensajes Update Router $6VPE_3$ - RR_1

Frame Length $6VPE_3$ - RR_1: 2093 *bytes*.

4.5.1.3 eBGP *Confederations BGP* - *Router* 6VPE1 - 6VPE2

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE1 con el 6VPE2 en la figura 4.44.

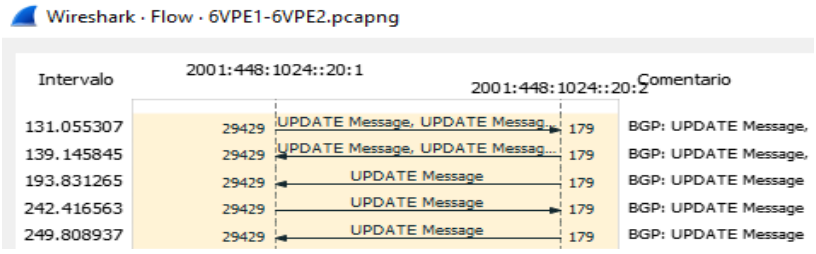


Figura 4.44 Escenario 5. Flujo Mensajes Update Router 6VPE1 - 6VPE2

Frame Length 6VPE1 - 6VPE2: 1831 *bytes*.

4.5.1.4 iBGP *Confederations BGP* - *Router* 6VPE2 - RR2

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE2 con el RR2 en la figura 4.45.

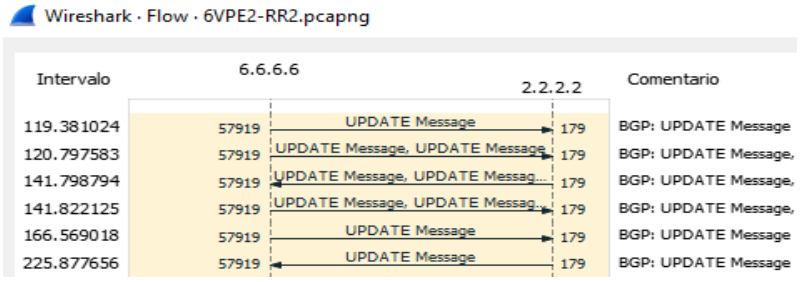


Figura 4.45 Escenario 5. Flujo Mensajes Update Router 6VPE2 - RR2

Frame Length 6VPE2 - RR2: 2469 *bytes.*

4.5.1.5 iBGP *Confederations BGP* - *Router* 6VPE4 - RR2

Se obtiene la gráfica de Flujo de mensajes *Update* del *Router* 6VPE4 con el RR2 en la figura 4.46.

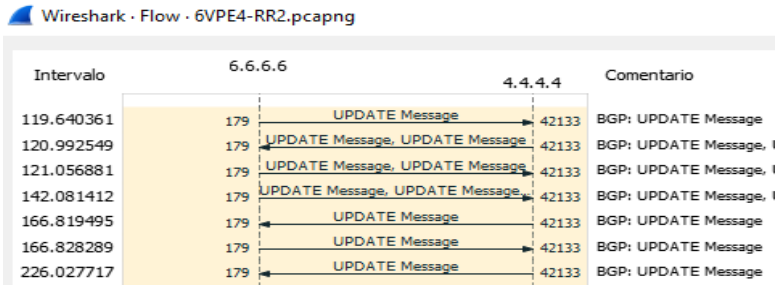


Figura 4.46 Escenario 5. Flujo Mensajes Update Router 6VPE4 - RR2

Frame Length 6VPE4 - RR2: 2072 *bytes*.

4.5.2 Mediciones Estadísticas TCP

4.5.2.1 *Round Trip Time*

Se obtienen valores para iRTT de los mensajes TCP que es el tiempo de ida y vuelta inicial o latencia inicial cuando se realiza el establecimiento de la conexión *Three-way Hanshake* entre los *Routers* 6VPE en el Escenario 5 (4 sesiones iBGP y 1 sesión eBGP), para lo cual se toman los dos primeros mensajes tiempo de ida y vuelta (SYN y SYN-ACK) respecto a cada segmento y también se calcula el RTT promedio de las sesiones BGP durante toda la conexión.

- **RTT Sesión iBGP 6VPE$_1$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==1.1.1.1 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_1$ y RR$_1$ es:
  iRTT: 30,928 ms.
- Filtro: *tcp.analysis.acks_frame and ip.src==1.1.1.1 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.47, y su valor promedio es:
  RTT: 322,154 ms.

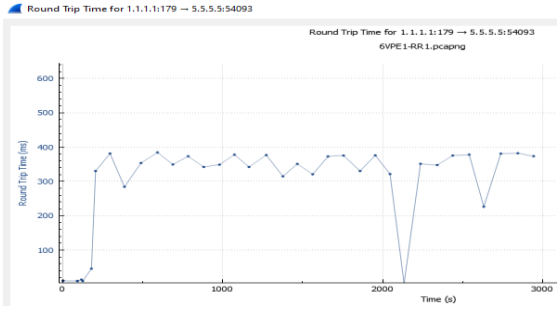


Figura 4.47 Escenario 5. RTT 6VPE$_1$ - RR$_1$

- **RTT Sesión iBGP 6VPE$_3$ - RR$_1$:**

- Filtro: *tcp and bgp and ip.src==3.3.3.3 and ip.dst==5.5.5.5.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_3$ y RR$_1$ es:
  iRTT: 31,438 ms.
- Filtro: *tcp.analysis.acks_frame and ip.src==3.3.3.3 and ip.dst==5.5.5.5.*
  Se observa el RTT de toda la captura en la figura 4.48, y su valor promedio es:
  RTT: 329,905 ms.

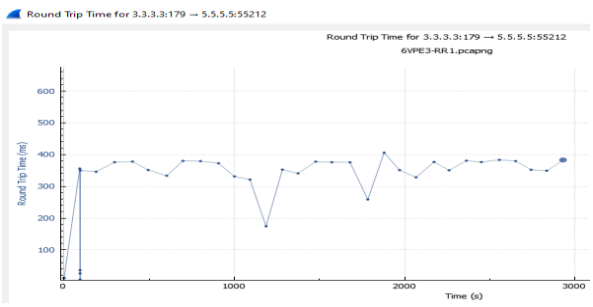


Figura 4.48 Escenario 5. RTT 6VPE$_3$ - RR$_1$

- **RTT Sesión eBGP 6VPE$_1$ - 6VPE$_2$:**

- Filtro: *tcp and bgp and ipv6.src==2001:448:1024::20:1 and ipv6.dst==2001:448:1024::20:2.*
  El iRTT de la sesión eBGP entre los *Routers* 6VPE$_1$ y 6VPE$_2$ es:
  iRTT: 30,335 ms.
- Filtro: *tcp.analysis.ack_rtt and ipv6.src==2001:448:1024::20:1 and ipv6.dst==2001:448:1024::20:2.*
  Se observa el RTT de toda la captura en la figura 4.49, y su valor promedio es:
  RTT: 331,807 ms.

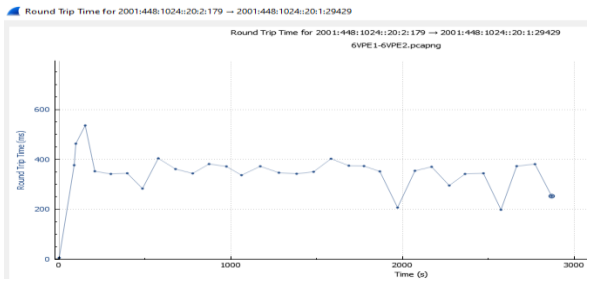


Figura 4.49 Escenario 5. RTT 6VPE$_1$ - 6VPE$_2$

- **RTT Sesión iBGP 6VPE$_2$ - RR$_2$:**

- Filtro: *tcp and bgp and ip.src==2.2.2.2 and ip.dst==6.6.6.6.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_2$ y RR$_2$ es:
  iRTT: 92,745 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==2.2.2.2 and ip.dst==6.6.6.6*
  Se observa el RTT de toda la captura en la figura 4.50, y su valor promedio es:
  RTT: 326,343 ms.

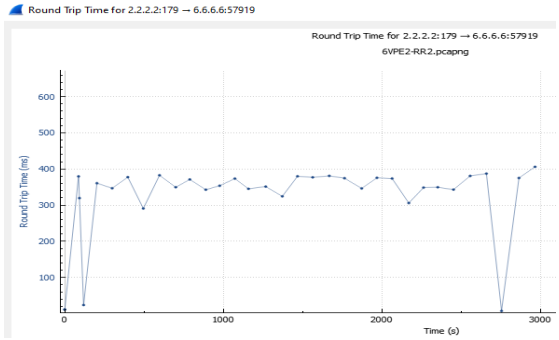


Figura 4.50 Escenario 5. RTT 6VPE$_2$ - RR$_2$

- **RTT Sesión iBGP 6VPE$_4$ - RR$_2$:**

- Filtro: *tcp and bgp and ip.src==4.4.4.4 and ip.dst==6.6.6.6.*
  El iRTT de la sesión iBGP entre los *Routers* 6VPE$_4$ y RR$_2$ es:
  iRTT: 50,966 ms.
- Filtro: *tcp.analysis.ack_rtt and ip.src==4.4.4.4 and ip.dst==6.6.6.6.*
  Se observa el RTT de toda la captura en la figura 4.51, y su valor promedio es:
  RTT: 330,986 ms.

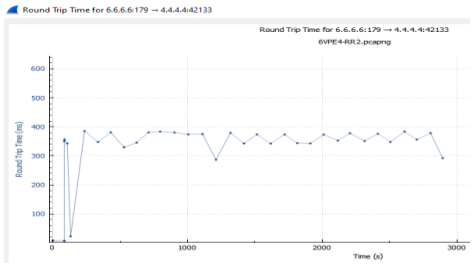


Figura 4.51 Escenario 5. RTT 6VPE$_4$ - RR$_2$

Con los resultados obtenidos para el Escenario 5 con métodos *Confederations BGP* y *Route Reflector* se genera la tabla 4.13 de análisis estadístico para los valores de iRTT y RTT.

Tabla 4.13 Escenario 5. Análisis Estadístico RTT

	iRTT (ms)	**RTT (ms)**
Promedio	47,282	328,239
Desviación Estándar	26,861	3,989
Coeficiente de variación	568,106	12,154

4.5.2.2 *Frame Arrival Delay* y *Jitter*

Es la diferencia de tiempo entre la trama que llega y la última trama recibida en los *Routers* 6VPE para las sesiones BGP del Escenario 5 *Confederations BGP*, se aplica el filtro: *tcp and bgp and ip.src==x.x.x.x and ip.dst==x.x.x.x* y el *Jitter* que es la variación del *Delay*, como se muestra en la tabla 5.14 con los valores promedios obtenidos para cada una de las sesiones BGP.

Tabla 4.14 Escenario 5. Frame Arrival Delay y Jitter Sesiones BGP

Conexión	***Frame Arrival Delay* (ms)**	***Jitter* (ms)**
6VPE$_1$ - RR$_1$	299,218	104,381
6VPE$_3$ - RR$_1$	321,269	92,546
6VPE$_1$ - 6VPE$_2$	316,817	73,041
6VPE$_2$ - RR$_2$	301,15	95,038
6VPE$_4$ - RR$_2$	316,711	84,523

Con los resultados obtenidos para el Escenario 5 con métodos *Confederations BGP* y *Route Reflector* se genera la tabla 4.15 Análisis Estadístico para los valores de *Frame Arrival Delay* y *Jitter*.

Tabla 4.15 Escenario 5. Análisis Estadístico Frame Arrival Delay y Jitter

	***Frame Arrival Delay* (ms)**	***Jitter* (ms)**
Promedio	311,033	89,906
Desviación Estándar	10,096	11,792
Coeficiente de variación	32,461	131,164

4.6 Análisis Comparativo de Resultados Finales

Se realizaron mediciones para los 5 Escenarios BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) para ISP sin y con métodos *Route Reflector y Confederations BGP* en un entorno de pruebas virtualizado durante 50 minutos, donde se compara el tamaño de los mensajes *Update* BGP, así como de los parámetros de desempeño tales como RTT, *Frame Arrival Delay y Jitter*.

4.6.1 Análisis Comparativo de Mensajes *UPDATE BGP*

Se obtuvo la cantidad de sesiones BGP y el tamaño promedio de los mensajes *UPDATE* de las mismas para todos los Escenarios, y se realiza el análisis comparativo calculando el porcentaje de variación con base al Escenario 1 donde no se aplicaron métodos, como se observa en la tabla 4.16.

Tabla 4.16 Análisis Comparativo Tamaño Mensajes Update entre Escenarios

N° Sesiones TCP - BGP	Escenario	Métodos	Tamaño (*bytes*)	Variación
IBGP = 6	**1**	Sin métodos	2460	**0%**
IBGP = 4	**2**	*Route Reflector*	1812,5	**- 26,32%**
IBGP = 8	**3**	*Cluster Route Reflector*	3570	**45,12%**
IBGP = 2 + EBGP = 1	**4**	*Confederations BGP*	1542,7	**- 37,29%**
IBGP = 4 + EBGP = 1	**5**	*Confederations BGP y Route Reflector*	2251	**- 8,50%**

Comparando el tamaño promedio de los mensajes *UPDATE* de las sesiones BGP entre todos los Escenarios, se puede observar que:

- Aplicando el método *Route Reflecto*r se disminuye el tamaño de los mensajes *UPDATE BGP* 26,32% que cuando no se aplican métodos, debido a que el *Router Reflector* se encarga de redistribuir los paquetes VPNv6 entre todos los *Routers* de borde 6VPE reduciendo 2 sesiones BGP.
- Aplicando el método *Cluster Route Reflecto*r se aumenta el tamaño de los mensajes *UPDATE BGP* 45,12% que cuando no se aplican métodos, debido a que al tener 2 *Routers Reflectors* aumenta la redundancia al redistribuir los paquetes VPNv6 los *Routers* de borde 6VPE con cada *Router Reflector;* por lo tanto, se establecen 2 sesiones BGP adicionales.
- Aplicando el método *Confederations BGP* se disminuye el tamaño de los mensajes *UPDATE BGP* 37,29% que cuando no se aplican métodos, debido a que al subdividir el AS solo se establecen 2 sesiones iBGP en los subsistemas y 1 sesión eBGP entre los mismos, reduciendo 3 sesiones BGP.
- Aplicando el método *Confederations BGP* y *Route Reflector* simultáneamente se disminuye el tamaño de los mensajes *UPDATE BGP* 8,5% que cuando no se aplican

métodos, debido a que al subdividir el AS se establecen 5 sesiones BGP (1 sesión eBGP entre los subsistemas y 2 sesiones iBGP entre los *Routers Reflectors* con los 6VPE de cada subsistema), reduciendo 1 sesión BGP.

4.6.2 Análisis Comparativo de Mediciones Estadísticas TCP

Se obtuvo las Mediciones Estadísticas Promedio TCP para los todos los Escenarios y se realiza el análisis comparativo para los parámetros de desempeño *Round Trip Time*, *Frame Arrival Delay* y *Jitter.*

4.6.2.1 *Round Trip Time* (RTT)

Se realiza el análisis comparativo del iRTT y del RTT promedio de los mensajes TCP de las sesiones BGP, calculando el porcentaje de variación entre los diferentes Escenarios con base al Escenario 1 donde no se aplicaron métodos, como se observa en la tabla 4.17.

Tabla 4.17 Análisis Comparativo iRTT y RTT entre Escenarios

N° Sesiones TCP - BGP	Escenario	Métodos	iRTT (ms)	Variación	RTT (ms)	Variación
IBGP = 6	**1**	Sin métodos	85,37	**0%**	331,63	**0%**
IBGP = 4	**2**	*Route Reflector*	46,21	**- 45,87%**	299,59	**- 9,66%**
IBGP = 8	**3**	*Cluster Route Reflector*	50,56	**- 40,78%**	319,35	**- 3,70%**
IBGP = 2 + EBGP = 1	**4**	*Confederations BGP*	58,39	**- 31,61%**	319,19	**- 3,75%**
IBGP = 4 + EBGP = 1	**5**	*Confederations BGP y Route Reflector*	47,28	**- 44,62%**	328,24	**- 1,02%**

Comparando el tiempo promedio iRTT y RTT de los mensajes TCP de las sesiones BGP entre todos los Escenarios, se puede observar que:

- Aplicando el método *Route Reflector* se disminuye el iRTT 45,87% y el RTT promedio 9,66% que cuando no se aplican métodos, por lo tanto, el *Router Reflector* reduce el número de sesiones BGP y optimiza el establecimiento de las sesiones BGP al convertir los *Routers* 6VPE en sus clientes.
- Aplicando el método *Cluster Route Reflector* se disminuye el iRTT 40,78% y el RTT promedio 3,7% que cuando no se aplican métodos, por lo tanto, la configuración de 2 *Routers Reflectors* optimiza el establecimiento de las sesiones BGP al convertir los *Routers* 6VPE en sus clientes, pero aumenta el número de sesiones BGP, comportándose mejor cuando solo se configura un *Router Reflector*.
- Aplicando el método *Confederations BGP* se disminuye el iRTT 31,61% y el RTT promedio 3,75% que cuando no se aplican métodos, debido a que al subdividir el AS se reducen la cantidad de sesiones BGP.
- Aplicando el método *Confederations BGP* y *Route Reflector* simultáneamente se disminuye el iRTT 44,62% y el RTT promedio 1,02% que cuando no se aplican métodos, debido a que al subdividir el AS sólo se establecen 2 sesiones iBGP en

cada subAS y una sesión eBGP entre los mismos. Se reducen el número de sesiones BGP y el *Router Reflector* optimiza el establecimiento de las sesiones BGP al convertir los *Routers* 6VPE en sus clientes.

4.6.2.2 *Frame Arrival Delay*

Se realiza el análisis comparativo del *Frame Arrival Delay* promedio de los mensajes TCP de las sesiones BGP, calculando el porcentaje de variación entre los diferentes Escenarios con base al Escenario 1 donde no se aplicaron métodos, como se observa en la tabla 4.18.

Tabla 4.18 Análisis Comparativo Frame Arrival Delay entre Escenarios

N° Sesiones TCP - BGP	Escenario	Métodos	*Frame Arrival Delay* (ms)	Variación
IBGP = 6	1	Sin métodos	302,4	**0%**
IBGP = 4	2	*Route Reflector*	275,01	**- 9,06%**
IBGP = 8	3	*Cluster Route Reflector*	294,81	**- 2,51%**
IBGP = 2 + EBGP = 1	4	*Confederations BGP*	300,89	**- 0,50%**
IBGP = 4 + EBGP = 1	5	*Confederations BGP y Route Reflector*	311,03	**2,85%**

Comparando el tiempo promedio *Frame Arrival Delay* de los mensajes TCP de las sesiones BGP entre todos los Escenarios, se puede observar que:

- Aplicando el método *Route Reflecto*r se disminuye el *Delay* 9,06% que cuando no se aplican métodos, por lo tanto, el *Router Reflector* reduce el número de sesiones BGP y optimiza el tiempo de las sesiones BGP al convertir los *Routers* 6VPE en sus clientes.
- Aplicando el método *Cluster Route Reflecto*r se disminuye el *Delay* 2,51% que cuando no se aplican métodos, por lo tanto, los *Routers Reflectors* optimizan el establecimiento de las sesiones BGP con los 6VPE al hacerlos sus clientes, pero establecen demasiadas sesiones BGP por lo que es mejor cuando solo se implementa un *Router Reflector*.
- Aplicando el método *Confederations BGP* se disminuye el *Delay* sólo 0,5% que cuando no se aplican métodos, por lo que no se observa un cambio significativo.
- Aplicando el método *Confederations BGP* y *Route Reflector* simultáneamente se aumenta el *Delay* 2,85% que cuando no se aplican métodos.

4.6.2.3 *Jitter*

Se realiza el análisis comparativo del *Jitter* promedio de los mensajes TCP de las sesiones BGP, calculando el porcentaje de variación entre los diferentes Escenarios con base al Escenario 1 donde no se aplicaron métodos, como se observa en la tabla 4.19.

Tabla 4.19 Análisis Comparativo Jitter entre Escenarios

N° Sesiones TCP - BGP	Escenario	Métodos	*Jitter* (ms)	Variación
IBGP = 6	**1**	Sin métodos	100,28	**0%**
IBGP = 4	**2**	*Route Reflector*	98,44	**- 1,83%**
IBGP = 8	**3**	*Cluster Route Reflector*	103,15	**2,86%**
IBGP = 2 + EBGP = 1	**4**	*Confederations BGP*	69,83	**- 30,37%**
IBGP = 4 + EBGP = 1	**5**	*Confederations BGP y Route Reflector*	89,9	**- 10,34%**

Comparando el tiempo promedio *Jitter* de los mensajes TCP de las sesiones BGP entre todos los Escenarios, se puede observar que:

- Aplicando el método *Route Reflecto*r se disminuye el *Jitter* 1,83% que cuando no se aplican métodos.
- Aplicando el método *Cluster Route Reflecto*r se aumenta el *Jitter* 2,86% que cuando no se aplican métodos.
- Aplicando el método *Confederations BGP* se disminuye el *Jitter* 30,37% que cuando no se aplican métodos, por lo tanto, este método aplica una mejora significativa brindando una conexión más estable de las sesiones BGP.
- Aplicando el método *Confederations BGP* y *Route Reflector* simultáneamente se disminuye el *Jitter* 10,34% que cuando no se aplican métodos, por lo tanto, la aplicación de estos métodos simultáneamente también brinda una conexión más estable de las sesiones BGP.

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

5.1 Conclusiones

Con el desarrollo del presente Trabajo de Grado respecto al análisis del desempeño de una red BGP/IPV6/VPN/MPLS aplicando los métodos *Route Reflector, Confederations BGP* y ambos simultáneamente en un entorno de pruebas virtualizado, se determinan las siguientes conclusiones:

- La escalabilidad en la implementación de redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de los ISP sobre la infraestructura existente *CORE* MPLS/IPv4, ha permitido coexistir los Protocolos de Internet IPv4 e IPv6, mejorando la capacidad, la calidad del servicio y los mecanismos de seguridad en forma gradual.

- Con los métodos propuestos *Route Reflector* y *Confederations BGP* aplicados en las redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) para ISP, se evita la interconexión de malla completa, logrando un mejor desempeño al reducir la cantidad de sesiones TCP, tamaño de mensajes de actualización en la red y mejores tiempos de convergencia.

- La implementación del método *Route Reflector* permite tener un mejor control y gestión en una red de operador en caso de expansión, dado que admite agregación de nuevos *Routers* de Borde 6VPE sin mayores configuraciones en la red, donde solamente se debe configurar el *Router Reflector* y el nuevo *Router* Cliente.

- Se observa a nivel general que al aplicar el método *Route Reflector* en la topología de red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de ISP implementada, se genera un mejor desempeño con respecto a los demás Escenarios, al presentar menores tiempos de establecimiento de las sesiones BGP y tiempos de respuesta promedio.

- En otras topologías de red BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) de ISP, la elección y aplicación de estos métodos deben ser evaluados en función de la estructura de la red. Se sugiere aplicar el método *Route Reflector* con uno o varios reflectores de ruta. Si se tiene una red extensa, primeramente, se debe subdividir aplicando el método de *Confederations BGP* para tener un mejor control y gestión de la red, y posteriormente, aplicar el método *Route Reflector,* configurando uno o varios reflectores de ruta. Se busca alcanzar un equilibrio que garantice la eficiencia, disponibilidad y robustez del sistema.

5.2 Recomendaciones

- Para realizar las simulaciones satisfactoriamente de los Escenarios descritos en este Trabajo de Grado, se recomienda utilizar un equipo (*Hardware*) que permita la tecnología de Virtualización tanto en la BIOS como en el Procesador simultáneamente y además cuente por lo menos con memoria RAM de 12 GB.

- Se recomienda utilizar la misma versión de GNS3 y del *Server* GNS3, ya que con versiones diferentes no permite la conectividad de la interfaz gráfica GNS3 con el servidor.

- En la implementación de redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) para ISP sobre la infraestructura existente *CORE* MPLS/IPv4, se recomienda aplicar el método *Route Reflector* y en Escenarios grandes o extensos primero aplicar el método *Confederations BGP* para subdividir en sistemas más pequeños y luego aplicar el método *Route Reflector,* con el fin de reducir el número de sesiones TCP y mejorar el desempeño de la red.

5.3 Trabajos Futuros

- Realizar análisis de desempeño de redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) basándose en modificación de las métricas de BGP y otros métodos que eviten realizar conexión *Full mesh* por la regla de *Split Horizon*.

- Realizar análisis de desempeño de redes BGP/IPV6/VPN/MPLS (*6VPE over MPLS*) aplicando tecnologías emergentes como lo son *Segment Routing*, *Traffic Engineering* (SR-TE) y comparar el desempeño de la red.

REFERENCIAS BIBLIOGRÁFICAS

[1] Z. Ashraf, A. Sohail, S. Latif, A. Hameed, y M. Yousaf, "*Challenges and Mitigation Strategies for Transition from IPv4 Network to Virtualized Next-Generation IPv6 Network*", *Int. Arab J. Inf. Technol.*, vol. 20, pp. 78-91, 2023, doi: 10.34028/iajit/20/1/9.

[2] M. M. Chinguel Rodríguez, "Revisión sistemática de los mecanismos de transición para la migración de ipv4- ipv6", *Repos. Inst. - USS*, 2019, Disponible en: http://repositorio.uss.edu.pe//handle/20.500.12802/6214

[3] D. Torres Sanchez, "INTRODUCCIÓN Y CONFIGURACIÓN DEL PROTOCOLO IPV6", Universidad Autonoma del Estado de Mexico, Estado de Mexico, 2017. Disponible en: https://core.ac.uk/works/8778622

[4] D. W. R. Bautista, «ESTUDIO DEL FUNCIONAMIENTO DE UNA RED IMPLEMENTADA EN PROTOCOLO IPv6», UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA, CAMPUS BUCARAMANGA, 2012. Disponible en: https://repository.unab.edu.co/bitstream/handle/20.500.12749/3327/2012_Tesis_Dewar_Willmer_Rico_Bautista.pdf?sequence=1&isAllowed=y

[5] J. F. Kurose y K. W. Ross, *"Redes de computadoras Un enfoque descendente"*, 7ma edición. Madrid, España, 2017. [En línea]. Disponible en: https://www.academia.edu/40738627/Redes_de_computadoras_Un_enfoque_descendente_7a_Edici%C3%B3n

[6] A. Cabellos-Aparicio y J. Domingo-Pascual, "Visión General del Protocolo IPv6", 2005. Disponible en: https://personals.ac.upc.edu/acabello/PDF/Vision_General_del_protocolo_IPv6-Novatica2005.pdf

[7] M. Gupta y A. Conta, "*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*", *Internet Engineering Task Force, Request for Comments* RFC 4443, mar. 2006. doi: 10.17487/RFC4443.

[8] Huawei Enterprise, "¿Qué es ICMPv6?". Disponible en: https://forum.huawei.com/enterprise/es/diferencias-entre-vrrp-y-hsrp/thread/667235910196019200-667212882523336704

[9] A. Viswanathan, E. C. Rosen, y R. Callon, "*Multiprotocol Label Switching Architecture*", Internet Engineering Task Force, Request for Comments RFC 3031, ene. 2001. doi: 10.17487/RFC3031.

[10] J. Lores Jacinto, "Configuración y pruebas de funcionamiento de la interconexión de redes heterogéneas con troncal MPLS", Universitat Politècnica de Catalunya, 2011. Disponible en: https://upcommons.upc.edu/handle/2099.1/11730

[11] A. S. Tanenbaum y D. J. Wetherall, *"*Redes de Computadoras*"*, 5ta Edición. México, 2012. Disponible en: https://bibliotecavirtualapure.files.wordpress.com/2015/06/redes_de_computadoras-freelibros-org.pdf

[12] S. C. Quintana Tejeda y M. O. Tabares Rodríguez, "*Multiprotocol Label Switching* (MPLS): usos, aplicaciones y áreas promisorias de la tecnología", *http://biblioteca.utb.edu.co/notas/tesis/0062650.pdf*, 2011 Disponible en: https://repositorio.utb.edu.co/handle/20.500.12585/503

[13] U. L. González, "Diseño de una Red Privada Virtual usando una red MPLS", 2017. Disponible en: https://oa.upm.es/49980/1/PFC_UNAI_LOPEZ_GONZALEZ.pdf

[14] L. H. Paredes Malpartida, «"Diseño de una red de proveedor de servicios de telecomunicaciones basado en Arquitectura SR-MPLS"», Pontificia Universidad Católica del Perú, Perú, 2021. Disponible en:

https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/19394/PAREDES_MALPARTIDA_LUIS_DISE%C3%91O_RED_PROVEEDOR.pdf?sequence=1
[15] R. Juárez y J. Paulino, “Propuesta de seguridad para el protocolo BGP en el atributo AS_PATH”, Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, 2009. Disponible en: https://repositorio.tec.mx/handle/11285/570000
[16] G. F. Lascano Tacuri, “ANÁLISIS COMPARATIVO DEL PROTOCOLO BGP EN IPV4 E IPV6 PARA LA TRANSMISIÓN DE SERVICIOS WEB DENTRO DE UN ESCENARIO BÁSICO”, ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, Riobamba – Ecuador, 2016. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/5057
[17] R. O. C. Granja, “Despliegue de IPv6 en un *backbone* MPLS/IPV4 para un Proveedor de Servicios”, ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL, GUAYAQUIL – ECUADOR, 2015. Disponible en: https://dspace.espol.edu.ec/retrieve/100162/D-84927.pdf
[18] S. Salih, A. Abdalrahman, y K. Elsharif, "*Performance Evaluation of IPv6 VPN Provider Edge Router*", en *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*, Manama, Bahrain: IEEE, may 2017, pp. 1-4. doi: 10.1109/IEEEGCC.2017.8448146.
[19] D. Gopinath, S. Vijayakumar, R. Ramalakshmi, M. Jeevalinggesh, y D. Janarthanan, "*Implementation of Internal and External Border Gateway Protocol for Efficient Traffic Flow using MPLS L3 VPN in Green Network*", *Journal of Green Engineering (JGE)*, vol. Volume 10, sep. 2020. Disponible en: http://www.jgenng.com/wp-content/uploads/2020/11/volume10-issue9-67.pdf
[20] V. Sánchez García, "DISEÑO DE REDES CON BGP", Universitat Politecnica de Valencia, Valencia, 2017. Disponible en: https://riunet.upv.es/bitstream/handle/10251/91691/S%C3%81NCHEZ%20-%20Dise%C3%B1o%20de%20redes%20con%20BGP.pdf?sequence=1
[21] E. Collado Cabeza, "Fundamentos de *Routing"*. 2009. [En línea]. Disponible en: https://books.google.co.ve/books?id=zfaN9k840xsC&printsec=frontcover#v=onepage&q&f=false
[22] G. Jambrina y V. Solla, "Estudio de algoritmos de localización de reflectores de rutas en un sistema autónomo de internet", Universidad de la República (Uruguay), Uruguay, 2016. Disponible en: https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/19029
[23] E. Chen, T. J. Bates, y R. Chandra, "*BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*", *Internet Engineering Task Force*, Request for Comments RFC 4456, abr. 2006. doi: 10.17487/RFC4456.
[24] Cisco Partner, Cisco Certified CCIE," *Route Reflector* Clusters | CCIE Docs". Disponible en: http://www.bscottrandall.com/3.7.2.2.html
[25] R. Molenaar, "*BGP Route Reflector*", *NetworkLessons.com*. Disponible en: https://networklessons.com/bgp/bgp-route-reflector
[26] O. J. S. Parra, L. F. Pedraza, y M. Espinosa, "Evaluación de redes MPLS/VPN/BGP con rutas reflejadas", *Tecnura*, vol. 16, n.º 32, Art. n.º 32, abr. 2012, doi: 10.14483/udistrital.jour.tecnura.2012.2.a09.
[27] P. S. Traina, J. Scudder, y D. R. McPherson, "*Autonomous System Confederations for BGP*", Internet Engineering Task Force, Request for Comments RFC 5065, ago. 2007. doi: 10.17487/RFC5065.
[28] Junos OS, Juniper Networks, "Confederaciones del BGP para el escalamiento del IBGP". Disponible en: https://www.juniper.net/documentation/mx/es/software/junos/bgp/topics/topic-map/bgp-confederations-for-scaling.html
[29] W. Eddy, "*Transmission Control Protocol (TCP)*", *Internet Engineering Task Force*, *Request for Comments* RFC 9293. 2022. doi: 10.17487/RFC9293.

[30] V. Cadin y C. Talay, "Evolución de los algoritmos de control de congestión en las distintas variantes del protocolo TCP", *Inf. Científico Téc. UNPA*, vol. 13, n.º 1, pp. 125-144, 2021. Disponible en: https://dialnet.unirioja.es/servlet/articulo?codigo=8009651
[31] R Martin, "TCP - Protocolos de la familia Internet". Disponible en: http://personales.upv.es/rmartin/TcpIp/cap02s12.html
[32] Cisco, CCNA Certification Community, "*TCP - Three-way handshake*". Disponible en: https://learningnetwork.cisco.com/s/article/tcp-three-way-handshake
[33] GeeksforGeeks, "*TCP 3-Way Handshake Process*". Disponible en: https://www.geeksforgeeks.org/tcp-3-way-handshake-process/
[34] C. A. Talay, D. R. R. Herlein, M. D. Labrador, C. N. González, y L. A. Marrone, "Protocolo TCP: El RTT como un factor de evaluación del rendimiento», 2021.
[35] StormIT, "*What is RTT (Round-Trip Time) and How to Reduce it*?". Disponible en: https://www.stormit.cloud/blog/what-is-round-trip-time-rtt-meaning-calculation/
[36] D. A. Miller y A. E. Kamal, "*Delay-stable communications in simultaneous multicast networks*", en *2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, USA: IEEE, dic. 2012, pp. 1733-1738. doi: 10.1109/GLOCOM.2012.6503365.
[37] Telcomanager, "La importancia del seguimiento de las estadísticas del jitter de la red". Disponible en: https://www.telcomanager.com/es/blog/la-importancia-del-seguimiento-de-las-estadisticas-del-jitter-de-la-red/
[38] Sonary, "*Jitter – What Is It And How To Deal With It?*". Disponible en: https://sonary.com/content/jitter-what-it-is-and-how-to-deal-with-it/
[39] Gantt Chart GanttPRO Blog, "Modelo cascada, qué es y cuándo conviene usarlo". Disponible en: https://blog.ganttpro.com/es/metodologia-de-cascada/
[40] Networks Training "*Comparison of GNS3 vs EVE-NG vs Packet Tracer for Networks Simulation*". Disponible en: https://www.networkstraining.com/gns3-vs-eve-ng-vs-cisco-packet-tracer/
[41] GNS3, "*The software that empowers network professionals*". Disponible en: https://www.gns3.com/

ANEXOS

- **Anexo A. Simulaciones GNS3 (Carpeta).**
- **Anexo B. Configuraciones (Carpeta).**
- **Anexo C. Capturas *Wireshark* (Carpeta).**