



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES  
Y REDES**

**IMPLEMENTACIÓN Y EVALUACIÓN DE SEGMENT ROUTING  
EN EL TRANSPORTE DE PAQUETES EN TIEMPO REAL  
MEDIANTE TÉCNICAS DE INGENIERÍA DE TRÁFICO CON  
IPV4.**

Trabajo de titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y  
REDES**

**AUTORES: LEIBERT SEBASTIÁN NAVARRETE ANDRADE**

**EDWIN PATRICIO RIVERA CAZCO**

**DIRECTOR: Ing. Msc ALBERTO LEOPOLDO ARELLANO AUCANCELA.**

Riobamba – Ecuador

2021

©2021, Leibert Sebastián Navarrete Andrade y Edwin Patricio Rivera Cazco

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Nosotros, Leibert Sebastián Navarrete Andrade y Edwin Patricio Rivera Cazco, declaramos que el presente trabajo de titulación es de nuestra autoría y que los resultados de este son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Como autores, asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 05 de enero de 2021



Leibert Sebastián Navarrete Andrade

1207342526



Edwin Patricio Rivera Cazco

160091070-5

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y**  
**REDES**

El tribunal del trabajo de titulación certifica que: El trabajo de titulación Tipo: Proyecto Técnico **IMPLEMENTACIÓN Y EVALUACIÓN DE SEGMENT ROUTING EN EL TRANSPORTE DE PAQUETES EN TIEMPO REAL MEDIANTE TÉCNICAS DE INGENIERÍA DE TRÁFICO CON IPV4,**

realizado por los señores: **LEIBERT SEBASTIÁN NAVARRETE ANDRADE Y EDWIN PATRICIO RIVERA CAZCO,** ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

|   | <b>FIRMA</b>  | <b>FECHA</b>      |
|---|---|-------------------|
| Ing. Mónica Andrea Zabala Haro<br><b>PRESIDENTE DEL TRIBUNAL</b>                  | <br>MONICA ANDREA ZABALA HARO<br>Firmado digitalmente por MONICA ANDREA ZABALA HARO<br>Fecha: 2021.04.08 08:17:05 -05'00'                     | <u>08/04/2021</u> |
| Ing. Msc. Alberto Arellano Aucancela<br><b>DIRECTOR DEL TRABAJO DE TITULACIÓN</b> | <br>ALBERTO LEOPOLDO ARELLANO AUCANCELA<br>Firmado digitalmente por ALBERTO LEOPOLDO ARELLANO AUCANCELA<br>Fecha: 2021.04.08 20:28:22 -05'00' | <u>08/04/2021</u> |
| Ing. Edwin Vinicio Altamirano Santillan<br><b>MIEMBRO DEL TRIBUNAL</b>            | <br>EDWIN VINICIO ALTAMIRANO SANTILLAN<br>Firmado digitalmente por EDWIN VINICIO ALTAMIRANO SANTILLAN<br>Fecha: 2021.04.09 07:07:08 -05'00'   | <u>08/04/2021</u> |

## DEDICATORIA

De todo corazón, es grato dedicar el presente trabajo a mi familia, en especial a mi madre Alexandra Andrade, a mis abuelos, quienes han estado presente para apoyarme y darme ánimos en los momentos más difíciles de la carrera, de igual manera quiero dedicarlo a Hypatia León ya que nunca perdió la confianza en mí siendo así mi motor y motivo para continuar. Ellos han sido el pilar más sólido, la fuerza y la motivación para salir adelante en cada momento difícil en que me encontraba.

**Leiber Sebastián Navarrete Andrade**

La elaboración de esta tesis de grado ha demandado un gran esfuerzo y sobre todo mucho sacrificio, es por ello que, quiero dedicar este trabajo a Dios, por iluminarme por el camino de la sapiencia. A mis padres Rigoberto Rivera y Julia Cazco, por todo el sustento, dedicación y comprensión de su parte, este apoyo se evidencia al culminar la etapa universitaria de mi vida. De Igual forma a mis hermanos y Marisol Fiallos, quienes son una gran inspiración de trabajo y superación.

**Edwin Patricio Rivera Cazco**

## **AGRADECIMIENTO**

El camino al éxito es largo y se logra cuando hay grandes personas a nuestro lado, por ello, queremos agradecer a nuestras familias por inculcarnos buenos valores como la perseverancia, responsabilidad y honestidad. A nuestra segunda familia (los buenos amigos), quienes han estado en las buenas, en las malas y en las peores; a nuestros maestros, quienes con su paciencia y determinación lograron plantar la semilla del conocimiento durante toda la carrera, y he aquí los evidentes frutos en nuestra vida profesional. Queremos extender un agradecimiento eterno al Mg. Alberto Arellano, quien contribuyó a la realización de este proyecto de titulación y como no, agradecer a nuestra grandiosa Alma Mater la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, quien nos supo recibir con las puertas abiertas y se convirtió en un hogar cálido que nos brindó la oportunidad de ser profesionales de calidad.

Sebastián Navarrete y Patricio Rivera

## TABLA DE CONTENIDOS

|   |      |
|---|------|
| ÍNDICE DE TABLAS .....  | ix   |
| ÍNDICE DE FIGURAS .....   | x    |
| ÍNDICE DE GRÁFICOS .....  | xi   |
| RESUMEN .....   | xiii |
| SUMARY .....  | xiv  |
| INTRODUCCIÓN .....  | 1    |
| CAPÍTULO I.....   | 8    |
| 1. MARCO TEÓRICO REFERENCIAL .....  | 8    |
| 1.1. <i>MPLS (Multiprotocol Label Switching)</i> .....                        | 8    |
| 1.1.1. <i>¿Qué es?</i> .....  | 8    |
| 1.1.2. <i>Ventajas y desventajas de MPLS</i> .....                            | 9    |
| 1.1.2.1. <i>Ventajas</i> .....  | 9    |
| 1.1.2.2. <i>Desventajas</i> .....   | 10   |
| 1.1.3. <i>Arquitectura de MPLS</i> .....                                      | 11   |
| 1.1.3.1. <i>LSR</i> .....   | 12   |
| 1.1.3.2. <i>LER</i> .....   | 12   |
| 1.1.3.3. <i>FEC</i> .....   | 12   |
| 1.1.3.4. <i>LSP</i> .....   | 12   |
| 1.1.4. <i>Cabecera de MPLS</i> .....  | 13   |
| 1.1.4.1. <i>Apilamiento de pilas</i> .....                                    | 14   |
| 1.2. <i>Ingeniería de Tráfico</i> .....                                       | 15   |
| 1.2.1. <i>IT orientada a Tráfico</i> .....                                    | 15   |
| 1.2.2. <i>IT orientada a Recursos</i> .....                                   | 16   |
| 1.2.3. <i>Componentes de Ingeniería de Tráfico</i> .....                      | 16   |
| 1.2.4. <i>Proceso de la Ingeniería de Tráfico</i> .....                       | 17   |
| 1.2.5. <i>Algoritmos de Ingeniería de Tráfico</i> .....                       | 18   |
| 1.2.5.1. <i>Algoritmo CSPF (Constraint Shortest Path First)</i> .....         | 19   |
| 1.2.5.2. <i>Algoritmo MIRA (Minimun Interference Routing Algorithm)</i> ..... | 19   |
| 1.2.5.3. <i>Algoritmo FAIR NETWORKS (Redes Justas)</i> .....                  | 20   |
| 1.3. <i>Segment Routing SR</i> .....  | 21   |
| 1.3.1. <i>Introducción</i> .....  | 21   |



|                           |  |    |
|---------------------------|--|----|
| 1.3.2.                    | <i>Operación de Segment Routing</i> .....  | 23 |
| 1.3.3.                    | <i>Segment Routing frente a los protocolos de señalización de etiquetas existentes</i> | 25 |
| 1.3.3.1.                  | <i>Segment Routing frente a LDP</i> .....  | 25 |
| 1.3.3.2.                  | <i>Segment Routing frente a RSVP-TE</i> .....  | 26 |
| 1.3.4.                    | <i>Arquitectura Segment Routing</i> .....  | 27 |
| 1.3.4.1.                  | <i>Plano de Datos</i> .....  | 28 |
| 1.3.4.2.                  | <i>Plano de control</i> .....  | 31 |
| 1.3.4.3.                  | <i>Ventajas de Segment Routing</i> .....   | 32 |
| 1.3.5.                    | <i>Aplicaciones de Segment Routing</i> .....   | 35 |
| 1.3.5.1.                  | <i>SR-BE</i> .....   | 35 |
| 1.3.5.2.                  | <i>SR-TE</i> .....   | 35 |
| 1.4.                      | <i>Software para pruebas</i> .....   | 36 |
| 1.4.1.                    | <i>Simulador de Redes</i> .....  | 36 |
| 1.4.1.1.                  | <i>Cisco Packet Tracer</i> .....   | 37 |
| 1.4.1.2.                  | <i>eNSP Huawei Technologies</i> .....  | 38 |
| 1.4.1.3.                  | <i>GNS3</i> .....  | 38 |
| 1.4.1.4.                  | <i>EVE-NG (Emulated Virtual Enviroment)</i> .....                                      | 39 |
| 1.4.1.5.                  | <i>VIRL</i> .....  | 40 |
| 1.4.2.                    | <i>Generadores de tráfico</i> .....  | 41 |
| 1.4.2.1.                  | <i>OSTINATO</i> .....  | 42 |
| 1.4.3.                    | <i>Analizadores de Tráfico</i> .....   | 44 |
| 1.4.3.1.                  | <i>Wireshark</i> .....   | 45 |
| <b>CAPITULO II</b> .....  |  | 47 |
| 2.                        | <b>MARCO METODOLOGICO</b> .....  | 47 |
| 2.1.                      | <i>Introducción</i> .....  | 47 |
| 2.2.                      | <i>Parámetros para determinar</i> .....  | 47 |
| 2.3.                      | <i>Análisis de software</i> .....  | 49 |
| 2.3.1.                    | <i>Software de simulación</i> .....  | 49 |
| 2.3.2.                    | <i>Software de Pruebas</i> .....   | 50 |
| 2.4.                      | <i>Configuración de Gns3</i> .....   | 51 |
| 2.5.                      | <i>Pruebas de rendimiento</i> .....  | 52 |
| 2.5.1.                    | <i>Implementación de escenario</i> .....   | 55 |
| 2.5.2.                    | <i>Evidencia de túneles MPLS por defecto</i> .....                                     | 60 |
| 2.5.3.                    | <i>Designación de túneles para ingeniería de tráfico</i> .....                         | 64 |
| <b>CAPITULO III</b> ..... |  | 66 |
| 3.                        | <b>MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN DE RESULTADOS</b> .....                   | 66 |

|                     |  |           |
|---------------------|--|-----------|
| <b>3.1.</b>         | <b><i>Implementación de túneles LDP con Ingeniería de Tráfico.....</i></b> | <b>66</b> |
| <b>3.1.1.</b>       | <b><i>Construcción de Túneles .....</i></b>                                | <b>66</b> |
| <b>3.1.2.</b>       | <b><i>Túneles para MPLS-TE .....</i></b>                                   | <b>67</b> |
| <b>3.1.3.</b>       | <b><i>Tráfico túneles MPLS-TE .....</i></b>                                | <b>68</b> |
| <b>3.1.3.1.</b>     | <b><i>Túnel 1 .....</i></b>  | <b>68</b> |
| <b>3.1.3.2.</b>     | <b><i>Túnel 2 .....</i></b>  | <b>69</b> |
| <b>3.1.3.3.</b>     | <b><i>Túnel 3 .....</i></b>  | <b>69</b> |
| <b>3.1.3.4.</b>     | <b><i>Túnel 4 .....</i></b>  | <b>70</b> |
| <b>3.2.</b>         | <b><i>Configuración Segment Routing .....</i></b>                          | <b>70</b> |
| <b>3.3.</b>         | <b><i>Pruebas de rendimiento.....</i></b>                                  | <b>73</b> |
| <b>3.4.</b>         | <b><i>Pruebas de interferencias latencia y Jitter .....</i></b>            | <b>78</b> |
| <b>BIBLIOGRAFÍA</b> |  |           |
| <b>ANEXOS</b>       |  |           |

## ÍNDICE DE TABLAS

|  |    |
|--|----|
| <b>Tabla 1-1:</b> Pronóstico de Cisco VNI: contexto histórico de Internet..... | 2  |
| <b>Tabla 1-2:</b> Comparativa de Tecnologías .....                             | 10 |
| <b>Tabla 2-1:</b> Comparación entre Segment Routing y MPLS .....               | 32 |
| <b>Tabla 1-2:</b> Requisitos mínimos .....                                     | 49 |
| <b>Tabla 2-2:</b> Comparativa de Software .....                                | 50 |
| <b>Tabla 3-2:</b> Direccionamiento IP .....                                    | 54 |
| <b>Tabla 4-2:</b> Ruta del tráfico MPLS.....                                   | 63 |
| <b>Tabla 5-2:</b> Asignación de túneles para comparativa .....                 | 64 |
| <b>Tabla 1-3:</b> Asignación de túneles para comparativa .....                 | 77 |
| <b>Tabla 2-3:</b> Asignación de túneles para comparativa .....                 | 81 |

## ÍNDICE DE FIGURAS

|   |    |
|---|----|
| <b>Figura 1-1:</b> Tráfico Global por dispositivos .....  | 3  |
| <b>Figura 2-1:</b> Escenario tentativo .....  | 6  |
| <b>Figura 2-2:</b> Arquitectura de MPLS .....   | 13 |
| <b>Figura 3-1:</b> Apilamiento de etiquetas .....   | 14 |
| <b>Figura 4-1:</b> Fases de Ingeniería de Tráfico .....   | 17 |
| <b>Figura 5-1:</b> Fases de Ingeniería de Tráfico .....   | 22 |
| <b>Figura 6-1:</b> Segment Routing y SID de adyacencia .....                                      | 23 |
| <b>Figura 7-1:</b> Prefijo SR o Nodo SID Señalización y reenvío utilizando IGP .....              | 24 |
| <b>Figura 8-1:</b> Plano de Datos .....   | 28 |
| <b>Figura 9-1:</b> Modo Basado en segmento de prefijo .....                                       | 29 |
| <b>Figura 10-1:</b> Modo Basado en segmentos de adyacencia.....                                   | 30 |
| <b>Figura 11-1:</b> Modo Basado en el que se combinan los segmentos de adyacencia y nodo<br>..... | 31 |
| <b>Figura 12-1:</b> Número de Estados de Conexión.....  | 34 |
| <b>Figura 13-1:</b> Optimización de Control .....   | 34 |
| <b>Figura 14-1:</b> Modos de Dirección del Tráfico .....  | 36 |
| <b>Figura 15-1:</b> Interfaz de Cisco Packet Tracer.....  | 37 |
| <b>Figura 16-1:</b> Interfaz de eNSP .....  | 38 |
| <b>Figura 17-1.</b> Interfaz de GNS3 .....  | 39 |
| <b>Figura 18-1:</b> Interfaz de EVE-NG .....  | 40 |
| <b>Figura 19-1.</b> Interfaz de VIRT .....  | 41 |
| <b>Figura 20-1.</b> Interfaz de OSTINATO .....  | 44 |
| <b>Figura 21-1.</b> Interfaz de WIRESHARK.....  | 46 |
| <b>Figura 1-2:</b> Ciclo Deming.....  | 48 |
| <b>Figura 2-2:</b> Ciclo Deming.....  | 48 |

## ÍNDICE DE GRÁFICOS

|  |    |
|--|----|
| <b>Gráfico 1-2:</b> Configuración de Routers en Gns3 .....                   | 51 |
| <b>Gráfico 2-2:</b> Configuración de Ostinato en Gns3 .....                  | 52 |
| <b>Gráfico 3-2:</b> Escenario de prueba .....                                | 52 |
| <b>Gráfico 4-2:</b> Clasificación de routers en el escenario de prueba ..... | 53 |
| <b>Gráfico 5-2:</b> Configuración Routers PE.....                            | 56 |
| <b>Gráfico 6-2:</b> Configuración Routers P .....                            | 56 |
| <b>Gráfico 7-2:</b> Prueba conectividad IP entre PCs .....                   | 57 |
| <b>Gráfico 8-2:</b> Rutas aprendidas por OSPF desde router PE1 .....         | 57 |
| <b>Gráfico 9-2:</b> Configuración de MPLS en router PE .....                 | 58 |
| <b>Gráfico 10-2:</b> Configuración MPLS en router P.....                     | 58 |
| <b>Gráfico 11-2:</b> Tabla de forwarding equipos PE .....                    | 59 |
| <b>Gráfico 12-2:</b> Tabla de forwarding equipos P1 .....                    | 59 |
| <b>Gráfico 13-2:</b> Tráfico enviado desde A1 hasta A3 .....                 | 60 |
| <b>Gráfico 14-2:</b> Tráfico enviado desde A1 hasta A4 .....                 | 60 |
| <b>Gráfico 15-2:</b> Tráfico enviado desde A2 hasta A3 .....                 | 61 |
| <b>Gráfico 16-2:</b> Tráfico enviado desde A2 hasta A4 .....                 | 61 |
| <b>Gráfico 17-2:</b> Tráfico enviado desde A3 hasta A1 .....                 | 61 |
| <b>Gráfico 18-2:</b> Tráfico enviado desde A3 hasta A2 .....                 | 62 |
| <b>Gráfico 19-2:</b> Tráfico enviado desde A4 hasta A1 .....                 | 62 |
| <b>Gráfico 20-2:</b> Tráfico enviado desde A4 hasta A2.....                  | 62 |
| <b>Gráfico 21-2:</b> Ruta para el paso de tráfico .....                      | 63 |
| <b>No se encuentran elementos de tabla de ilustraciones.</b>                 |    |
| <b>Gráfico 2-3:</b> Túneles para MPLS TE en equipos P .....                  | 67 |
| <b>No se encuentran elementos de tabla de ilustraciones.</b>                 |    |
| <b>Gráfico 4-3:</b> Túneles para MPLS TE en equipos PE .....                 | 68 |
| <b>No se encuentran elementos de tabla de ilustraciones.</b>                 |    |
| <b>Gráfico 6-3:</b> Verificación Túnel 2 .....                               | 69 |
| <b>Gráfico 7-3:</b> Verificación Túnel 3 .....                               | 70 |

|   |    |
|---|----|
| <b>Gráfico 8-3:</b> Verificación Túnel 3 .....  | 70 |
| <b>Gráfico 9-3:</b> Configuración Segment Routing.....                                    | 71 |
| <b>Gráfico 10-3:</b> Configuración Segment Routing.....                                   | 71 |
| <b>Gráfico 11-3:</b> Configuración Segment Routing.....                                   | 72 |
| <b>Gráfico 12-3:</b> Configuración Segment Routing.....                                   | 72 |
| <b>Gráfico 13-3:</b> Configuración Segment Routing.....                                   | 73 |
| <b>Gráfico 14-3:</b> Asignación de etiqueta única por Segment Routing .....               | 73 |
| <b>Gráfico 15-3:</b> Configuración Ostinato .....   | 74 |
| <b>Gráfico 16-3:</b> Captura MPLS-LDP con 64 bytes de longitud .....                      | 74 |
| <b>Gráfico 17-3:</b> Captura MPLS-LDP con 750 bytes de longitud .....                     | 75 |
| <b>Gráfico 18-3:</b> Captura MPLS-LDP con 1500 bytes de longitud .....                    | 75 |
| <b>Gráfico 19-3:</b> Captura MPLS-SR con 64 bytes de longitud .....                       | 76 |
| <b>Gráfico 20-3:</b> Captura MPLS-SR con 750 bytes de longitud .....                      | 76 |
| <b>Gráfico 21-3:</b> Captura MPLS-SR con 1500 bytes de longitud .....                     | 77 |
| <b>Gráfico 22-3:</b> Gráfico comparativo tiempos de conmutación .....                     | 78 |
| <b>Gráfico 23-3:</b> Interferencias en MPLS-LDP con longitud de trama de 64 bytes .....   | 79 |
| <b>Gráfico 24-3:</b> Interferencias en MPLS-LDP con longitud de trama de 750 bytes .....  | 79 |
| <b>Gráfico 25-3:</b> Interferencias en MPLS-LDP con longitud de trama de 1500 bytes ..... | 79 |
| <b>Gráfico 26-3:</b> Interferencias en MPLS-SR con longitud de trama de 64 bytes .....    | 80 |
| <b>Gráfico 27-3:</b> Interferencias en MPLS-SR con longitud de trama de 750 bytes .....   | 80 |
| <b>Gráfico 28-3:</b> Interferencias en MPLS-SR con longitud de trama de 1500 bytes .....  | 80 |
| <b>Gráfico 29-3:</b> Interferencias MPLS-LDP y MPLS-SR.....                               | 81 |

## RESUMEN

El objetivo del presente trabajo de titulación fue la implementación y evaluación de Segment Routing en el transporte de paquetes en tiempo real, mediante técnicas de ingeniería de tráfico con ipv4. En la investigación fue necesario la realización e implementación de diversos laboratorios de prueba para evaluar el tráfico dentro de una red MPLS, para ellos se hizo uso de IOS XR de los router Cisco System de última generación. Los escenarios fueron desarrollados dentro de la plataforma de simulación GNS3, por otra parte, se realizó diversas pruebas de rendimiento enviando tráfico mediante el Software OSTINATO, y para la comparación se desarrolló ingeniería de tráfico con los algoritmos más conocidos en la actualidad que son: CSPF, MIRA, FAIR NETWORKS. El tráfico inyectado por el software mencionado fue estudiado gracias a WIRESHARK, quien se encargó de capturar el comportamiento de los paquetes enviados por toda la red, de esta forma se pudo evidenciar tiempos de retardo, paquetes perdidos, velocidad de transmisión, entre otros. Segment Routing demostró ser un algoritmo eficiente y eficaz para ingeniería de tráfico en comparación con los algoritmos utilizados en la actualidad, convirtiéndose en una nueva pero útil alternativa para que los proveedores de ISP brinden un mejor servicio dentro de su Core MPLS. De esta manera se reduce los cuellos de botella en la salida del tráfico de datos, y es por esta razón que, se recomienda realizar más investigaciones acerca de esta técnica de ingeniería de tráfico, pero sobre el protocolo de internet versión 6 (IPv6), debido a que es la tecnología hacia donde estamos encaminados.

PALABRAS CLAVE: <REDES DE COMPUTADORES>, <INGENIERÍA DE TRÁFICO>, <CALIDAD DE SERVICIO>, <PROTOCOLOS>, <ENRUTAMIENTO>, <TRÁFICO DE DATOS>, <RED>. <ANÁLISIS DE TRÁFICO>, <CONMUTACIÓN DE ETIQUETAS>, <SOFTWARE>.



## SUMMARY

This work aimed at analyzing the performance and interoperability of Hardware/Software (HW/SW) open source platform. An operating system called FRRouting (FRR) free software was installed in a Jetway mainboard and several performance and interoperability tests with other network devices were done. The scientific method was used to develop the software. The HW/SW platform has the Linux-Ubuntu 16.04 operating system, in which FRR – a set of routing protocols IP including daemons for dynamic routing protocols that is a current version of Quagga – was configured. Several interoperability tests were carried out using an internet protocol version 4 (IPv4) and internet protocol version 6 (IPv6) in different scenarios with several current existing dynamic routing protocols and with redistribution. For the scenario, testing equipment from different developers, such as Mikrotik, Cisco were used. The terminal performance was verified studying two important cases: receiving injected traffic from the Ostinato testing tools and determining the behavior during the test with Wireshark; the transmission speed and the quantity of packet loss, and so on were obtained. A virtual route injector (vRIN) – a generator system that permitted to send created route requests and to know the device transmission speed – was used. These tests were compared to Cisco testing scenario including similar features. The device is 100% interoperable and has a high performance in different posed scenarios. This equipment is less expensive and more profitable for enterprises that do not require specific areas of advanced networks. It is recommended to verify the compatibility of the open source systems with hardware mainboard when installing them.

KEYWORDS: <TECHNOLOGY AND ENGINEERING SCIENCES>, <COMPUTER NETWORKS>, <INTEROPERATION OF NETWORK OF COMPUTERS>, <PERFORMANCE>, <ROUTING PROTOCOLS>, <DATA TRAFFIC>, <PACKETS>, <HARDWARE>, <SOFTWARE>.





## INTRODUCCIÓN

Internet es una tecnología que se está implementando en todo ámbito, ya sea empresarial, social, médico, financiero, entre otros. Podemos decir que, dicha tecnología es un fenómeno global, el cual está vinculado directamente con la comunicación y la sociedad.

La nueva edición del informe presentado por We Are Social y Hootsuite en 2019, que año a año muestra las estadísticas, análisis y principales tendencias acerca del número de usuarios de internet, el uso de la telefonía móvil, las redes sociales y el e-Commerce a nivel global, asegura que: “Hasta el momento el número de usuarios de Internet en el mundo crece un 9,1% y alcanza los 4.388 millones” (We Are Social Ltd., 2019, p. 1)

En los últimos años, el internet se ha transformado en una red de muy alta difusión en cuanto al número de usuarios conectados. Esto ha sido visto por parte de los operadores como una oportunidad de ofrecer nuevos servicios a dichos usuarios a más de: email, ftp y navegación Web; algunos de estos servicios son por ejemplo telefonía IP, videoconferencia, televisión, radio, entre otros (Crow, 2016, p. 1).

Estos nuevos servicios poseen diferentes requerimientos en cuanto a su volumen de tráfico, calidad de servicio y seguridad. Dado que, en sus inicios el internet no fue pensado por sus diseñadores originales para trabajar en el contexto de grandes servicios, sino más bien en un contexto académico con intercambio de información básica del tipo emails, o ftp (Crow, 2016, p. 1).

En inicios del uso de internet, los proveedores estaban conectados por dispositivos que se comunicaban de forma directa entre ellos, este tipo de conexión generaba una saturación en el tráfico de datos de alta demanda, razón por la cual se empezaba a incrementar el rendimiento en el hardware de los enrutadores, sin embargo, esta solución era momentánea debido al incremento de demandas de servicio.

A fin de solventar el inconveniente de cuellos de botella en el tráfico de datos se empezó a implementar diversas técnicas de comunicación a nivel de software y hardware mismas que se denominaron Protocolos de enrutamiento y conmutación de paquetes.

Al respecto, Chávez (2003) indica que: “En el inicio MPLS fue presentado como una solución para mejorar la velocidad en los enrutadores, sin embargo, en la actualidad está emergiendo como una tecnología de estándares crucial, para ofrecer nuevas funcionalidades a redes IP de gran escala como, por ejemplo: Ingeniería de Tráfico, Calidad de Servicio, entre otros” (p. 3).

Como podemos apreciar en el informe presentado por Cisco Mobile VNI, lanzado en febrero de 2019, se ha puesto en evidencia los datos de los siguientes tipos de tráfico de internet:

- El tráfico anual de IP global alcanzará 4.8 ZB por año para 2022, o 396 exabytes (EB) por mes. En 2017, la tasa de ejecución anual para el tráfico IP global fue de 1.5 ZB por año, o 122 EB por mes, en la Tabla 1 se puede apreciar el crecimiento del tráfico de internet desde el año 1992 el cual ha tenido un crecimiento exponencial como se detalla a continuación.

**Tabla 1-1:** Pronóstico de Cisco VNI: contexto histórico de Internet

| <b>Año</b> | <b>Tráfico global de internet</b> |
|------------|-----------------------------------|
| 1992       | 100 GB por día                    |
| 1997       | 100 GB por hora                   |
| 2002       | 100 GB por segundo                |
| 2007       | 2,000 GB por segundo              |
| 2017       | 46,600 GB por segundo             |
| 2022       | 150,700 GB por segundo            |

**Fuente:** (Cisco VNI, 2018)

**Realizado por:** Navarrete Sebastián, Rivera Patricio, 2019

A nivel mundial, los dispositivos y las conexiones están creciendo 10% más rápido. Esta tendencia está acelerando el aumento en el número promedio de dispositivos y conexiones por hogar y por usuario.

Cada año, se introducen y adoptan en el mercado una variedad de nuevos dispositivos en diferentes formas con mayores capacidades e inteligencia. Un número creciente de aplicaciones M2M (máquina a máquina), como medidores inteligentes, video vigilancia, monitoreo de atención médica, transporte y seguimiento de paquetes o activos, están contribuyendo de manera

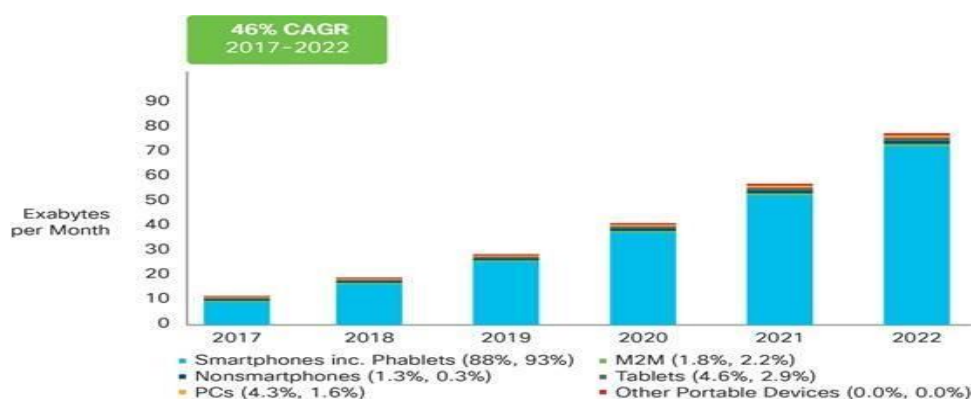
importante al crecimiento de dispositivos y conexiones. Para 2022, las conexiones M2M serán el 51 por ciento del total de la conexión de dispositivos (Cisco, 2018, p. 1).

Las conexiones M2M serán la categoría de más rápido ascenso, creciendo casi 2.4 veces durante el período de pronóstico, con un 19% de CAGR (Tasa anual compuesta de crecimiento), a 14.6 mil millones de conexiones para el año 2022 (Cisco, 2018, p. 1).

Los teléfonos inteligentes irán aumentando sus conexiones, con un CAGR del 9 por ciento. Los televisores conectados (que incluyen televisores de panel plano, decodificadores, adaptadores de medios digitales (DMA), reproductores de discos Blu-ray y consolas de juegos) crecerán más rápidamente con un CAGR del 7 por ciento a 3.200 millones en 2022. Las PC sufrirán una disminución del 2.5 por ciento durante el período de pronóstico. Sin embargo, habrá más PC que tabletas a lo largo del período de pronóstico y para fines de 2022 (1.200 millones de PC frente a 790 millones de tabletas) (Cisco, 2018, p. 1).

Para 2022, la participación del consumidor en el total de dispositivos, incluidos los dispositivos fijos y móviles, será del 72 por ciento, y las empresas reclamarán el 28 por ciento restante. La participación del consumidor crecerá a una tasa ligeramente más lenta, a una tasa anual compuesta anual de 8.8 por ciento en relación con el segmento de negocios, que crecerá a una tasa anual compuesta anual de 12.0 por ciento (Cisco, 2018, p. 1).

En la figura 1 se muestra las proyecciones del consumo de internet, en los próximos 5 años:



**Figura 1-1:** Tráfico Global por dispositivos

Fuente: (Cisco VNI, 2018)

MPLS, es una tecnología que trabaja a nivel de Capa 2 y 3, esto le permite tener varias funcionalidades para mejorar el tráfico de datos por el protocolo IP, uno de ellos es el Balanceo de Carga, el cual ayuda a disminuir los cuellos de botella en la transmisión de datos.

## **FORMULACIÓN DEL PROBLEMA**

¿Cómo mejorará la ingeniería de tráfico en una red que usa Segment Routing frente a una red que usa MPLS Tradicional?

¿Cuál es el fundamento teórico de los mecanismos de ingeniería de tráfico en Redes IPv4/MPLS?

¿Cómo es la arquitectura de la tecnología Segment Routing en redes IPv4/MPLS?

¿Dónde implementar las técnicas de ingeniería de tráfico en la transmisión de paquetes que soporte IPv4/MPLS y Segment Routing?

¿Cuál es el comportamiento de la latencia, el jitter y el número de paquetes perdidos en la transmisión en el escenario propuesto?

¿Cómo aplicar la mejor técnica a utilizarse en la tecnología Segment Routing en redes IPv4-MPLS entre los diferentes fabricantes de routing a implementarse en la red?

¿Qué pruebas de conectividad se puede realizar entre los diferentes equipos y proveedores para la validación de Segment Routing en tráfico con IPv4?

## **JUSTIFICACIÓN**

La tecnología MPLS, al ser un protocolo híbrido, tiene la capacidad de trabajar tanto en las capas dos y tres del modelo OSI, esto le convierte en un protocolo muy eficiente dentro de las redes de computadores, sin embargo, una gran limitante es quizá su alto coste de adquisición.

El ruteo basado en restricciones es un pilar fundamental de la ingeniería de tráfico en MPLS. Si bien CBR (Tasa de Bits Constantes) es un servicio, muy útil para el flujo de datos multimedia con canales de capacidad limitada. Cabe señalar que esta tecnología no tiene la capacidad de asignar suficientes bits en secciones complicadas lo que produce un consumo innecesario de bits en secciones simples, este servicio es utilizado por el algoritmo de CSPF (Constrained Shortest Path First) (Delfino, Rivero & San Martín, 2006).

La congestión es un fenómeno nada deseable, cuya causa con factores como: la insuficiencia de recursos en la red. En casos de congestión de algunos enlaces, el problema se resolvía al añadir más capacidad a los enlaces. Otra causa de congestión es la utilización ineficiente de los recursos debido al mapeado del tráfico. Cabe resaltar que, el objetivo básico de la Ingeniería de Tráfico es adaptar los flujos de tráfico a los recursos físicos de la red (Delfino, Rivero & San Martín, 2006, pp. 27-28)

Segment Routing (SR) se basa en el concepto de enrutamiento de origen suelto. Un nodo puede incluir una lista ordenada de instrucciones en los encabezados de paquetes. Estas instrucciones dirigen el reenvío y el procesamiento del paquete a lo largo de su recorrido en la red. Las instrucciones individuales se denominan segmentos, una secuencia de instrucciones puede denominarse una lista de segmentos o una política de SR.

Cuando la lista de segmentos es insertada por un nodo intermedio, puede ser eliminada por otro nodo a lo largo de la ruta del paquete, lo que apoya el concepto de tunelización a través de un dominio SR desde un nodo de ingreso a un nodo de egreso.

Para el presente trabajo se ha estudiado diferentes temas de investigación, como: “Segment routing principles and applications for SDN” realizado por Ana Koss, de Politecnico Di Milano, o también “Reaping the benefits of IPV6 segment routing” del autor David Lebrun, de la Universidad Católica de Louvain-Belgica y teniendo en consideración que, las investigaciones acerca de Segment Routing son solo extranjeras se resolvió ser los pioneros en este tema.

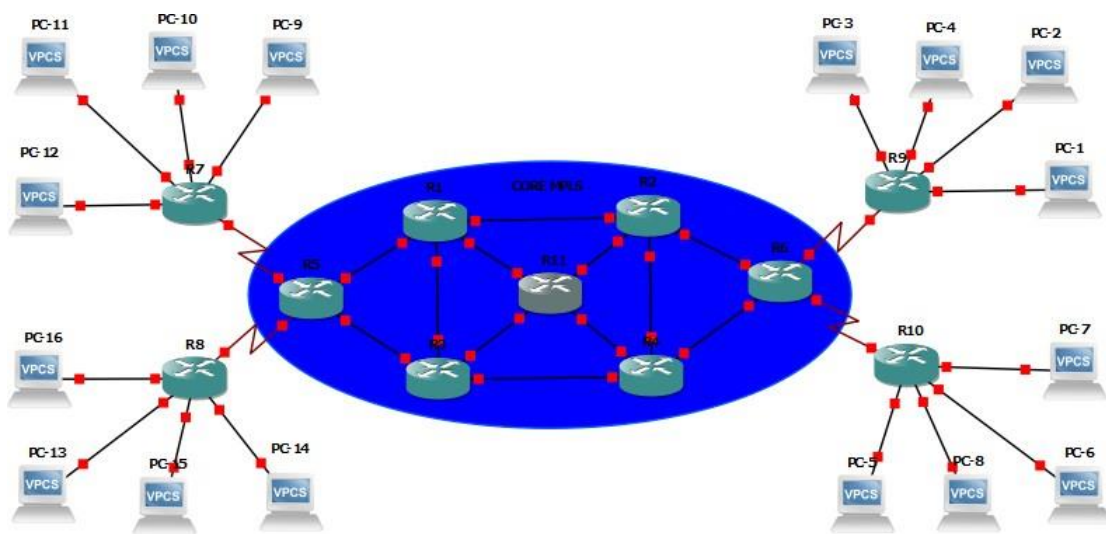
En este estudio se busca analizar el comportamiento de la red mediante la ingeniería de tráfico con Segment Routing en MPLS, frente a las redes convencionales MPLS.

Una manera eficaz para analizar el rendimiento de SRv4, es realizar simulaciones con IOS que de marcas capaces de soportar esta tecnología como son Cisco, Juniper, Huawei, ya que al ser una tecnología nueva no todos los sistemas operativos de los routers cuentan con la capacidad para acoplarse a esta tecnología.

En el proyecto de investigación, se pretende realizar escenarios de prueba para evaluar el comportamiento frente a la tradicional técnica de ingeniería de tráfico de MPLS en Ipv4,

analizando la calidad de servicio en la transmisión de paquetes, basándonos en parámetros básicos que garantizan un buen QoS (Calidad de servicio) los cuales son, Velocidad de Transferencia, latencia, jitter y pérdida de paquetes.

Para el trabajo de titulación, se presenta el escenario tentativo que nos permitirá evaluar el rendimiento de la red con ingeniería de tráfico mediante Segment Routing, frente a la ingeniería de tráfico en redes MPLS convencional; el escenario se muestra a continuación en la Figura 2



**Figura 2-1:** Escenario tentativo

**Realizado por:** Navarrete Sebastián, Rivera Patricio. 2019

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Implementar y evaluar Segment Routing en el transporte de paquetes en tiempo real mediante técnicas de ingeniería de tráfico con ipv4.

### **OBJETIVOS ESPECÍFICOS**

- Analizar el fundamento teórico de los mecanismos de ingeniería de tráfico en Redes IPv4/MPLS.
- Estudiar la arquitectura de la tecnología Segment Routing en redes IPv4/MPLS.
- Implementar las técnicas de ingeniería de tráfico en la transmisión de paquetes en un escenario propuesto que soporte IPv4/MPLS y Segment Routing.
- Determinar el comportamiento de la latencia, el jitter y el número de paquetes perdidos en la transmisión en el escenario propuesto.
- Aplicar la mejor técnica a utilizarse en la tecnología Segment Routing en redes IPV4-MPLS entre los diferentes fabricantes de routing a implementarse en la red.
- Realizar pruebas de conectividad entre los diferentes equipos y proveedores para la validación de Segment Routing en tráfico con iIPv4.

# CAPÍTULO I

## 1. MARCO TEÓRICO REFERENCIAL

### 1.1. MPLS (Multiprotocol Label Switching)

#### 1.1.1. *¿Qué es?*

El protocolo de conmutación de etiquetas mejor conocido como MPLS por sus siglas en inglés Multiprotocol Label Switching, se usa para la transmisión de información a través de etiquetas, este estándar fue creado por la organización IETF. Fue creado con el objetivo de unificar diferentes tipos de datos que se transmiten dentro de una red, busca que el envío de paquetes no ocasione problemas en la misma.

También, se debe comprender que “MPLS no es un servicio, sino una técnica de transferencia de datos. No es muy popular debido a sus elevados costos, pero beneficia principalmente a los servicios WAN y en soluciones de privacidad internacional como una VPN” (Optical Networks, 2019, p. 1).

MPLS implementa enrutadores y etiquetas específicas a la información de diferentes tipos para enviarlas por una ruta de baja latencia, esto significa que los datos como voz e imágenes pueden viajar por largas distancias de manera rápida y liviana. (Optical Networks, 2019, p. 2).

Por otra parte, el libro “MPLS Fundamentals” publicado por la empresa Cisco System (2020), afirma que:

Las etiquetas MPLS se anuncian entre enrutadores para que puedan crear una asignación de etiqueta a etiqueta. Estas etiquetas se adjuntan a los paquetes IP, lo que permite a los enrutadores reenviar el tráfico mirando la etiqueta y no la dirección IP de destino. Estos se reenvían mediante conmutación de etiquetas en lugar de utilizar la tradicional conmutación de IP (Ostoa, 2020, p. 23).



La técnica de cambio de etiqueta no es nueva. Frame Relay y ATM lo usan para mover tramas o celdas a través de una red. En Frame Relay, la trama puede tener cualquier longitud, mientras que, en ATM, una celda de longitud fija consta de un encabezado de 5 bytes y una carga útil de 48 bytes. El encabezado de la celda ATM y la trama Frame Relay se refieren al circuito virtual en el que reside la celda o la trama. La similitud entre Frame Relay y ATM es que, en cada salto a través de la red, se cambia el valor de la "etiqueta" en el encabezado (Ostoa, 2020, p. 23).

Esto es diferente del reenvío de paquetes IP. Cuando un enrutador reenvía un paquete IP, no cambia el valor que pertenece al destino del paquete; es decir, no cambia la dirección IP de destino del paquete. El hecho que, para enviar un paquete se usa las etiquetas MPLS y ya no la dirección IP de destino ha llevado a la popularidad de MPLS (Ostoa, 2020, p. 23).

En la actualidad, los grandes proveedores de internet como CNT, CEDIA, TELCONET, están utilizando la tecnología MPLS, para poder proveer del servicio de internet y de esta forma mejorar el servicio, lo que coadyuva a ser más competitivos en el país.

### ***1.1.2. Ventajas y desventajas de MPLS***

Se debe considerar que el objetivo de la creación de MPLS no fue reemplazar ATM sino más bien complementarlo, además que este protocolo fue desarrollado como una solución IP sobre Frame Relay.

#### ***1.1.2.1. Ventajas***

Dentro de las principales ventajas que podemos mencionar tenemos:

- **Ahorro de Costos:** Dependerá directamente de la combinación específica y la configuración de red de una empresa.
- **Soporte QoS:** Una de las principales ventajas de MPLS es la capacidad de aplicar Calidad de Servicio QoS (Calidad de Servicio).

- **Optimización de Rendimiento:** Gracias a su configuración “muchos a muchos” que este protocolo posee, mejora el rendimiento de la red.
- **Capacidad para recuperación de desastres:** Todo servicio implementado en el protocolo MPLS, tiene la capacidad de recuperar la información en un desastre, mediante una diversidad de técnicas.
- **Escalable:** Al ser una red con capacidad de poderse ampliar, la mayoría de las empresas de telecomunicaciones como Claro o Telconet aseguran que, MPLS representa 'el camino al futuro'.
- **Ingeniería de tráfico:** Brinda la oportunidad para la creación de túneles, con el objetivo de priorizar el tráfico en la red.

#### 1.1.2.2. Desventajas

A pesar de que MPLS es un protocolo de red muy utilizado, robusto y avanzado, aún presenta ciertas desventajas como son:

- **Incremento de consumo de recursos en los enrutadores:** Los enrutadores necesitan más recursos de hardware para procesar información MPLS
- **Aumento de la proporción de cabecera transportada:** Esto puede reducir el porcentaje de eficiencia de la red.

Como se puede evidenciar MPLS posee más ventajas que desventajas en comparación a sus antecesores, para un mejor entendimiento se ha realizado una comparativa mostrada en la Tabla 1-1 detallada a continuación:

**Tabla 1-1:** Comparativa de Tecnologías

|                   | <b>ATM</b>                                     | <b>FRAME RELAY</b>                 | <b>MPLS</b>                           |
|-------------------|--|------------------------------------|---------------------------------------|
| <b>DEFINICIÓN</b> | Basado en la conmutación de celdas y circuitos | Basado en la transmisión de tramas | Basado en la conmutación de etiquetas |

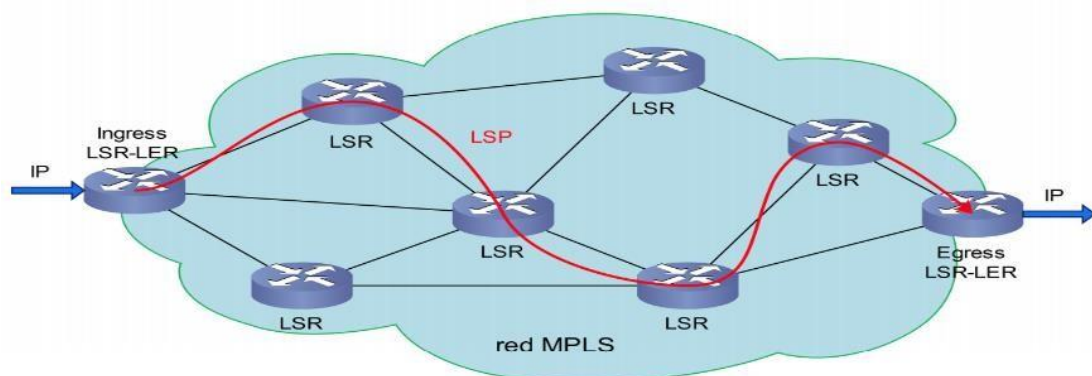
|                          |  |  |   |
|--------------------------|--|--|---|
| TIPO DE CONEXIÓN         | Posee una Multiplexación estadística, es un protocolo orientado a la conexión y posee un balanceo de carga punto a punto                         | Su Multiplexación posee una conexión de lógica en la capa 2 del modelo OSI, con esto se elimina un completo nivel de procesamiento | Posee soporte para diferentes SLAs que logra convertir a IP ser orientado a la conexión.            |
| VELOCIDAD DE TRANSMISIÓN | 34Mbps a 622 Mbps  | 64Kbps a 2 Mbps  | Velocidades de Backbone F.O, 155.52 Mbps, a 622 Gbps  |
| CONTROL DE CONGESTIÓN    | Posee el campo PTI (Identificador de tipo de carga), el cual se encargará de señalar los tipos de congestión según la configuración de los bits. | Utiliza técnicas estadísticas aleatorias, con notificación y descarte a través de los identificadores FECN, BECN, Y DE.            | Utiliza Ingeniería de Tráfico para evaluar los cuellos de botella y proyectar una expansión de red. |

Fuente: (García, 2018)

Realizado por: Navarrete Sebastián, Rivera Patricio. 2019

### 1.1.3. Arquitectura de MPLS

MPLS consta de los siguientes elementos mostrados en la Figura 1-1:



**Figura 1-1.** Arquitectura de MPLS.

Fuente: (García, 2018, p. 16)

#### *1.1.3.1. LSR*

También son conocidos como Routers P, quienes “están en el interior de una red MPLS, generan instrucciones de repartición de etiquetas y además permiten el reenvío de paquetes en el dominio MPLS, mediante el análisis de etiquetas adosadas a cada paquete hasta la capa 2, son los routers que forman el núcleo de la red MPLS” (Apolo & Coral, 2017, p. 18).

#### *1.1.3.2. LER*

Son denominados Routers PE, los que “están en el extremo del dominio MPLS, su función es insertar etiquetas a los paquetes IP para enviarlos dentro del dominio MPLS o removerlos para enviarlos fuera del dominio, considerando hasta la capa 3” (Apolo & Coral, 2017, p. 18). Para la comunicación entre nodos utiliza el protocolo de enrutamiento BGP (protocolo de puerta de enlace de frontera), logrando tener sistemas autónomos entre sus nodos.

#### *1.1.3.3. FEC*

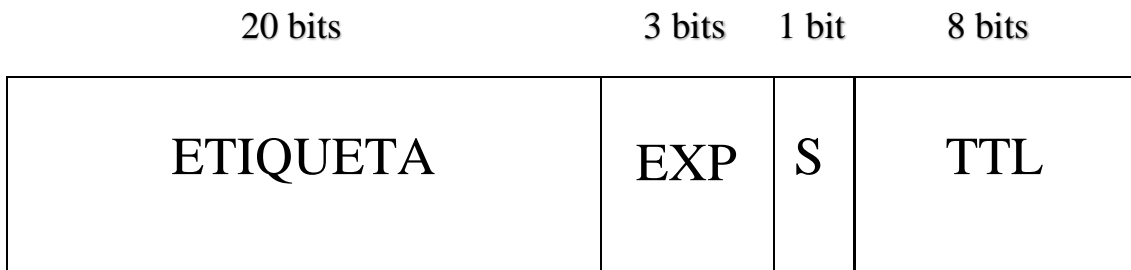
Son un grupo de paquetes que tienen características similares y que se transmiten de manera similar dentro del LSP con diferentes destinos, estos son asignados a un mismo FEC utiliza una etiqueta, y de esta forma brinda una ruta definida por los LSR's.

#### *1.1.3.4. LSP*

Es un camino unidireccional que dirige todos los paquetes etiquetados de acuerdo con un FEC, mediante un conjunto de LSR's para alcanzar el destino. En la creación de un LSP se requiere un protocolo de enrutamiento para difundir las tablas de enrutamiento en todos los LSR's y así determinar el camino más corto hacia el destino (Apolo & Coral, 2017, p. 7)

#### 1.1.4. Cabecera de MPLS

La cabecera está definida por la RFC (Request For Comments-Peticiones de comentarios) 3031, la cual describe la arquitectura de la red MPLS desde el año 2001, y distribuye la cabecera MPLS de cuatro elementos los cuales se muestra en la Figura 2-1 presentada a continuación:



**Figura 2-1.** Arquitectura de MPLS

**Realizado por:** Navarrete Sebastián, Rivera Patricio. 2019

Como se puede apreciar en la Figura 2-1, los primeros 20 bits representan el valor de la etiqueta, la cual identifica una FEC, estos valores pueden estar en la escala de 0 a 9, y se debe tomar en cuenta que los valores hasta 16 pertenecen a un uso específico.

La segunda parte de la cabecera, es decir, de 20 a 22 bits son los 3 bits experimentales, estos bits son los utilizados para la calidad de servicio, fueron denominados así porque históricamente nadie sabía cuál sería su uso.

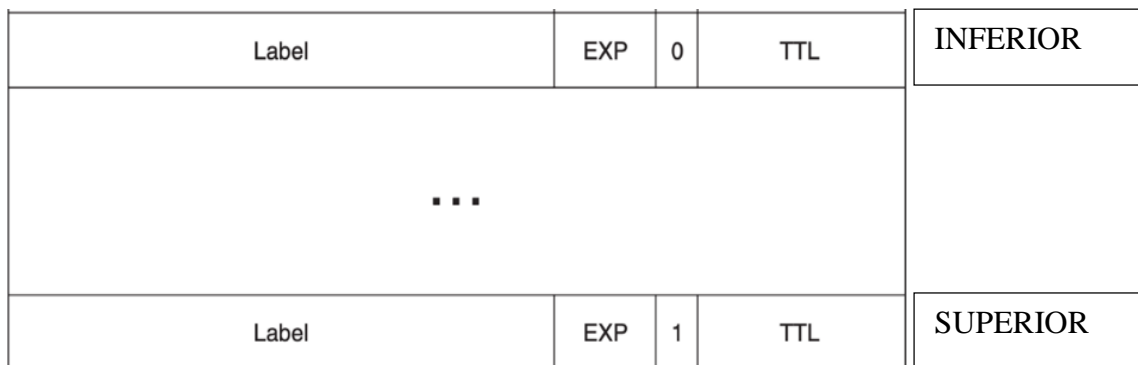
El bit 23 pertenece a la parte inferior de la pila, por lo general posee un valor de 0 siempre y cuando se encuentre como la etiqueta que está en la parte inferior de la pila, si este es el caso, el bit tomará un valor de 1.

La pila es la colección de etiquetas que se encuentran en la parte superior del paquete, esta puede constar de una sola etiqueta o puede tener más. El número de etiquetas que se puede encontrar en la pila es ilimitado, aunque rara vez debería ver una pila que consta de cuatro o más etiquetas (Chuy, 2019, pp. 99-100).

Desde el bit 24 a 31, son los ocho bits utilizados para el TTL o tiempo de vida, este cumple con una tarea similar al que tiene la cabecera IP, es decir, tiene un decremento de 1 al dar un salto en la red, ello evita que los datos se atoren durante la transmisión, cuando un paquete recibe un valor 0 en su etiqueta, este será descartado.

#### 1.1.4.1. Apilamiento de pilas

Los equipos compatibles con MPLS puede necesitar más de una etiqueta en aplicaciones como video, en la parte superior del paquete al levantar la conexión, para lograr este mecanismo se procede a un empacado de las etiquetas en una sola pila, donde la primera etiqueta se denomina etiqueta superior y la última etiqueta inferior, como se aprecia en la Figura 3-1 mostrada a continuación:



**Figura 3-1.** Apilamiento de etiquetas

**Fuente:** (Luc De Ghein, 2016)

En MPLS, cada enrutador estudia la cabecera de cada paquete y realiza el algoritmo respectivo de ingeniería de tráfico, siempre basándose en la información de dicha cabecera, como vimos anteriormente cada cabecera posee información suficiente para poder decidir el próximo salto, y se componen por dos funciones diferentes:

- Utilizar los FECs para particionar el conjunto de paquetes posibles en clases de transmisión iguales.
- Relacionar los FECs, con un destino.

## **1.2. Ingeniería de Tráfico**

Es aquella técnica que nos ayuda a mejorar la utilización de recursos disponibles en una red de datos con el fin de evitar cuellos de botella a causa del envío de tráfico por la ruta más corta, la cual es seleccionada por el protocolo de enrutamiento sin tener en consideración rutas alternas; por lo que, llega a congestionar muchas veces los enlaces dentro de la red.

Gracias a la Ingeniería de Tráfico (IT) podemos proveer calidad de servicio, de modo que, se obtiene un control de uso de los recursos dentro de la red.

La IT, tiene la principal característica de enrutar diferentes paquetes por rutas descongestionadas a pesar de no ser el camino más corto, de forma que, ayuda a mejorar el servicio brindado al usuario. En referencia a ello, Apolo y Coral (2017) indican que, la ingeniería de tráfico provee de las siguientes funciones dentro de una red MPLS, así:

- Definir rutas secundarias en entornos de congestión de la red.
- Asegurar un uso más eficiente del ancho de banda y los recursos de la red, evitando dejar enlaces o segmentos de red inutilizada, sino más bien, tomándolos en cuenta como caminos alternos en escenarios de congestión de recursos sobre utilizados.
- Minimizar la congestión y pérdida de paquetes transportados a través de la red y maximizar la eficiencia de la red.
- Facilitar un adecuado control sobre el sistema de re-enrutamiento de paquetes cuando el camino secundario encuentra puntos de falla (p. 21).

### ***1.2.1. IT orientada a Tráfico***

La ingeniería de tráfico en esta modalidad tiene como objetivo hacer más eficiente el transporte de datos y reducir el retardo, la pérdida de paquetes, maximizar el throughput (tasa de transferencia efectiva) y brindar calidad de servicio para los datos transportados a través de la red (Apolo & Coral, 2017, p. 21).

### ***1.2.2. IT orientada a Recursos***

En el trabajo de Titulación denominado “Análisis y simulación de tráfico de la red de datos de las Fuerzas Armadas con tecnologías MPLS” realizado por Apolo y Coral (2017) los autores indican que la TE orientada a recursos tiene por objetivos disminuir el consumo de hardware en los equipos de la red, gracias a esto se evita que exista alguna congestión durante la comunicación, esta técnica se usa para un mejor uso del ancho de banda. Por cualquiera de los dos métodos descritos se llega a un objetivo común “optimizar los recursos de una red haciendo más eficiente su uso para minimizar la congestión de la red” (Apolo & Coral, 2017, p. 21).

### ***1.2.3. Componentes de Ingeniería de Tráfico***

La ingeniería de tráfico está definida en el RFC 3272, la cual indica que, es una opción para la optimización de recursos en los dispositivos de una red MPLS, dicha tecnología está contenida por los parámetros que permita tener un control sobre el tráfico de la red.

En una red, la congestión se produce cuando el tráfico cursado es mucho mayor a la capacidad de la red, esto degrada la calidad de servicio que brindan los proveedores, y es así que dicha congestión puede producir: retardo de paquetes o pérdida de información por descarte de paquetes.

Para todos estos inconvenientes la solución más viable es la ingeniería de tráfico, la cual se basa en cuatro componentes principales:

1. **Reenvío de paquetes:** Se encarga de enrutar los paquetes IP por un camino específico a través de la red.
2. **Distribución de información:** Es el encargado de obtener información clara de cómo está conformada la red y se obtiene al dar extensión a los IGP (Protocolo de gateway interior) como OSPF o IS IS, con el objetivo de que sus atributos sean incluidos en cada aviso del estado de los enlaces.
3. **Selección de Camino:** En este componente, los routers usan la información generada por la ingeniería de tráfico para poder obtener los caminos en la red MPLS de su propio grupo LSP. De modo que, el camino para cada LSP puede ser representado por una ruta



explícita. El router de ingreso determina el camino físico para cada LSP aplicando un algoritmo de camino más corto basado en restricciones (CSPF) a la información en la base de datos de la ingeniería de tráfico (Apolo & Coral, 2017, p. 11).

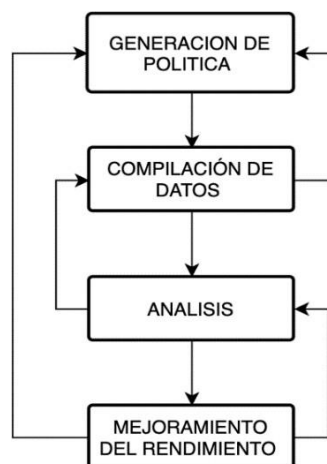
4. **Señalización:** Es la que se encarga de que los LSP establezcan un correcto funcionamiento e intercambio de etiquetas dentro de la red, para esto usa el protocolo RSVP-TE (Protocolo de reserva de recursos).

Por otra parte, cabe resaltar que RSVP-TE dispone de extensiones especiales para establecer rutas ópticas y hacer una red ágil, además es conocido también como un algoritmo basado en restricciones, comúnmente usada como protocolo de señalización y requiere de datagramas IP para comunicarse entre LSR's (Apolo & Coral, 2017, p. 23).

#### ***1.2.4. Proceso de la Ingeniería de Tráfico***

Es importante revisar el modelo del proceso de la ingeniería de tráfico. Este representa las diferentes fases del ciclo de vida de la ingeniería de tráfico en un contexto operacional, contiene cuatro principales fases son: fase de formulación de políticas, fase de recopilación de datos, fase de análisis y caracterización y fase de optimización del rendimiento (Hernández, 2015, p. 24).

Estas cuatro fases se pueden apreciar en la Figura 4-1 mostrada a continuación:



**Figura 4-1.** Fases de Ingeniería de Tráfico

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

- **Generación de Políticas:** Están directamente relacionadas con la topología que contienen la red, el patrón de la industria, la organización de costos, políticas imperantes técnicas de mejoramiento. Este proceso se utiliza para la definición de TE estratégica con una topología virtual en los ISPs, o si la optimización está orientada al rendimiento de la red.
- **Recopilación de Datos:** Hace referencia a que “se recogen estadísticas empíricas de la red operativa a través de un sistema de medición” (Hernández, 2015, p. 25). En estas tomas de datos se captura información como patrones de tráfico, utilización del enlace las estadísticas de paquetes descartados o perdidos.
- **Análisis y Caracterización:** Se enfoca principalmente en la evaluación del rendimiento y optimización de recursos de la TE, sirve para entender las fallas que existen en la red.
- **Optimización del rendimiento:** Esta fase implica la toma de decisiones con el fin de mejorar el rendimiento de la red.

En el contexto de MPLS, la fase de optimización puede implicar: (1) la creación de nuevos LSPs y control cuidadoso de las rutas utilizando un apropiado mecanismo de selección de caminos; (2) re-enrutamiento de los LSPs para lograr una distribución del tráfico más equilibrado; (3) desactivación y reactivación de un LSP existente; (4) modificar los parámetros de los LSPs para controlar sus características de comportamiento; (5) la modificación de los atributos asociados a los recursos de red para influir en la colocación de LSPs, entre otros. Esta fase puede implicar también actividades como la modificación de los parámetros de enrutamiento y protocolos de señalización (Hernández, T. 2015, p. 26).

### ***1.2.5. Algoritmos de Ingeniería de Tráfico***

Como se menciona en el presente trabajo de investigación, el objetivo de la ingeniería de tráfico es reducir los tiempos de la transmisión de datos, al igual que evitar los cuellos de botella y conseguir una red MPLS más eficiente. Para cumplir con estos objetivos se puede usar diferentes algoritmos de ingeniería de tráfico, entre los más conocidos tenemos: CSPF, MIRA y FAIR NETWORKS.

### *1.2.5.1. Algoritmo CSPF (Constraint Shortest Path First)*

Está basado en un algoritmo dedicado a optimizar una métrica escalar específica. Para el caso del protocolo Open Shortest Path First (OSPF) el administrador de red asigna a cada enlace en la red una métrica administrativa. Este protocolo de enrutamiento trabaja con el algoritmo de Dijkstra el cual está basado en el camino más corto, y es conocido en inglés como Shortest Path First (SPF).

Para el caso de algoritmos de ruteo basados en restricciones, Constraint Based Routing (CBR), se requiere la habilidad de calcular un camino óptimo respecto a alguna métrica escalar y que no altere las restricciones.

El algoritmo SPF plano, deduce la ruta más adecuada en relación a su métrica escalar. Entonces, para calcular un camino que no viole dichas restricciones, todo lo que necesitamos es modificar el algoritmo de manera tal que pueda tomar en cuenta esas restricciones (Delfino, Rivero & San Martín, 2006, pp. 33-34).

Dentro de las restricciones más comunes tenemos:

- **Restricción por ancho de banda.** En este caso se evalúa si la disponibilidad del ancho de banda es mucho mayor al especificado por la restricción, de ser este el caso se puede examinar el nodo ubicado al otro extremo del enlace.
- **Restricción basada en el tipo de ruteo.** Para usar este tipo de restricciones es necesario el uso de protocolos de estado de enlace como OSPF o IS IS.

### *1.2.5.2. Algoritmo MIRA (Minimum Interference Routing Algorithm)*

El objetivo es minimizar la 'interferencia' que provoca el establecimiento de un nuevo LSP a potenciales nuevos LSPs que son desconocidos. Mapear un LSP en la red puede reducir el máximo ancho de banda disponible entre algunos pares de nodos ingreso-egreso críticos en la red, dependiendo de por dónde se dirija el mismo. Este fenómeno es conocido como "interferencia". (Delfino, Rivero & San Martín, 2006, p. 37).

Si los caminos que reducen el ancho de banda entre nodos ingreso-egreso son evitados entonces la creación de cuellos de botella se evitará también. En otras palabras, dada una nueva demanda entre los nodos de adyacencia, se considera el impacto de mapear esta demanda, en futuras demandas entre nodos ingreso-egreso. Dicho impacto se caracteriza al asignarle pesos a los enlaces que pueden ser usados por estas futuras demandas y pasando luego a calcular SPF con dichos pesos (Delfino, Rivero & San Martín, 2006, p. 39).

Para el uso del algoritmo mencionado se debe conocer la topología de la red, y la capacidad que esta posee, dicho conocimiento debe abarcar información como el protocolo de enrutamiento configurado en la red y los enlaces conocidos dentro de la misma.

#### *1.2.5.3. Algoritmo FAIR NETWORKS (Redes Justas)*

Debido a que la distribución del tráfico en las redes de datos cambia rápidamente, es complicado para los operadores de una red, el determinar cuándo y dónde incrementar la capacidad de esta, lo cual puede llevar a problemas de sobrecarga (Delfino, Rivero & San Martín, 2006, p. 40).

Las tomas de decisión sobre cuántos usuarios pueden ser atendidos y cuánto tráfico puede generar cada uno de ellos, es un tema del que se encarga el proceso de admisión de tráfico. El mismo, es responsable de garantizar que los usuarios tengan un acceso justo a los servicios de la red, al mismo tiempo de minimizar la probabilidad de negar un servicio a un cliente (Delfino, Rivero & San Martín, 2006, p. 40).

Fair Networks se encarga de determinar cuánto tráfico de cada flujo debe ser admitido por la red y por dónde rutear al mismo, una vez que ingresa, satisfaciendo los requerimientos de alta utilización de la red y garantizando justicia a los usuarios, es uno de los retos más grandes en el diseño de las redes de telecomunicaciones de hoy en día (Delfino, Rivero, & San Martín, 2006, p. 40).

Frente a ello, la pregunta que nos debemos hacer es la siguiente: ¿qué principio debemos seguir al alojar las demandas entre los recursos que la red tiene para ofrecer, de manera de cumplir con algún criterio de justicia? Una posible respuesta es utilizar el esquema de asignación de recursos llamado Max-Min Fairness (MMF) (Delfino, Rivero & San Martín, 2006, p. 40).

## **1.3. Segment Routing SR**

### ***1.3.1. Introducción***

Tradicionalmente, la dirección del tráfico dentro de las redes de proveedores de servicios estaba basada principalmente en una búsqueda de la dirección IP en cada enrutador que se encontraba en el camino hacia el destino. Los protocolos IGP se utilizan para distribuir la información de topología y calcular las rutas más cortas para poder realizar el envío de los paquetes. En la última década, la dependencia de muchas empresas del rendimiento de la red y la disponibilidad se hizo cada vez más relevante.

Brindar un servicio estricto acorde del nivel (SLA) en términos de pérdida de paquetes, demoras, fluctuaciones, y el ancho de banda disponible se convirtió en un diferenciador comercial clave. Estos requisitos impulsaron la evolución de la red hacia arquitecturas que permitieron dirigir el tráfico con más flexibilidad (Filsfils, Kumar, Pignataro, Cardona & Francois, 2015, p. 1).

En los años 90, con la introducción de la tecnología MPLS el poderoso mecanismo de túnel y su funcionalidad, fue extraordinario para dicha tecnología. Sin embargo, la Ingeniería de Tráfico en MPLS se enfrenta a diferentes desafíos como son: las deficientes características de equilibrio dentro del modelo RSVP-TE (Protocolo de reserva de recursos - Ingeniería de tráfico), ya que esta no se adapta a la verdadera realidad de IP; para poder superar este inconveniente, el modelo RSVP-TE necesita poder replicarse, pero esto complica la dificultad para monitorear la red.

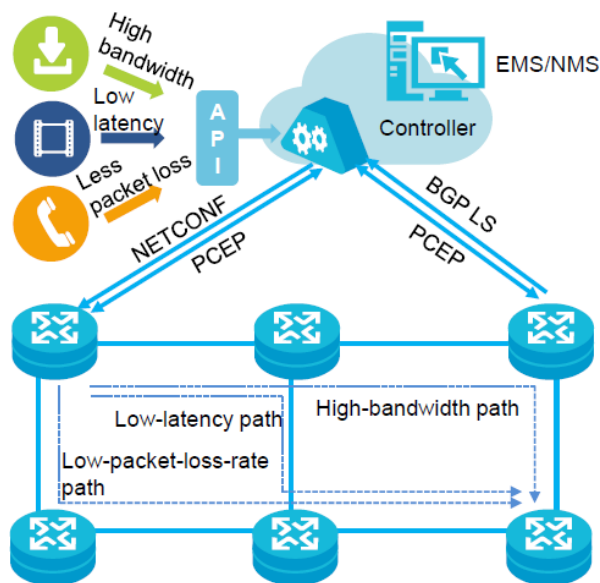
Con SR, el tráfico se puede dirigir para ello utiliza construcciones de reenvío, estos se benefician de un ECMP (Protocolo de enrutamiento Multipath), de manera que una sola configuración de túnel SR puede capturar un conjunto de rutas. Además, los problemas de escalabilidad en los planos de control y de datos, son causados por un estado requerido en cada salto a lo largo de cualquier ruta explícita. Si bien los operadores ya expresaron su preocupación por estos problemas de escala, se anticipa que esta situación se amplificaría para las redes fusionadas que ofrecen una gran cartera de servicios.

El modelo de implementación observado RSVP-TE, se basa en el cálculo distribuido, esto conduce a una ubicación impredecible del tráfico, al uso no óptimo de los recursos y a una lenta

optimización (Filsfils, Kumar, Pignataro, Cardona & Francois, 2015, p. 1). Hay que tomar en cuenta que cada segmento es una etiqueta, y Segment Routing es un enfoque relativamente nuevo para la ingeniería de tráfico, ya que su principal idea es dividir el camino de enrutamiento en diversos segmentos, de modo que, puede controlar de mejor manera el enrutamiento, y se utiliza de forma más eficiente la red.

El enrutamiento de segmento (SR) es un protocolo diseñado para reenviar paquetes de datos en una red en función de las rutas de origen, este divide una ruta de red en varios segmentos y asigna un ID de segmento (SID) a cada segmento y nodo de reenvío. Los segmentos y nodos se ordenan secuencialmente en un listado para formar una ruta de reenvío (Huawei, 2019, p. 1).

En la Figura 5-1 se puede apreciar una red impulsada por servicios donde las rutas explícitas se calculan en función de los requisitos de las aplicaciones.



**Figura 5-1.** Fases de Ingeniería de Tráfico

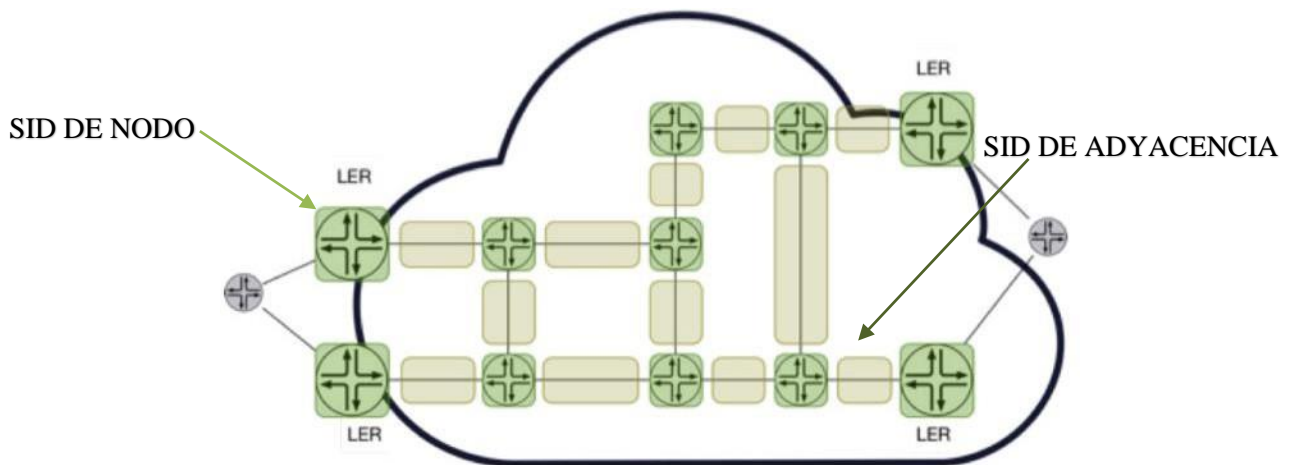
Fuente: (Huawei, 2019, p. 1)

Dentro de Segment Routing se puede ubicar dos tipos básicos de segmentos que son: el segmento de nodo y el segmento de adyacencia. De modo que “el segmento de nodo identifica un nodo de enrutador y en segmento de adyacencia representa una interfaz local de un nodo” (Bhatia, Hao, Kodialam, & Lakshman, 2015, p. 1).

### 1.3.2. Operación de Segment Routing

Segment Routing opera dividiendo la red en segmentos, en la cual se le asigna un SID (Identificador de Segmento), de esta forma es anunciado por el protocolo de enrutamiento estándar, mismo que puede ser: ISIS, OSPF, o BGP, de esta manera se elimina la necesidad de la ejecución de protocolos adicionales con distribución de etiquetas como LDP TDP, etc.

En la Figura 6-1 se puede apreciar las adyacencias mediante los SID Globales:



**Figura 6-1.** Segment Routing y SID de adyacencia

Fuente: (Arista, 2019, p. 2)

Cabe resaltar que “estos SID son únicos globalmente o SID localmente significativos” (Arista, 2019, p. 1). Como se evidenció en la Figura 6-1 los SID de nodo significativo se presentan en un color verde oscuro y los SID de adyacencia local están en un color verde claro.

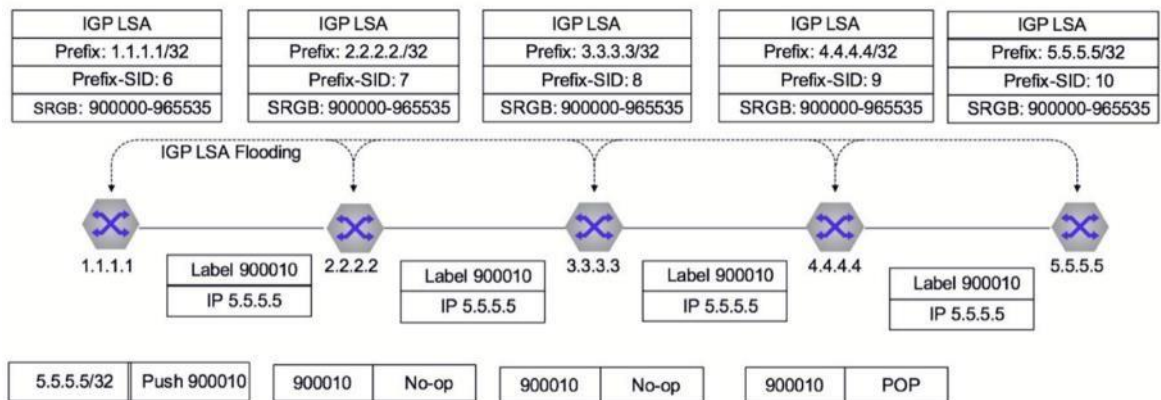
Por otro lado, los SID globales tienen tres variantes principales:

1. **Prefijo SID:** Es el identificador de segmento asociado con un prefijo
2. **Nodo SID:** Es un identificador de segmento asignado un nodo específico
3. **Anycast SID:** Es un ID de segmento compartido asignado a una loopback.

Frente a esto, Arista (2019) en el artículo científico de Segment Routing, manifiesta que:

“Los SID globalmente además de ser significativos deben ser únicos y se le puede asignar un valor absoluto del Bloque global de enrutamiento de segmentos (SRGB). Alternativamente, el valor SID puede ser derivado por cada nodo usando un valor de índice y agregando esto al valor base de SRGB” (Arista, 2019, p. 1).

Dentro del dominio se instalan un valor único del prefijo SID, al ser valores únicos a nivel mundial reducen la explosión del plano de datos MPLS, en este caso el valor de SID inunda todo el dominio IGP, y utiliza el mismo rango SRGB. Además, cada nodo deriva el mismo valor de etiqueta del SID para los nodos remotos, estos parámetros se pueden evidenciar en la Figura 7-1 presentada a continuación:



**Figura 7-1.** Prefijo SR o Nodo SID Señalización y reenvío utilizando IGP

Fuente: (Arista, 2019, p. 2)

En la figura también se muestra las operaciones de: Push, swap y pop, quienes se encargan de cumplir con las siguientes funciones:

- **Pop:** Elimina la etiqueta superior. El paquete se reenvía con la pila de etiquetas restante o como un paquete sin etiqueta, esta operación es usada comúnmente por el enrutador de regreso excepto en casos de PHP (Chuy, 2019, p.129).
- **Swap:** Elimina la etiqueta superior y se reemplaza con una nueva etiqueta, esta operación se da en los routers P de la arquitectura MPLS.



- **Push:** Reemplaza la etiqueta superior por una nueva etiqueta y se agregan más etiquetas en la parte superior de la etiqueta intercambiada, esta operación ocurre en los routers PE de la arquitectura MPLS (Chuy, 2019, p. 130).

El enrutamiento de segmentos también conserva la noción de SID localmente significativos para permitir que la ingeniería de tráfico sea más precisa. Los SID de adyacencia son localmente significativos para cada nodo y se anuncian e instalan solo en vecinos conectados directamente, de forma que se identifica un enlace de adyacencia específico.

Los SID de adyacencia son similares a las etiquetas asignadas mediante LDP (Label Distribution Protocol), estas son localmente significativas y se instalan directamente de manera adyacente. Los SID de adyacencia se pueden usar en rutas de ingeniería de tráfico de segmento de enrutamiento como una entrada en la lista de pedidos, o pila de etiquetas (Arista, 2019, p. 3).

### ***1.3.3. Segment Routing frente a los protocolos de señalización de etiquetas existentes***

Uno de los objetivos principales de segment routing es eliminar la necesidad de una señalización y distribución de etiquetas mediante protocolos como LDP o RSVP-TE, esto nos permite aprovechar el protocolo de enrutamiento para poder distribuir de una mejor forma sus etiquetas.

#### ***1.3.3.1. Segment Routing frente a LDP***

En segment routing la distribución de etiquetas es 'plug and play' (conectar y usar) permitiéndonos tener una distribución automática de etiquetas, esto, además de facilitar la configuración, permite que las etiquetas se anuncien automáticamente entre los enrutadores en el emparejamiento de adyacencia.

Frente a ello, Arista (2019) afirma que “LDP se basa en el estado del IGP, las características de LDP como la sincronización LDP-IGP fueron introducidos para que estos siempre permanezcan sincronizados, reduciendo la posibilidad de un bloqueo de tráfico” (p. 3). En la actualidad, segment routing los nodos y prefijos mantienen una etiqueta global única que se asigna para el

dominio en general, a diferencia de LDP, las etiquetas son localmente significativas, por lo que se otorga un valor único para cada salto.

Las etiquetas globales reducen significativamente los datos del plano de control en cada salto de la red, estas dependen de: las opciones de implementación del proveedor y de una implementación con LSP (Proveedor de lenguaje de servicio) independiente a LDP. El modo de control puede proliferar un control innecesario en el plano de datos, lo que provoca desafíos de escala (Arista, 2019, p. 3).

### *1.3.3.2. Segment Routing frente a RSVP-TE*

Una de las razones por las que RSVP-TE es muy usado actualmente es por el cambio rápido de ruta. Segment Routing mantiene una escalabilidad y flexibilidad mucho más amplia en ingeniería de tráfico.

Por otro lado, la ingeniería de tráfico con RSVP-TE nos permite:

- Cálculo de rutas mediante una restricción de ancho de banda, la cual utiliza un grupo de riesgos de enlace compartido y señalización de ruta explícita.
- Reserva y seguimiento de ancho de banda, para una mejor utilización de enlaces a larga distancia.
- Seguridad en los enlaces y nodos en caso de falla a través de las capacidades de FRR (Cambio de ruta rápido).

A pesar de esto “muchas implementaciones usan rutas explícitas configuradas estáticamente que usan herramientas de cálculo para rutas sin conexión o sin reserva de ancho de banda. En muchas implementaciones. Las redes RSVP-TE se crean utilizando una malla completa de punto a punto” (Arista, 2019, p. 4).

Los Túneles TE, que son los encargados del control en el plano de datos por cada túnel transitado, son inherentes en la red. También, ante cualquier error en el núcleo de la red, los LSP pueden volver a señalarse, ocasionando un cambio de estado de rotación y un nuevo plano de control.

Según el estudio de la Empresa Arista Networks (2019) sobre Segment Routing manifiesta que:

El enrutamiento de segmentos aborda ciertos desafíos al adoptar un enfoque centralizado en SDN (redes definidas por software) para la ingeniería de tráfico. Es decir, en lugar de encapsular la ruta explícita enrutada de origen en el plano de control, define la ruta de ingeniería de tráfico como una lista de SID encapsulados en el plano de datos.

Solo los enrutadores de borde o de entrada (LER) quienes imponen la ruta de ingeniería de tráfico, como una pila de etiquetas MPLS, así como también, dejan el núcleo libre del plano de control de flujo.

El controlador centralizado puede utilizar cualquier cantidad de restricciones o atributos para definir la ruta TE a más de los atributos ofrecidos en RSVP-TE. Con ese fin, el enrutamiento de segmento ofrece una escalabilidad mejorada y proporciona el control granular requerido para ingeniería de tráfico en redes modernas (Arista, 2019, p. 4).

#### ***1.3.4. Arquitectura Segment Routing***

Segment Routing está compuesto principalmente por dos planos, los cuales son:

- a. El plano de datos SR.** Definido para codificar la secuencia de segmentos que se podrán aplicar en un paquete con su semántica respectiva. Es decir, la manera en la que cada dispositivo debe procesar un paquete, esta operación descrita en SR deberá ser independiente del protocolo de enrutamiento real, para poder llevar la información del encabezado de SR.
  
- b. El plano de control.** Será el encargado de definir los identificadores para el segmento entre los diversos dispositivos de red, estos dispositivos tienen las indicaciones para aplicar secuencias dadas de segmentos en un flujo.

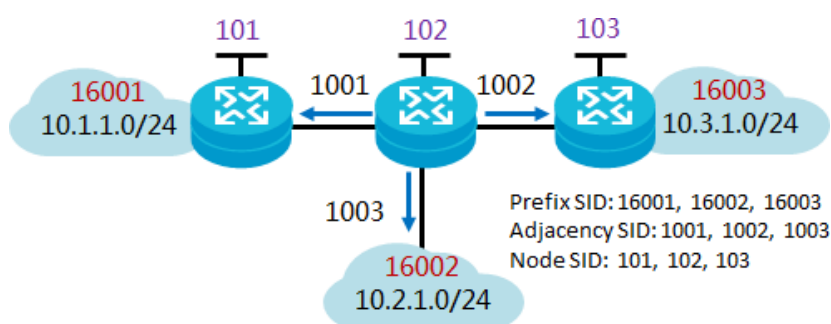
### 1.3.4.1. Plano de Datos

Al respecto, este plano “contiene una secuencia de segmentos y un puntero de segmento activo del paquete, teniendo la instrucción que debe ser ejecutada por el dispositivo que procesa el paquete” (Filsfils, Kumar, Pignataro, Cardona & Francois, 2015, p. 2).

El plano de control contiene los siguientes elementos:

- **NODO SID:** Es la encargada de reenviar el paquete por la ruta más corta hacia el nodo asociado con el ID del segmento, este nodo será asignado por cada enrutador y se puede hacer manualmente o con un control centralizado.
- **SID de adyacencia:** Se encarga de reenviar el paquete con SID adyacente correspondiente, estos serán asignados para cada enrutador con un ID localmente significativo en cada uno de los IGP.
- **SID de servicio:** Se encarga de entregar el paquete por el nodo que se procesa, estos nodos son asignados con un ID, en cada servicio brindado por la red.

Los nodos de Servicio, adyacencia y el SID se puede apreciar en la Figura 8-1 mostrada a continuación:



**Figura 8-1.** Plano de Datos

Fuente: (Huawei, 2019, p. 1)

En términos simples, un segmento de prefijo indica una dirección de destino, y un segmento de adyacencia indica un enlace para paquetes de datos salientes. Los segmentos de prefijo y adyacencia son similares a la dirección IP de destino y la interfaz de salida en el reenvío IP

convencional respectivamente. En un área IGP, un PE propaga su SID de nodo y su SID de adyacencia a través de mensajes IGP extendidos, de tal modo que cualquier PE en el área pueda obtener información sobre los otros Pes (Huawei, 2019, p. 1).

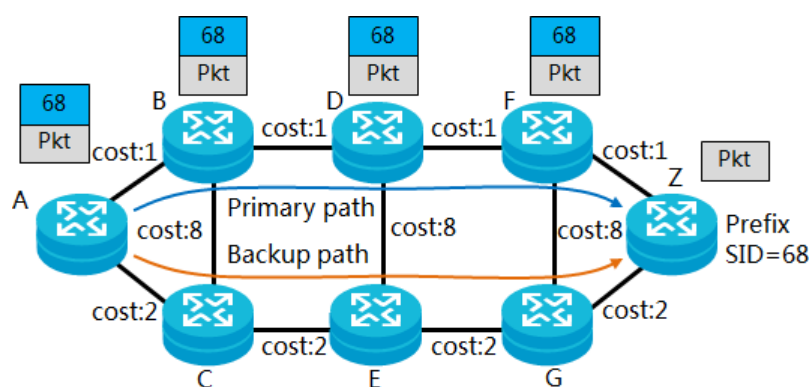
La combinación entre nodos y SID se pueden implementar en todo tipo de red, en cada salto se identifica el próximo salto, esto se basa en la información del segmento en la pila superior.

Si el SID superior identifica otro nodo, el extremo de recepción reenvía el paquete de datos recibido a ese nodo a través del enrutamiento de ruta múltiple de igual costo (ECMP). Si el SID superior identifica el nodo local, el extremo de recepción elimina el SID superior y continúa con el siguiente procedimiento (Huawei, 2019, p. 1)

Estos prefijos se usan de forma aleatoria o combinada. Los modos principales son:

- **Modo basado en segmentos de prefijo:** Un IGP utiliza el algoritmo de la ruta más corta primero (SPF) para calcular la mejor ruta. Este modo también se llama Segment Routing-Best Effort (SR-BE) (Huawei, 2019, p. 1).

Un ejemplo de este modo es el que se muestra en la Figura 9-1 mostrada a continuación:

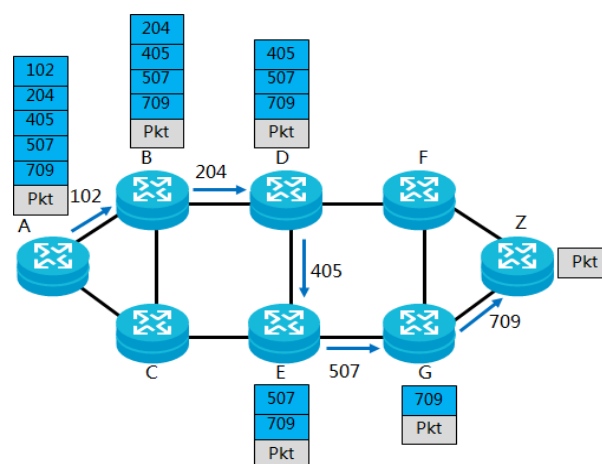


**Figura 9-1.** Modo Basado en segmento de prefijo

Fuente: (Huawei, 2019, p. 1)

El nodo Z está conectado a la red de destino con un prefijo SID de 68; después un IGP propaga el prefijo SID, cada nodo en el área IGP aprende el prefijo SID de la red desde el nodo Z y luego se ejecuta SPF para calcular la ruta más corta a la red (Huawei, 2019, p. 1).

- **Modo basado en segmentos de adyacencia:** Se asigna un fragmento de adyacencia a cada unión en la red, y se genera un listado de segmentos con variados segmentos de adyacencia en la entrada, de modo que cualquier ruta explícita estricta se pueda especificar, como se muestra en la Figura 10-1 mostrada a continuación:

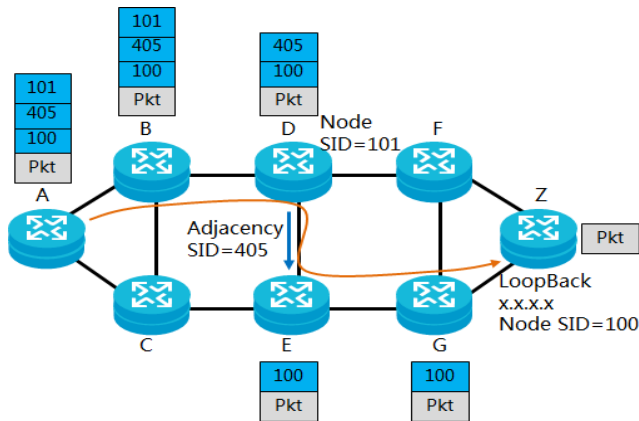


**Figura 10-1.** Modo Basado en segmentos de adyacencia

Fuente: (Huawei, 2019, p. 1)

En este modo, el ajuste de ruta y la optimización del tráfico se pueden implementar de manera centralizada, facilitando la implementación de redes definidas por software (SDN). Este modo se utiliza principalmente para la Ingeniería de tráfico y enrutamiento de segmentos (SR-TE) (Huawei, 2019, p. 1).

- **Modo en el que se combinan los segmentos de adyacencia y nodo:** Los segmentos se combinan, y el segmento de adyacencia permite que una ruta contenga por una adyacencia específica, este modo se usa principalmente para SR-TE. Como se evidencia en la Figura 11-1 presentada a continuación:



**Figura 11-1.** Modo Basado en el que se combinan los segmentos de adyacencia y nodo

Fuente: (Huawei, 2019, p. 1)

Cabe recalcar que “los nodos pueden ejecutar SPF para calcular la ruta más corta, se basa en segmentos de nodo y establecen múltiples rutas para el tráfico con balanceo de carga. Las rutas calculadas en este modo no son estrictamente fijas” (Huawei, 2019, p. 1).

#### 1.3.4.2. Plano de control

Es el encargado de definir como se comunica la información del ID del segmento entre los dispositivos de la red, para este proceso se anunciará una red SR, un nodo y un SID de adyacencia mediante el IGP, también otro elemento importante del Plano de Control es cómo un enrutador recibe instrucciones para seleccionar la ruta SR que el paquete debe seguir utilizando los siguientes métodos:

- **Cálculo de SPF restringido distribuido (CSPF):** Para este enfoque, un enrutador de ingreso calcula el camino más corto para un destino, bajo la restricción que esta ruta coincida con algunos criterios y después calcula una secuencia de segmentos de nodo y adyacencia que codifican esta ruta. (Filsfils, Kumar, Pignataro, Cardona & Francois, 2015, p. 3).
- **Enfoque basado en el controlador SDN:** Es el que proporciona un plano de datos escalable y resistente al tiempo que permite la flexibilidad de control comúnmente asumida para los entornos SDN. Este aspecto condujo al apoyo planificado de SR en

diseños de algunos controladores orientados a SDN. Por ejemplo, Open Daylight admite el control de SR utilizando la ruta de protocolo de elementos de computación (PCEP).

- **Definido estáticamente por el operador:** Es una configuración estática de los túneles que podrían usarse para fines específicos tales como prueba o solución de problemas, pero generalmente no es recomendado para el funcionamiento de la red a largo plazo, debido a la evidente escala, resistencia y limitaciones (Filsfils, Kumar, Pignataro, Cardona & Francois, 2015, p. 2).

#### 1.3.4.3. Ventajas de Segment Routing

Como se mencionó, Segment Routing es una solución para una necesidad de requerimientos de red cada vez más exigentes, dentro de las principales ventajas tenemos:

- **Plano de control MPLS simplificado:** Segment Routing, no requiere una implementación de LDP o RSVP-TE, al contrario, utiliza el IGP para propagar sus etiquetas y de esta forma calcular sus rutas, esto lo logra sin modificar la arquitectura de reenvío. En la Tabla 2-1 mostrada a continuación, compara el enrutamiento entre MPLS y Segment routing, así:

**Tabla 2-1:** Comparación entre Segment Routing y MPLS

| Artículo                  | Enrutamiento de segmento   | MPLS   |
|---------------------------|--|--|
| Protocolo de control      | IGP  | LDP / RSVP-TE / BGP / IGP  |
| Distribución de etiquetas | Se asigna una etiqueta a cada adyacencia o nodo, y la cantidad de etiquetas a distribuir es independiente de la cantidad de túneles, lo que reduce la cantidad de recursos necesarios. | El número de etiquetas a distribuir aumenta con el número de túneles, lo que requiere una gran cantidad de recursos. |



| Artículo                 | Enrutamiento de segmento  | MPLS  |
|--------------------------|---|---|
| Ajuste de ruta y control | La entrada realiza un recalcu para completar el ajuste y el control de la ruta. | Las configuraciones deben entregarse nodo por nodo para el ajuste y control de la ruta. |

Fuente: (Huawei, 2019)

Realizado por: Navarrete Sebastián, Rivera Patricio. 2019

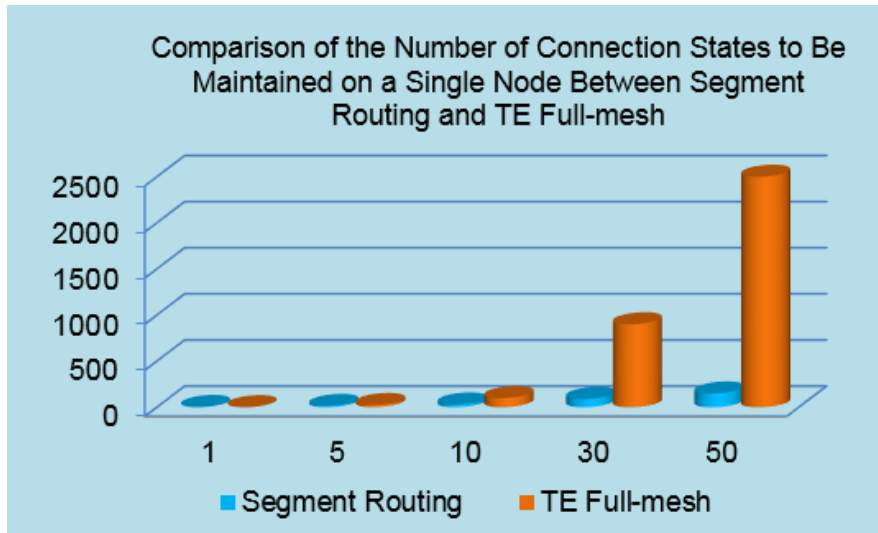
- **Tecnología TE simplificada y capacidad de expansión mejorada:** MPLS TE es una tecnología orientada a la conexión. Para mantener el estado de la conexión, los nodos deben intercambiar una gran cantidad de paquetes de actualización. Esto aumenta la carga del plano de control.

El enrutamiento por segmentos controla las rutas de servicio solo a través de operaciones en las etiquetas de entrada. No requiere nodos de tránsito para mantener la información de ruta, lo que reduce la carga del plano de control (Huawei, 2019, p. 1).

Como se sabe MPLS y SR requieren tener ciertas cantidades de estados de conexión, Huawei afirma que:

- Para MPLS TE, el número de estados de conexión es directamente proporcional al número de túneles TE. Es decir, el número de estados de conexión es igual al número de nodos elevados a la potencia de 2.
- Para el enrutamiento de segmentos, la entrada mantiene el estado del túnel. El número de estados de conexión que se debe mantener es igual a la suma del número de nodos y el número de conexiones (Huawei, 2019, p. 1).

En la Figura 12-1 podemos apreciar la leve comparación del número de estados de conexión que requiere cada uno:

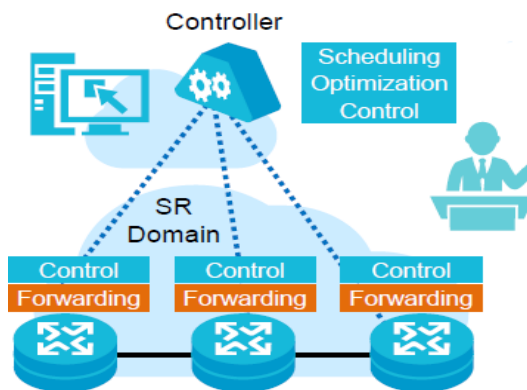


**Figura 12-1.** Número de Estados de Conexión

Fuente: (Huawei, 2019)

- **Evolución más fluida de la red que SDN:** Como se sabe Segment Routing brinda la oportunidad para que los protocolos se puedan extender acorde a la necesidad de la red, con el objetivo que esta pueda evolucionar sin problemas, ya que “la entrada controla y ajusta las rutas de servicio a través de la tecnología de enrutamiento de origen, lo que permite que la red responda rápidamente a los requisitos de las aplicaciones de capa superior” (Huawei, 2019, p. 1).

Se puede mantener un control troncalizado, además que el control y renvío distribuido previene que el rendimiento del controlador se convierta en un cuello de botella, como se aprecia en la Figura 13-1:



**Figura 13-1.** Optimización de Control

Fuente: (Huawei, 2019)

### ***1.3.5. Aplicaciones de Segment Routing***

Segment Routing es considerada una tecnología joven, sin embargo, las grandes empresas como Cisco, Huawei, Juniper, Nokia ya han incorporado a sus dispositivos esta tecnología, por lo cual Huawei en su página oficial presenta las siguientes aplicaciones para Segment Routing:

#### ***1.3.5.1. SR-BE***

Al respecto de SR- BE, Huawei (2019) indica que una vez establecidos los túneles SR, el tráfico de servicio debe dirigirse a los túneles. Este proceso se llama dirección del tráfico. Como los túneles SR-BE no tienen interfaces de túnel, el modo de dirección del tráfico de SR-BE es diferente del de SR-TE (Huawei, 2019, p. 1).

SR-BE admite los siguientes modos de dirección de tráfico:

- **Política de túnel:** Utilice una política de priorización de túnel para seleccionar túneles SR-BE.
- **Ruta estática:** Al configurar una ruta estática, especifique el próximo salto como la dirección de destino de un túnel SR-BE y configure la ruta para que vuelva al túnel SR-BE en función del siguiente salto.
- **Recurrencia de ruta IP a túneles:** Configure una ruta IP pública, para recurrir a un túnel SR-BE en función del siguiente salto de la ruta. (Huawei, 2019, p. 1).

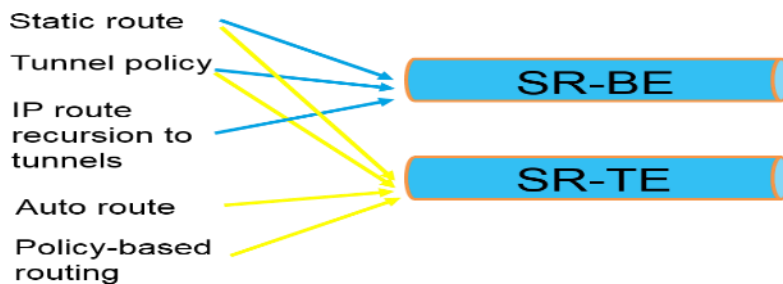
#### ***1.3.5.2. SR-TE***

SR-TE admite los siguientes modos de dirección de tráfico:

- **Política de túnel:** utilice una política de priorización de tipo túnel para seleccionar túneles SR-TE o una política de enlace de túnel para unir túneles SR-TE.
- **Ruta estática:** al configurar una ruta estática, especifique la interfaz de salida de la ruta como una interfaz de túnel SR-TE.

- **Ruta automática:** al configurar un IGP, se utiliza una ruta automática relacionada a un túnel SR-TE que funcionará como un enlace lógico para calcular una ruta.
- **Enrutamiento basado en políticas:** configurar la interfaz de salida en una cláusula de aplicación, esto se utiliza como una interfaz de túnel SR-TE (Huawei, 2019, p. 1).

Para un mejor entendimiento podemos apreciar los diferentes modos de dirección de tráfico de Segment Routing en la Figura 14-1 presentada a continuación:



**Figura 14-1.** Modos de Dirección del Tráfico

Fuente: (Huawei, 2019)

## 1.4. Software para pruebas

Para realizar pruebas de la tecnología propuesta en el presente trabajo de investigación se ha visto en la necesidad de usar ciertas aplicaciones de prueba que listamos a continuación:

- Simulador de Redes
- Generadores de Tráfico
- Analizadores de Tráfico

### 1.4.1. *Simulador de Redes*

Para probar una red de datos sin generar mayores costos, lo más recomendable es usar un simulador, así podremos descartar posibles errores lo que garantiza no tener fallas en la implementación física. En la actualidad existe una diversa cantidad de simuladores e incluso algunos han sido desarrollados por grandes marcas de equipos de comunicación como son:

- Cisco Packet Tracer
- eNSP de Huawei Technologies

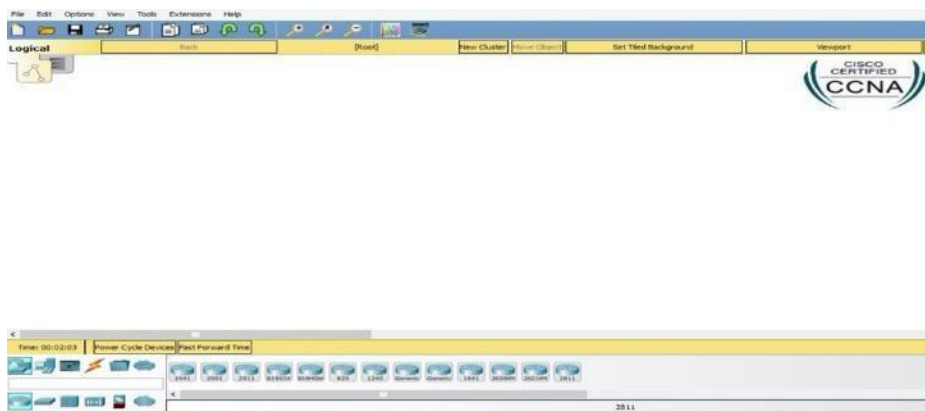
Por otra parte, otros son desarrollados por empresas como EVE-N6, VIRL, entre otros. El más conocido es GNS3 debido a su capacidad de simular IOS de varias marcas de equipos de comunicación además de brindar otras ventajas.

#### *1.4.1.1. Cisco Packet Tracer*

La página Oficial de Cisco Academy afirma indica que “Cisco desarrolló Packet Tracer para que los alumnos de la Networking Academy tengan la mejor experiencia de aprendizaje posible y a la vez adquieran habilidades prácticas en tecnología de redes” (Cisco Networking Academy, 2019, p. 1).

Packet Tracer es una potente plataforma simulación de redes que estimula a sus alumnos experimentar con el comportamiento de las redes y formular preguntas sobre situaciones hipotéticas. Funciona como complemento de los equipos físicos en el salón de clases: los estudiantes pueden crear una red con un número casi ilimitado de dispositivos, lo que estimula la práctica la detección y solución de problemas (Cisco Networking Academy, 2019, p. 1).

En la Figura 15-1 podemos apreciar el entorno de Cisco Packet Tracer:



**Figura 15-1.** Interfaz de Cisco Packet Tracer

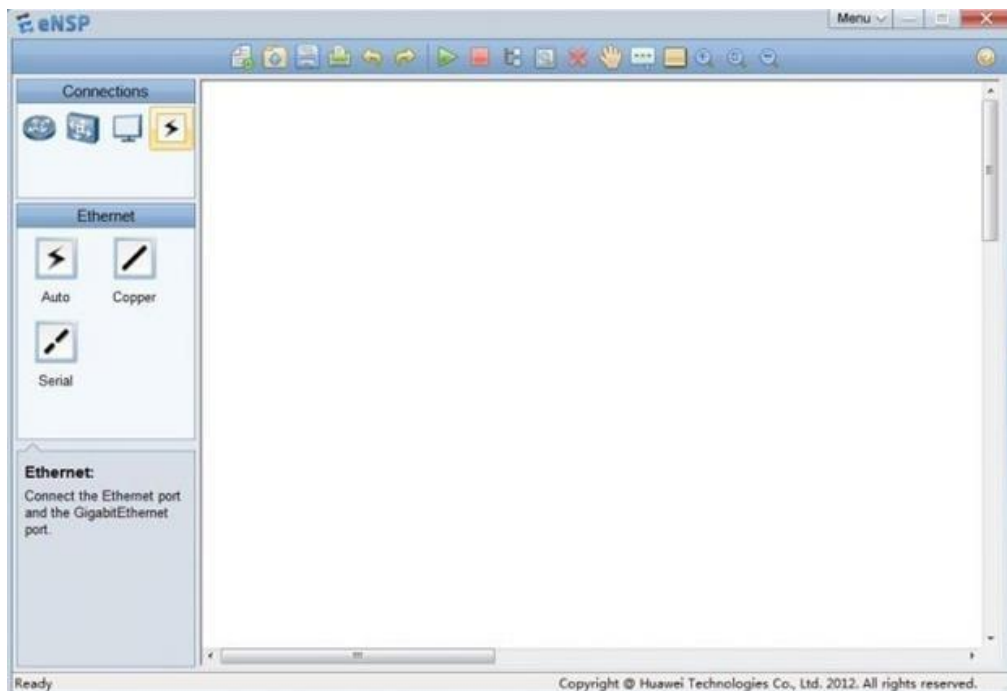
**Realizado por:** Navarrete Sebastián, Rivera Patricio. 2019

### 1.4.1.2. eNSP Huawei Technologies

A Juicio de Huawei Technologies (2019) afirman que:

eNSP es una plataforma simulación de gráfica para equipos de red reales, esta plataforma ayuda a los profesionales y clientes de las TIC a familiarizarse rápidamente con los productos Huawei Datacom, además, ayuda a comprender y dominar el funcionamiento y la configuración de los productos, para mejorar la red de datos en las empresas (Huawei, 2019, p. 1).

eNSP tiene una ventaja en particular la cual es que se puede conectar su interfaz con GNS3, el entorno de eSNP lo podemos apreciar en la Figura 16-1 mostrada a continuación:



**Figura 16-1:** Interfaz de eNSP

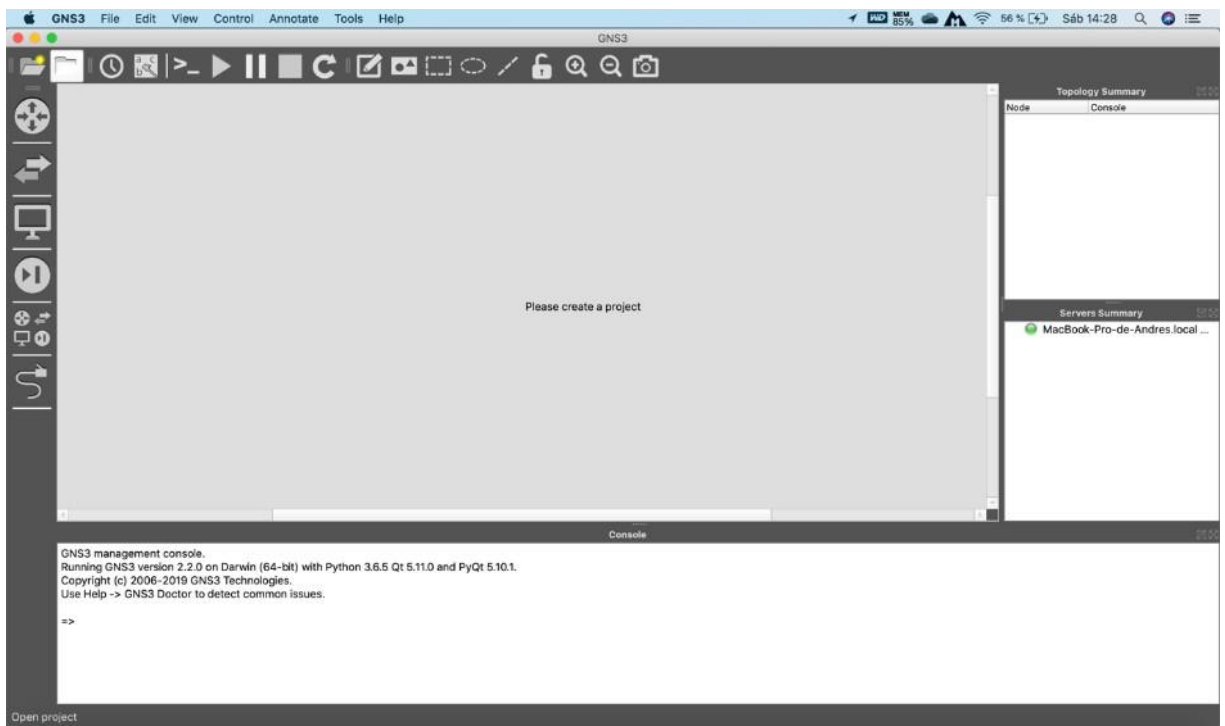
**Realizado por:** Navarrete Sebastián, Rivera Patricio. 2019

### 1.4.1.3. GNS3

En la actualidad Gns3, es quizá el simulador de redes más grande y uno de los más conocidos, por su gran cantidad de aplicaciones es un software que nos permite realizar diversos tipos de

simulaciones entre varias plataformas, ya sea Open Source o de paga, al igual que nos permite simular equipos de comunicación de casi todas las grandes marcas conocidas.

Fue desarrollado en 2008, es utilizado por una gran cantidad de grandes empresas como Exxon, Walmart, AT&T, inclusive la NASA, además es muy usada para las preparaciones de exámenes para certificación internacional como CCNA, CCNP o CCIE, hasta el 2015 el software ha sido descargado por más de 11 millones de veces a nivel mundial, el entorno de GNS3 se puede apreciar en la Figura 17-1 presentada a continuación:



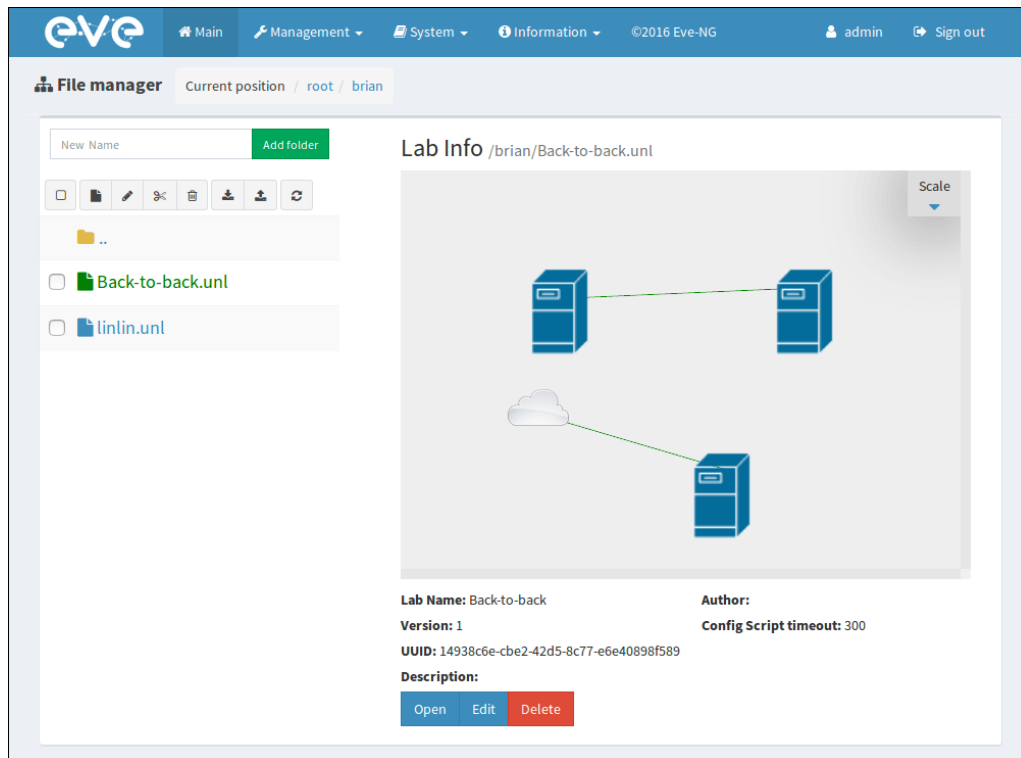
**Figura 17-1.** Interfaz de GNS3

**Realizado por:** Navarrete Sebastián, Rivera Patricio. 2019

#### *1.4.1.4. EVE-NG (Emulated Virtual Enviroment)*

EVE-NG PRO es el primer software de emulación de red de múltiples proveedores sin cliente que brinda a los profesionales de redes y seguridad grandes oportunidades en el mundo de las redes. Las opciones de administración sin cliente permitirán que EVE-NG PRO sea la mejor opción para los ingenieros empresariales sin influencia de las políticas de seguridad corporativas, ya que puede ejecutarse en un entorno completamente aislado. (EVE-NG Ltd, 2020, p. 1).

En la Figura 18-1 que se presenta a continuación, se puede apreciar el interfaz de EVE-NG:



**Figura 18-1:** Interfaz de EVE-NG

**Realizado por:** Navarrete Sebastián, Rivera Patricio. 2019

#### 1.4.1.5. VIRL

Cisco VIRL, o Virtual Internet Routing Labs, es un marco de simulación de red que permite el diseño, la configuración y las pruebas rápidas de soluciones integrales de red y prestación de servicios.

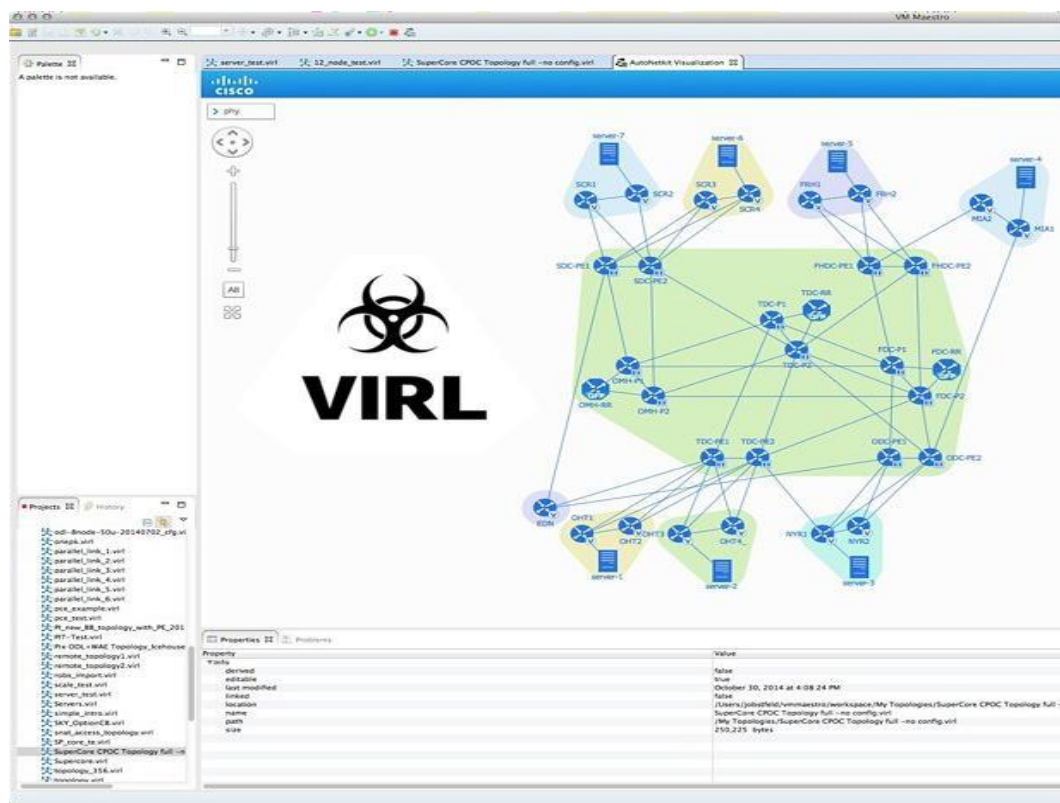
VIRL tiene muchas aplicaciones y usos y puede proporcionar un valor significativo para todos, desde estudiantes, educadores, ingenieros de redes, hasta los equipos de operaciones de redes corporativas más grandes. Algunos casos de uso típico incluyen:

- Aprendiendo tecnologías de red básicas
- Entrenamiento de red, protocolo o solución
- Probar nuevas funciones



- Desarrollar y probar nuevas configuraciones antes de los despliegues en vivo
- Resolución de problemas observados en redes en vivo
- Validación de cambios como parte de una cadena de herramientas DevOps / NetOps
- Entrega de demostraciones de características y aplicaciones.
- Proporcionar un objetivo de red para el desarrollo de aplicaciones SDN.

Una vez diseñado, VIRL brinda la capacidad de configurar automáticamente el direccionamiento IP y los protocolos de enrutamiento para que pueda ponerse en funcionamiento rápidamente. (Packet , 2020, p. 1). En la Figura 19-1 se aprecia el interfaz de VIRL.



**Figura 19-1.** Interfaz de VIRL

Fuente: (Routerjockey, 2019, p. 1)

### 1.4.2. Generadores de tráfico

En la investigación denominada 'Evaluación del rendimiento e interoperabilidad de una plataforma de Routing OpenSource' los autores afirman que:

Un generador de tráfico es un programa muy útil en el área de los enrutadores, nos sirven para simular el tráfico de ciertas aplicaciones o servicios de red sin necesidad de tenerlos instalados o corriendo para probar ciertos parámetros de nuestra red como ancho de banda, memoria consumida por un tipo de servicio en un enrutador, etc. (Lagua & Lugmania, 2017, p. 35).

Entre los principales generadores de tráfico existentes en la actualidad tenemos:

- TREX Traffic Generator
- OSTINATO
- D-ITG
- IRIS
- PACKET PLAYER
- PACKET PLAYERPIPO E2H

#### *1.4.2.1. Ostinato*

Ostinato es un generador de paquetes y un generador de tráfico de red con una GUI amigable. También es una potente API de Python usada en la automatización de pruebas en la red. Este genera y transmite paquetes de varias transmisiones con diferentes protocolos a diferentes velocidades. Piense en ello como "Wireshark en reversa" (Srivats, 2020, p. 1).

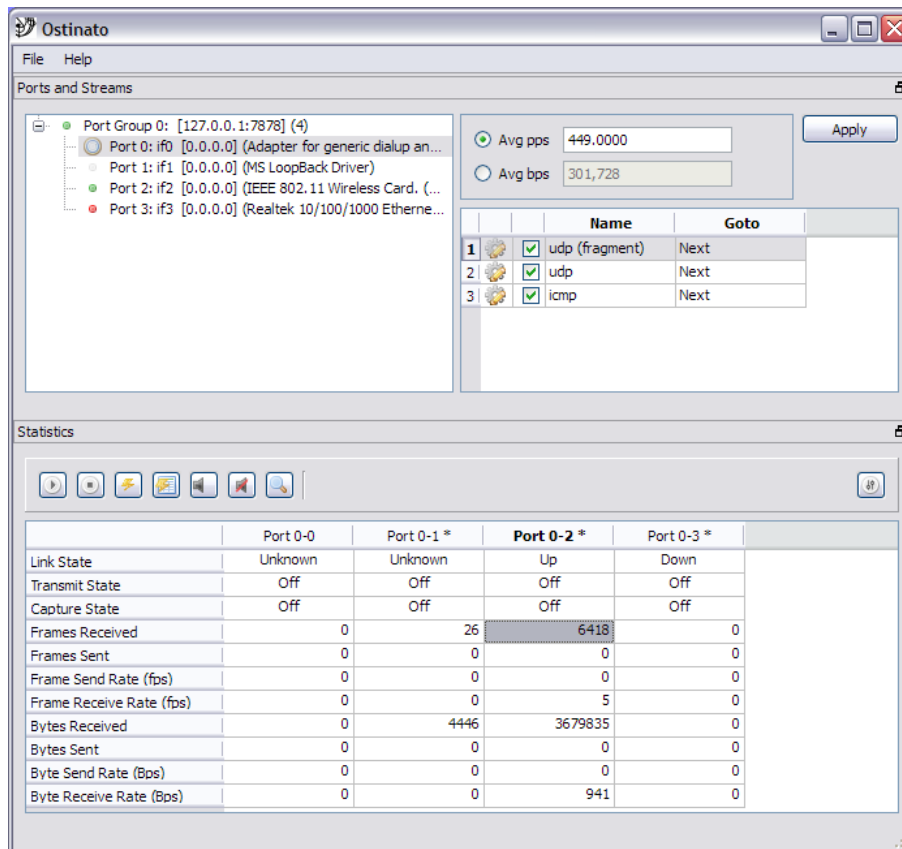
El principal objetivo que brinda Ostinato es ser un generador de tráfico y una herramienta de prueba para equipos de comunicación. Con el uso adecuado los desarrolladores e ingenieros de redes se puede realizar un mejor trabajo, de modo que se optimen los productos de red.

Las características principales de Ostinato las podemos encontrar en su página oficial (Srivats, 2019, p. 1), y son las siguientes:

- Es muy útil para pruebas de red y pruebas funcionales.
- Crear y configurar múltiples transmisiones.

- Configura las tasas de transmisión, ráfagas y número de paquetes.
- Estadísticas, tasas de recepción, transmisión a nivel de interfaz para monitoreo y medición de red en tiempo real.
- Emulación de dispositivo de red (ARP e ICMP) para múltiples hosts IP para ayudar en la simulación de tráfico de red.
- Soporte para los protocolos estándar más comunes.
- Tráfico TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD.
- Cualquier protocolo basado en texto (HTTP, SIP, RTSP, NNTP, etc.).
- Establece un valor para cualquier campo de cualquier protocolo.
- Varíe los campos de paquetes, en los paquetes en tiempo de ejecución, por ejemplo, cambiando las direcciones IP / MAC.
- Apila protocolos en cualquier orden arbitraria para probar casos de error.
- Hex Dump proporcionado por el usuario, puede especificar algunos o todos los bytes en un paquete.
- El control exclusivo de un puerto para evitar que el sistema operativo envíe paquetes perdidos proporciona un entorno de prueba controlado (solo Windows).
- Captura tráfico de red (necesita Wireshark para ver los paquetes capturados).
- Multiplataforma: se puede ejecutar en Windows, Linux, BSD y Mac OS X.
- Es un Software de código abierto.

El interfaz de OSTINATO lo podemos apreciar en la Figura 20-1 presentada a continuación:



**Figura 20-1.** Interfaz de OSTINATO

**Realizado por:** Navarrete Sebastián, Rivera Patricio. 2019

### 1.4.3. Analizadores de Tráfico

Una de las principales tareas que debe realizar un administrador de red es la monitorización de tráfico, esta tarea es la que nos puede ayudar a determinar algún error que esté sucediendo en la comunicación de la red, o también gracias a este análisis podemos ver si existe algún tipo de tráfico no deseado, además de poder comprobar si los paquetes están llegando a los dispositivos finales sin inconveniente (Lagua & Lugmania, 2017, p. 37).

En las redes corporativas este análisis debe incluir la salida a Internet, debido a que este es el enlace que contiene la mayor carga de tráfico, en toda red la complejidad radica en la diversidad de protocolos que se usa en el enrutamiento, por lo que, para interpretar es necesario separar por tipo de protocolo.

El análisis de tráfico de red se puede realizar sea por software o por hardware.

- **Análisis de Software:**

Los analizadores mediante software poseen análisis solo para protocolos extendidos y documentados, comúnmente son de herramienta de licencia libre (Lagua & Lugmania, 2017, p. 38). A partir de la herramienta siempre será necesario la colaboración de un operador para gestionarla, el analizador de tráfico de red más usado es Wireshark, sin embargo, existe algunos analizadores con funciones similares como Windump o TCP Dump.

Para poder usar estos analizadores de tráfico se debe redirigir el tráfico que se pretende analizar por dos métodos:

- a. **Con un Puerto Hub:** Se conecta un concentrador en la red y en uno de los puertos libres se ubica en el analizador. En este caso la cantidad de tráfico que se analiza es limitada.
- b. **Puerto Espejo:** Se conecta un puerto espejo en un Switch de la red, así el tráfico se dirige a un puerto específico, de tal forma que todo el tráfico que pasa por el Switch será copiado y enviado al puerto espejo para analizar el tráfico.

#### *1.4.3.1. Wireshark*

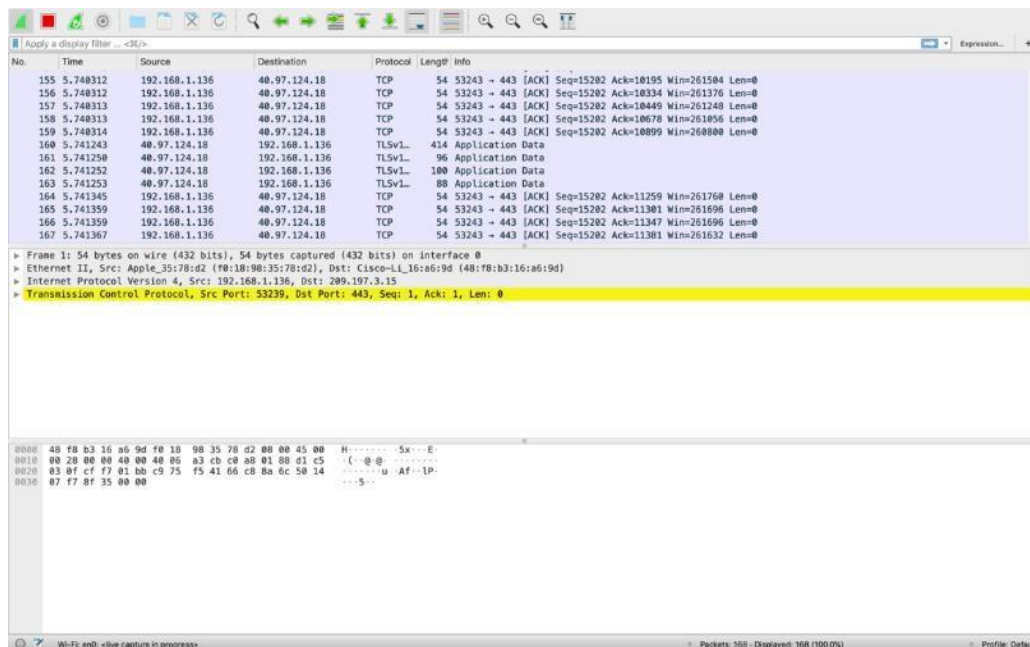
La página oficial de Wireshark indica que:

Wireshark es el analizador de protocolos de red más utilizado y ampliamente utilizado en el mundo. Le permite ver lo que sucede en su red a nivel microscópico y es el estándar de facto (y a menudo de jure) en muchas empresas comerciales y sin fines de lucro, agencias gubernamentales e instituciones educativas. El desarrollo de Wireshark prospera gracias a las contribuciones voluntarias de expertos en redes de todo el mundo y es la continuación de un proyecto iniciado por Gerald Combs en 1998 (WireShark, 2019, p. 1).

Las características de este analizador de tráfico las podemos encontrar en la página oficial de Wireshark (2019) y son:

- Inspección profunda de cientos de protocolos
- Captura en vivo y análisis fuera de línea
- Navegador de paquetes estándar de tres paneles
- Multiplataforma: se ejecuta en Windows, Linux, macOS, Solaris, FreeBSD, NetBSD y muchos otros
- Los datos de red capturados se pueden navegar a través de una GUI, o mediante la utilidad TShark TTY-mode
- Tiene los filtros de visualización más potentes de la industria
- Soporte de descifrado para muchos protocolos, incluidos IPsec, ISAKMP, Kerberos, SNMPv3, SSL / TLS, WEP y WPA / WPA2
- La salida se puede exportar a XML, PostScript®, CSV o texto sin formato

Wireshark es una herramienta muy efectiva en el momento de analizar tráfico en una red, además de poseer un interfaz de línea de comandos, también posee una interfaz gráfica con un entorno muy amigable para con el administrador de la red (Lagua & Lugmania, 2017, p. 41), mismo que podemos observar en la Figura 21-1 presentada a continuación:



**Figura 21-1.** Interfaz de WIRESHARK

Realizado por: Navarrete Sebastián, Rivera Patricio. 2019

## **CAPITULO II**

### **2. MARCO METODOLOGICO**

#### **2.1. Introducción**

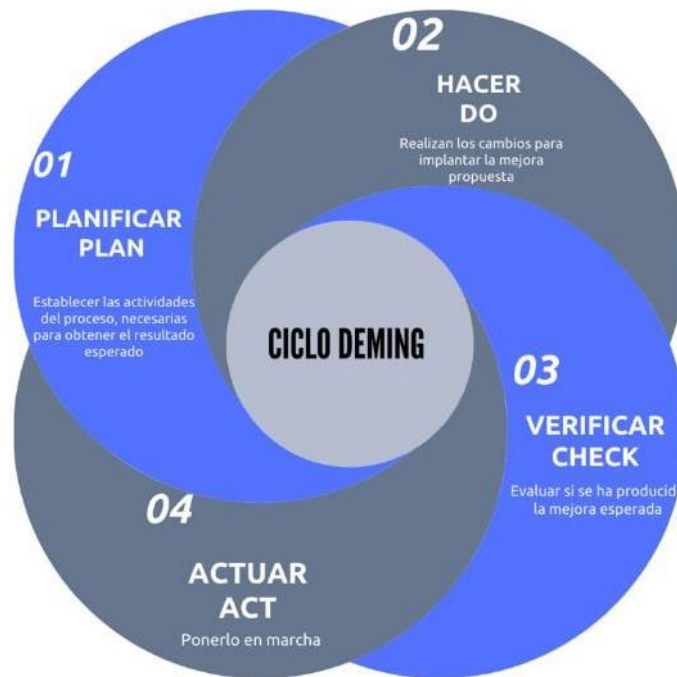
En el presente capítulo se detalla todos los procesos efectuados en la investigación, el análisis y estudio de la ingeniería de tráfico mediante segment routing, dentro de la cual se determinó varios parámetros especialmente en el software con el que se procedió a trabajar, en el que se realizaron pruebas para evaluar el rendimiento de este.

#### **2.2. Parámetros para determinar**

Como se explicó en el Capítulo 1, MPLS es considerado como 'La tecnología del futuro', sin embargo, el alto consumo de tráfico de internet hace que la calidad de servicio sea cada vez más eficiente, viéndose obligada a buscar nuevas tecnologías y técnicas para brindar así una mejor calidad de servicio. La ingeniería de tráfico es la técnica que ayuda a las mejoras constantes de MPLS, es por eso que, en la presente investigación se propone realizar una comparativa de una técnica casi nueva con las más usadas en la actualidad.

Para realizar esta investigación también se ha tomado en consideración el ciclo de Deming, en el cual ciertas empresas como Equipo Altran en su blog manifiestan que “deming es la sistemática más usada para implantar un sistema de mejora continua cuyo principal objetivo es la autoevaluación, destacando los puntos fuertes que hay que tratar de mantener y las áreas de mejora en las que se deberá actuar” (García, 2016, p. 1).

Basándonos en el ciclo Deming conocido también como el ciclo PDCA (Plan, Do, Act, Check) el ciclo se muestra en la Figura 1-2 presentada a continuación:



**Figura 1-2:** Ciclo Deming

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Dentro del ciclo y con los parámetros especificados podemos realizar una investigación eficiente y muy bien estructurada, como se muestra en la Figura 2-2 presentada a continuación:



**Figura 2-2:** Ciclo Deming

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019



## 2.3. Análisis de software

Desde el punto de vista técnico se ha podido observar los IOS de diferentes marcas como Cisco, Huawei, Juniper, HP, Mikrotik, cabe resaltar que son las empresas de comunicación más grandes en la actualidad. Para la presente investigación se evaluó que dispositivos son capaces de soportar Segment Routing. En este aspecto se ha considerado que los IOS capaces de soportar esta tecnología es el IOS XR de Cisco, sin embargo, la investigación quedara abierta a la posibilidad del uso de IOS de Huawei o Juniper que también soporten esta tecnología

Un punto para considerar es que los dispositivos que soporten esa tecnología se puedan simular en un software, con el fin de poder realizar diferentes pruebas de rendimiento.

El análisis de software de simulación que sea capaz de correr esta tecnología es también un parámetro muy importante, dado que existen diferentes plataformas que no pueden simular estas tecnologías. Por razones como esta se debía evaluar un software para simulación que cuente con la suficiente documentación y que sea flexible en su manejo y configuración, además que no tenga problemas en correr estos protocolos.

### 2.3.1. Software de simulación

Para poder realizar las simulaciones de prueba de esta investigación se procedió a evaluar ciertos parámetros o requisitos mínimos para el uso del software, los cuales se presentarán en la Tabla 1-2 presentada a continuación.

**Tabla 1-2:** Requisitos mínimos

| Actual  | Necesario    |
|---|--------------|
| Documentación   | Necesario    |
| Soporte MPLS  | Necesario    |
| Multiplataforma (compatible con IOS de diversas marcas) | Necesario    |
| Licencia Gratuita                                       | No necesario |

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2019

Para la selección del Software se evaluó de manera cuantitativa, y se asignó los valores entre 0 y 1, siendo 1 el valor más alto en caso de cumplir con el requerimiento y de 0 en caso de no cumplir. Este análisis fue realizado con los simuladores más conocidos en la actualidad y finalmente el simulador que obtuvo el puntaje más alto, será el usado para la presente investigación.

En la Tabla 2-2 mostrada a continuación se evidenciará la comparativa.

**Tabla 2-2:** Comparativa de Software

| <b>Software</b>           | <b>Actual</b> | <b>Documentación</b> | <b>Soporte<br/>Mpls</b> | <b>Multiplataforma</b> | <b>Licencia<br/>gratuita</b> | <b>TOTAL</b> |
|---------------------------|---------------|----------------------|-------------------------|------------------------|------------------------------|--------------|
| CISCO<br>PACKET<br>TARCER | 1             | 1                    | 0                       | 1                      | 1                            | 4            |
| GNS3                      | 1             | 1                    | 1                       | 1                      | 1                            | 5            |
| eNSP<br>(Huawei)          | 1             | 1                    | 1                       | 0                      | 1                            | 4            |

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2019

Como se puede evidenciar en cada una de las cualidades se le asignó una calificación, y de este modo obtener el software idóneo para el presente trabajo.

Por lo tanto, GNS3 es el software elegido para la simulación de las pruebas que se realizarán, debido a su fácil manejo, amplia documentación, además de que, es un software capaz de conectarse con aplicaciones para prueba como generadores de tráfico o medidores de tráfico.

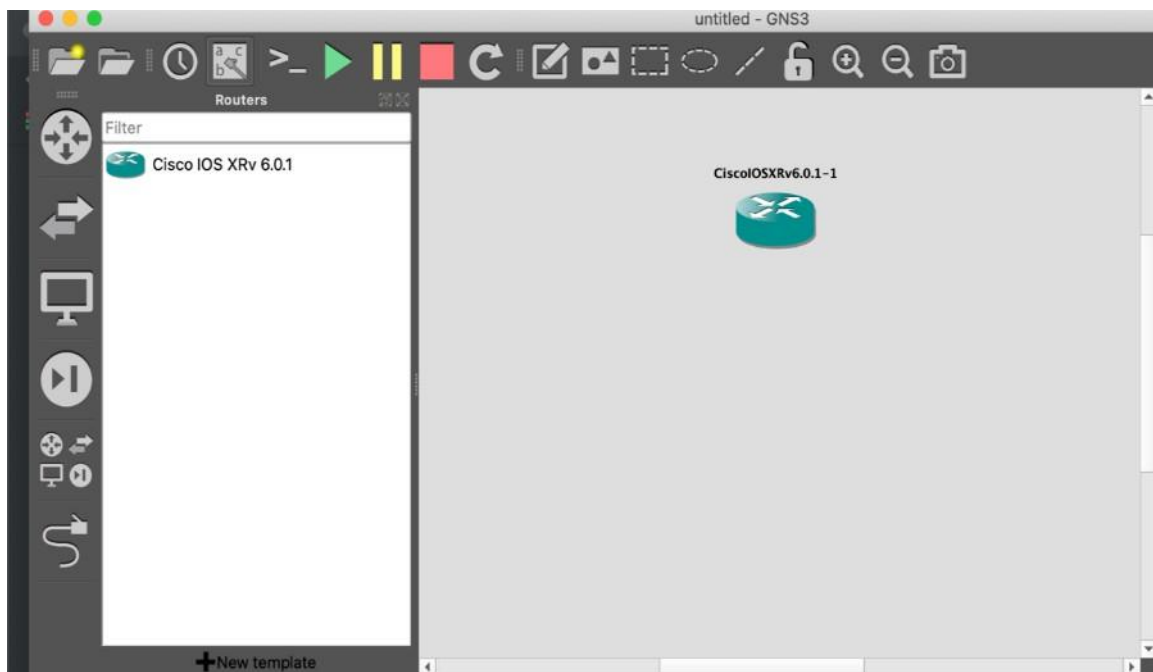
### **2.3.2. Software de Pruebas**

Para realizar las pruebas y poder comparar el rendimiento de la ingeniería de tráfico habitual, con la ingeniería de tráfico con segment routing, se utilizará Ostinato un generador de tráfico capaz de poder simular tráfico real de datos, su principal ventaja es que tiene total compatibilidad con GNS3, mediante sus appliance.

En esta investigación también se usó Wireshark, el software analizador de tráfico más conocido y usado en la actualidad, ya que es capaz de brindarnos la oportunidad de evaluar el tráfico en tiempo real. Con estas herramientas podremos obtener un escenario casi ideal con un entorno físico, permitiéndonos llegar a los resultados finales.

## 2.4. Configuración de Gns3

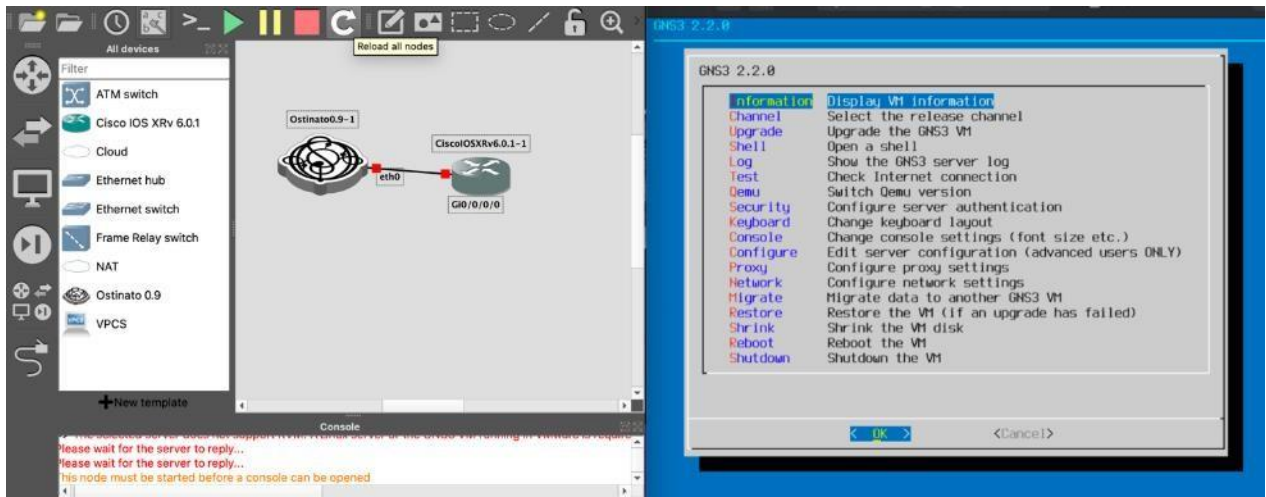
Para el presente estudio se debe configurar en Gns3 con ciertos dispositivos específicos, uno de ellos son los equipos de comunicación que se utilizará para las pruebas de ingeniería de tráfico y segment routing, como se muestra en el Gráfico 1-2 presentado a continuación:



**Gráfico 1-2.** Configuración de Routers en Gns3

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Otro parámetro que se debe configurar es el appliance de Ostinato, para poder realizar las pruebas e inyectar tráfico al escenario, a diferencia de los routers comunes, estos equipos se deben agregar a través de una máquina virtual ya sea en VMWare o Virtual Box. Por otro lado, Gns3 recomienda que el GNS3 VM sea configurado en Vmware como se muestra en el Gráfico 2-2:

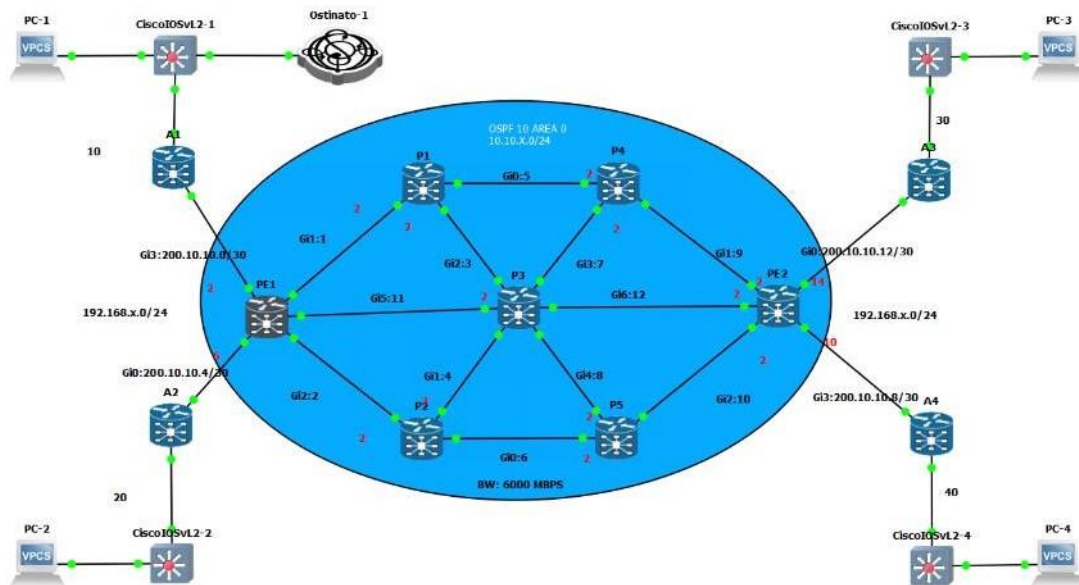


**Gráfico 1-2.** Configuración de Ostinato en Gns3

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

## 2.5. Pruebas de rendimiento

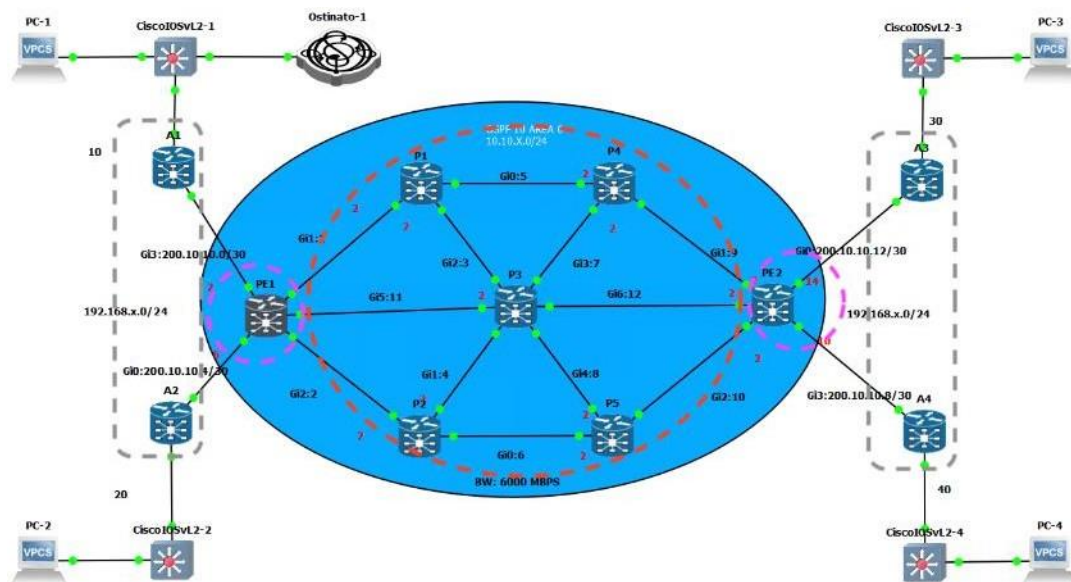
Para las pruebas de rendimiento se usará un escenario de prueba, dentro del cual se configurará MPLS y en él se aplicará Ingeniería de tráfico, así como también Segment Routing, en cada escenario propuesto se inyectará tráfico y se evaluará el comportamiento de cada red acorde al tráfico inyectando. El escenario propuesto se puede apreciar en el Gráfico 3-2 mostrado a continuación:



**Gráfico 2-2.** Escenario de prueba

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Como se mencionó en el Capítulo 1, una red MPLS está formada por diversos componentes, como son los router P, los cuales están dentro del Core. También están los router PE, que son los encargados de comunicar al Core con los router de los clientes, además para la asignación de etiquetas es necesario siempre asignar una interfaz de loopback a los equipos. Para un mejor entendimiento se presenta el Gráfico 4-2 en la que se evidencia la debida clasificación de los routers dentro de nuestro laboratorio de pruebas.



**Gráfico 3-2.** Clasificación de routers en el escenario de prueba

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Según se evidencia en el gráfico podemos apreciar los diferentes componentes de una red MPLS, los cuales se detallan a continuación:

**CE o A:** Es el router encargado de conectar la LAN al router de borde de la red MPLS.

**PE:** También es conocido como router de frontera está ubicado siempre en los extremos del core MPLS.

**P:** Son los routers de tránsito ubicados en el centro del core MPLS.

Nuestro laboratorio está formado por cinco routers P y 2 routers PE, además contiene dos inyectores de tráfico y adicional tiene cuatro routers para clientes de nominados A, una vez entendido los parámetros del escenario se procedió con una asignación de direcciones IPv4, mostrada en la Tabla 3-2 presentada a continuación:

**Tabla 3-2:** Direccionamiento IP

| <b>ROUTER</b> | <b>INTERFAZ</b> | <b>DIRECCION IP4</b> | <b>DIRECCION LOOPBACK</b> | <b>MASCARA</b>  |
|---------------|-----------------|----------------------|---------------------------|-----------------|
| A1            | GI0/0/0/3       | 200.10.10.0          | No es necesario           | 255.255.255.252 |
| A1            | GI0/0/0/1       | 192.168.10.0         | No es necesario           | 255.255.255.0   |
| A2            | GI0/0/0/0       | 200.10.10.4          | No es necesario           | 255.255.255.252 |
| A2            | GI0/0/0/1       | 192.168.20.0         | No es necesario           | 255.255.255.0   |
| A3            | GI0/0/0/0       | 200.10.10.12         | No es necesario           | 255.255.255.252 |
| A3            | GI0/0/0/1       | 192.168.30.0         | No es necesario           | 255.255.255.0   |
| A4            | GI0/0/0/3       | 200.10.10.8          | No es necesario           | 255.255.255.252 |
| A4            | GI0/0/0/1       | 192.168.40.0         | No es necesario           | 255.255.255.0   |
| PE1           | GI0/0/0/0       | 200.10.10.4          | 1.1.1.1/32                | 255.255.255.0   |
|               | GI0/0/0/1       | 10.10.1.0            |                           |                 |
|               | GI0/0/0/2       | 10.10.2.0            |                           |                 |
|               | GI0/0/0/3       | 200.10.10.0          |                           |                 |
|               | GI0/0/0/5       | 10.10.11.0           |                           |                 |
| P1            | GI0/0/0/0       | 10.10.5.0            | 2.2.2.2/32                | 255.255.255.0   |
|               | GI0/0/0/1       | 10.10.1.0            |                           |                 |
|               | GI0/0/0/2       | 10.10.3.0            |                           |                 |
| P2            | GI0/0/0/0       | 10.10.6.0            | 3.3.3.3/24                | 255.255.255.0   |
|               | GI0/0/0/1       | 10.10.4.0            |                           |                 |
|               | GI0/0/0/2       | 10.10.2.0            |                           |                 |
| P3            | GI0/0/0/1       | 10.10.4.0            | 4.4.4.4/24                | 255.255.255.0   |
|               | GI0/0/0/2       | 10.10.3.0            |                           |                 |
|               | GI0/0/0/3       | 10.10.7.0            |                           |                 |
|               | GI0/0/0/4       | 10.10.8.0            |                           |                 |

|     |           |              |            |               |
|-----|-----------|--------------|------------|---------------|
|     | GI0/0/0/5 | 10.10.11.0   |            |               |
|     | GI0/0/0/6 | 10.10.12.0   |            |               |
| P4  | GI0/0/0/0 | 10.10.4.0    | 5.5.5.5/24 | 255.255.255.0 |
|     | GI0/0/0/1 | 10.10.9.0    |            |               |
|     | GI0/0/0/3 | 10.10.7.0    |            |               |
| P5  | GI0/0/0/0 | 10.10.6.0    | 6.6.6.6/24 | 255.255.255.0 |
|     | GI0/0/0/2 | 10.10.10.0   |            |               |
|     | GI0/0/0/4 | 10.10.8.0    |            |               |
| PE2 | GI0/0/0/0 | 200.10.10.12 | 7.7.7.7/24 | 255.255.255.0 |
|     | GI0/0/0/1 | 10.10.9.0    |            |               |
|     | GI0/0/0/2 | 10.10.10.0   |            |               |
|     | GI0/0/0/3 | 200.10.10.8  |            |               |
|     | GI0/0/0/6 | 10.10.12.0   |            |               |

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2019

### ***2.5.1. Implementación de escenario***

Para la comparativa de segment routing con CSPF, en el trabajo de investigación 'Ingeniería de tráfico en redes MPLS de la Universidad de Uruguay' se afirma que:

NET-TE le muestra al usuario cuáles son todos los posibles caminos por los cuales puede rutear su tráfico, asegurándose que cumplan con el BW solicitado por el cliente, además de un conjunto predefinido de restricciones que él mismo puede ingresar. Finalmente, será decisión del usuario el elegir el camino que más le convenga, dentro de toda la gama de soluciones (Delfino, Rivero & San Martín, 2006, p. 20 ).

Así, en un principio usamos al CSPF para encontrar todos aquellos caminos posibles que cumplen con la restricción del ancho de banda. Para todas las demandas, el objetivo es ver cómo se asignan los recursos de la red a cada una de ellas, de manera que se obtenga el máximo aprovechamiento posible de la red (Delfino, Rivero & San Martín, 2006, p. 23).

Configuraciones Básicas: En base a los parámetros establecidos procedemos a configurar los equipos PE, en el escenario de prueba como se evidencia en el Gráfico 5-2 mostrado a continuación:

```

RP/0/0/CPU0:PE2#CONF T
Tue Jun 9 22:17:31.641 UTC
RP/0/0/CPU0:PE2(config)#hostname PE2
RP/0/0/CPU0:PE2(config)#commit
Tue Jun 9 22:17:43.890 UTC
RP/0/0/CPU0:PE2(config)#int lo0
RP/0/0/CPU0:PE2(config-if)#ip add 7.7.7.7 255.255.255.255
RP/0/0/CPU0:PE2(config-if)#exit
RP/0/0/CPU0:PE2(config)#int gi0/0/0/0
RP/0/0/CPU0:PE2(config-if)#ip add 200.10.10.14 255.255.255.252
RP/0/0/CPU0:PE2(config-if)#no shut
RP/0/0/CPU0:PE2(config-if)#exit
RP/0/0/CPU0:PE2(config)#int gi0/0/0/1
RP/0/0/CPU0:PE2(config-if)#ip add 10.10.9.2 255.255.255.0
RP/0/0/CPU0:PE2(config-if)#no shut
RP/0/0/CPU0:PE2(config-if)#exit
RP/0/0/CPU0:PE2(config)#int gi0/0/0/2
RP/0/0/CPU0:PE2(config-if)#ip add 10.10.10.2 255.255.255.252
RP/0/0/CPU0:PE2(config-if)#no shut
RP/0/0/CPU0:PE2(config-if)#exit
RP/0/0/CPU0:PE2(config)#int gi0/0/0/3
RP/0/0/CPU0:PE2(config-if)#ip add 200.10.10.10 255.255.255.252
RP/0/0/CPU0:PE2(config-if)#no shut
RP/0/0/CPU0:PE2(config-if)#exit
RP/0/0/CPU0:PE2(config)#commit
Tue Jun 9 22:17:44.240 UTC
RP/0/0/CPU0:PE2(config)#router ospf 10
RP/0/0/CPU0:PE2(config-ospf)#router-id 7.7.7.7
RP/0/0/CPU0:PE2(config-ospf)#area 0
RP/0/0/CPU0:PE2(config-ospf-ar)#inter lo0
RP/0/0/CPU0:PE2(config-ospf-ar-if)#int gi0/0/0/0
RP/0/0/CPU0:PE2(config-ospf-ar-if)#int gi0/0/0/1
RP/0/0/CPU0:PE2(config-ospf-ar-if)#int gi0/0/0/2
RP/0/0/CPU0:PE2(config-ospf-ar-if)#int gi0/0/0/3
RP/0/0/CPU0:PE2(config-ospf-ar-if)#exit
RP/0/0/CPU0:PE2(config-ospf-ar)#commit

```

**Gráfico 4-2.** Configuración Routers PE

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Además, se debe configurar los equipos P dentro del Core MPLS con el protocolo de enrutamiento, esta configuración se evidencia en el Gráfico 6-2:

```

RP/0/0/CPU0:ios#CONF T
Tue Jun 9 22:12:04.344 UTC
RP/0/0/CPU0:ios(config)#hostname P2
RP/0/0/CPU0:ios(config)#commit
Tue Jun 9 22:12:11.823 UTC
RP/0/0/CPU0:P2(config)#int lo0
RP/0/0/CPU0:P2(config-if)#ip add 3.3.3.3 255.255.255.255
RP/0/0/CPU0:P2(config-if)#exit
RP/0/0/CPU0:P2(config)#int gi0/0/0/0
RP/0/0/CPU0:P2(config-if)#ip add 10.10.6.1 255.255.255.0
RP/0/0/CPU0:P2(config-if)#no shut
RP/0/0/CPU0:P2(config-if)#exit
RP/0/0/CPU0:P2(config)#int gi0/0/0/1
RP/0/0/CPU0:P2(config-if)#ip add 10.10.4.2 255.255.255.0
RP/0/0/CPU0:P2(config-if)#no shut
RP/0/0/CPU0:P2(config-if)#exit
RP/0/0/CPU0:P2(config)#int gi0/0/0/2
RP/0/0/CPU0:P2(config-if)#ip add 10.10.2.2 255.255.255.252
RP/0/0/CPU0:P2(config-if)#no shut
RP/0/0/CPU0:P2(config-if)#exit
RP/0/0/CPU0:P2(config)#commit
Tue Jun 9 22:12:12.143 UTC
RP/0/0/CPU0:Jun 9 22:12:12.263 : ifmgr[228]: %PKT_INFRA-LINK-3-UPDOWN : Interface GigabitEthernet0/0/0/2, changed state to Down
RP/0/0/CPU0:Jun 9 22:12:12.263 : ifmgr[228]: %PKT_INFRA-LINK-3-UPDOWN : Interface GigabitEthernet0/0/0/1, changed state to Down
RP/0/0/CPU0:P2(config)#router ospf 10
RP/0/0/CPU0:P2(config-ospf)#router-id 3.3.3.3
RP/0/0/CPU0:P2(config-ospf)#area 0
RP/0/0/CPU0:P2(config-ospf-ar)#interRP/0/0/CPU0:Jun 9 22:12:12.263 : ifmgr[228]: %PKT_INFRA-LINK-3-UPDOWN : Interface GigabitEthernet0/0/0, changed state to Down
lo0
RP/0/0/CPU0:P2(config-ospf-ar-if)#int gi0/0/0/0
RP/0/0/CPU0:P2(config-ospf-ar-if)#int gi0/0/0/1
RP/0/0/CPU0:P2(config-ospf-ar-if)#int gi0/0/0/2
RP/0/0/CPU0:P2(config-ospf-ar-if)#exit
RP/0/0/CPU0:P2(config-ospf-ar)#commitRP/0/0/CPU0:Jun 9 22:12:12.303 : ifmgr[228]: %PKT_INFRA-LINK-3-UPDOWN : Interface GigabitEthernet0/0/2, changed state to up
RP/0/0/CPU0:Jun 9 22:12:12.303 : ifmgr[228]: %PKT_INFRA-LINK-3-UPDOWN : Interface GigabitEthernet0/0/1, changed state to up

```

**Gráfico 6-2.** Configuración Routers P

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019



Una vez realizada todas las configuraciones procedemos a verificar que exista una correcta conectividad IP entre los equipos, esta verificación se evidencia en el Gráfico 7-2:

```

PC-1> ping 192.168.30.10
192.168.30.10 icmp_seq=1 timeout
192.168.30.10 icmp_seq=2 timeout
84 bytes from 192.168.30.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=5 ttl=58 tim

PC-1> ping 192.168.20.10
84 bytes from 192.168.20.10 icmp_seq=1 ttl=61 tim
84 bytes from 192.168.20.10 icmp_seq=2 ttl=61 tim
84 bytes from 192.168.20.10 icmp_seq=3 ttl=61 tim
84 bytes from 192.168.20.10 icmp_seq=4 ttl=61 tim
84 bytes from 192.168.20.10 icmp_seq=5 ttl=61 tim

PC-1> ping 192.168.30.10
84 bytes from 192.168.30.10 icmp_seq=1 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=2 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=5 ttl=58 tim

PC-1> ping 192.168.40.10
84 bytes from 192.168.40.10 icmp_seq=1 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=2 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=5 ttl=58 tim

PC-1> █

PC-2> ping 192.168.10.10
192.168.10.10 icmp_seq=1 timeout
192.168.10.10 icmp_seq=2 timeout
84 bytes from 192.168.10.10 icmp_seq=3 ttl=61 tim
84 bytes from 192.168.10.10 icmp_seq=4 ttl=61 tim
84 bytes from 192.168.10.10 icmp_seq=5 ttl=61 tim

PC-2>
PC-2> ping 192.168.10.10
84 bytes from 192.168.10.10 icmp_seq=1 ttl=61 tim
84 bytes from 192.168.10.10 icmp_seq=2 ttl=61 tim
84 bytes from 192.168.10.10 icmp_seq=3 ttl=61 tim
84 bytes from 192.168.10.10 icmp_seq=4 ttl=61 tim
84 bytes from 192.168.10.10 icmp_seq=5 ttl=61 tim

PC-2> ping 192.168.30.10
84 bytes from 192.168.30.10 icmp_seq=1 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=2 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.30.10 icmp_seq=5 ttl=58 tim

PC-2> ping 192.168.40.10
192.168.40.10 icmp_seq=1 timeout
192.168.40.10 icmp_seq=2 timeout
84 bytes from 192.168.40.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=5 ttl=58 tim

PC-2> █

PC-3> ip 192.168.30.10 255.255.255.0 192.168.40.10
checking for duplicate address...
PC1 : 192.168.30.10 255.255.255.0 gateway

PC-3> ping 192.168.10.1
84 bytes from 192.168.10.1 icmp_seq=1 ttl=58 tim
84 bytes from 192.168.10.1 icmp_seq=2 ttl=58 tim
84 bytes from 192.168.10.1 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.10.1 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.10.1 icmp_seq=5 ttl=58 tim

PC-3> ping 192.168.20.10
192.168.20.10 icmp_seq=1 timeout
192.168.20.10 icmp_seq=2 timeout
84 bytes from 192.168.20.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.20.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.20.10 icmp_seq=5 ttl=58 tim

PC-3> ping 192.168.40.10
84 bytes from 192.168.40.10 icmp_seq=1 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=2 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.40.10 icmp_seq=5 ttl=58 tim

PC-3> █

PC-4> ip 192.168.40.10 255.255.255.0 192.168.30.10
checking for duplicate address...
PC1 : 192.168.40.10 255.255.255.0 gateway

PC-4>
PC-4> ping 192.168.10.10
84 bytes from 192.168.10.10 icmp_seq=1 ttl=58 tim
84 bytes from 192.168.10.10 icmp_seq=2 ttl=58 tim
84 bytes from 192.168.10.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.10.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.10.10 icmp_seq=5 ttl=58 tim

PC-4> ping 192.168.20.10
84 bytes from 192.168.20.10 icmp_seq=1 ttl=58 tim
84 bytes from 192.168.20.10 icmp_seq=2 ttl=58 tim
84 bytes from 192.168.20.10 icmp_seq=3 ttl=58 tim
84 bytes from 192.168.20.10 icmp_seq=4 ttl=58 tim
84 bytes from 192.168.20.10 icmp_seq=5 ttl=58 tim

PC-4> ping 192.168.30.10
84 bytes from 192.168.30.10 icmp_seq=1 ttl=61 tim
84 bytes from 192.168.30.10 icmp_seq=2 ttl=61 tim
84 bytes from 192.168.30.10 icmp_seq=3 ttl=61 tim
84 bytes from 192.168.30.10 icmp_seq=4 ttl=61 tim
84 bytes from 192.168.30.10 icmp_seq=5 ttl=61 tim

PC-4> █
fotoefectos

```

**Gráfico 7-2.** Prueba conectividad IP entre PCs

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Usando el comando **#show ip route ospf**, se puede apreciar las rutas aprendidas por OSPF dentro de nuestro Core, como se aprecia en el Gráfico 8-2:

```

RP/0/0/CPU0:PE1#SHOW ip route ospf
Tue Jun 9 22:19:55.951 UTC
O 2.2.2.2/32 [110/2] via 10.10.1.2, 00:12:05, GigabitEthernet0/0/0/1
O 3.3.3.3/32 [110/2] via 10.10.2.2, 00:07:21, GigabitEthernet0/0/0/2
O 4.4.4.4/32 [110/3] via 10.10.2.2, 00:06:37, GigabitEthernet0/0/0/2
O 5.5.5.5/32 [110/3] via 10.10.1.2, 00:06:37, GigabitEthernet0/0/0/1
O 6.6.6.6/32 [110/3] via 10.10.1.2, 00:05:00, GigabitEthernet0/0/0/1
O 7.7.7.7/32 [110/4] via 10.10.1.2, 00:05:00, GigabitEthernet0/0/0/1
O 10.10.3.0/30 [110/2] via 10.10.1.2, 00:12:05, GigabitEthernet0/0/0/1
O 10.10.4.0/24 [110/2] via 10.10.2.2, 00:07:21, GigabitEthernet0/0/0/2
O 10.10.5.0/24 [110/2] via 10.10.1.2, 00:12:05, GigabitEthernet0/0/0/1
O 10.10.6.0/24 [110/2] via 10.10.2.2, 00:07:21, GigabitEthernet0/0/0/2
O 10.10.7.0/30 [110/3] via 10.10.2.2, 00:06:37, GigabitEthernet0/0/0/2
O 10.10.8.0/30 [110/3] via 10.10.1.2, 00:06:37, GigabitEthernet0/0/0/1
O 10.10.9.0/24 [110/3] via 10.10.2.2, 00:06:37, GigabitEthernet0/0/0/2
O 10.10.10.0/30 [110/4] via 10.10.1.2, 00:05:00, GigabitEthernet0/0/0/1
O 200.10.10.8/30 [110/4] via 10.10.1.2, 00:05:00, GigabitEthernet0/0/0/1
O 200.10.10.12/30 [110/4] via 10.10.1.2, 00:05:00, GigabitEthernet0/0/0/1
RP/0/0/CPU0:PE1#

```

**Gráfico 8-2.** Rutas aprendidas por OSPF desde router PE1

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Una vez configurado el protocolo de enrutamiento procedemos a la configuración de MPLS, para de esta forma analizar los túneles por defecto de nuestro escenario; la teoría afirma que la configuración MPLS debe estar en los routers PE y P, esta configuración se aprecia en el Gráfico 9-2 que se aprecia a continuación:

```

Mon Jun 22 20:12:02.937 UTC
RP/0/0/CPU0:PE1(config)#mpls ldp
RP/0/0/CPU0:PE1(config-ldp)#router-id 1.1.1.1
RP/0/0/CPU0:PE1(config-ldp)#inter gi0/0/0/0
RP/0/0/CPU0:PE1(config-ldp-if)#no int gi0/0/0/0
RP/0/0/CPU0:PE1(config-ldp)#
RP/0/0/CPU0:PE1(config-ldp)#
RP/0/0/CPU0:PE1(config-ldp)#inter gi0/0/0/2
RP/0/0/CPU0:PE1(config-ldp-if)#exit
RP/0/0/CPU0:PE1(config-ldp)#inter gi0/0/0/1
RP/0/0/CPU0:PE1(config-ldp-if)#exit
RP/0/0/CPU0:PE1(config-ldp)#exit
RP/0/0/CPU0:PE1(config)#commit
Mon Jun 22 20:13:15.622 UTC
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#inter gi0/0/0/2
RP/0/0/CPU0:PE1(config-if)#mtu 9000
RP/0/0/CPU0:PE1(config-if)#exit
RP/0/0/CPU0:PE1(config)#inter gi0/0/0/1
RP/0/0/CPU0:PE1(config-if)#mtu 9000
RP/0/0/CPU0:PE1(config-if)#exit
RP/0/0/CPU0:PE1(config)#commit
Mon Jun 22 20:13:24.931 UTC
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#end

```

**Gráfico 9-2.** Configuración de MPLS en router PE

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 10-2 se evidencia un ejemplo de la configuración de los routers P:

```

RP/0/0/CPU0:P1(config)#mpls label range 20000 21999
RP/0/0/CPU0:P1(config)#mpls ldp
RP/0/0/CPU0:P1(config-ldp)#router-id 2.2.2.2
RP/0/0/CPU0:P1(config-ldp)#inter gi0/0/0/0
RP/0/0/CPU0:P1(config-ldp-if)#exit
RP/0/0/CPU0:P1(config-ldp)#inter gi0/0/0/2
RP/0/0/CPU0:P1(config-ldp-if)#exit
RP/0/0/CPU0:P1(config-ldp)#inter gi0/0/0/1
RP/0/0/CPU0:P1(config-ldp-if)#exit
RP/0/0/CPU0:P1(config-ldp)#exit
RP/0/0/CPU0:P1(config)#commit
Mon Jun 22 16:43:22.521 UTC
RP/0/0/CPU0:P1(config)#
RP/0/0/CPU0:P1(config)#inter gi0/0/0/0
RP/0/0/CPU0:P1(config-if)#mtu 9000
RP/0/0/CPU0:P1(config-if)#exit
RP/0/0/CPU0:P1(config)#inter gi0/0/0/2
RP/0/0/CPU0:P1(config-if)#mtu 9000
RP/0/0/CPU0:P1(config-if)#exit
RP/0/0/CPU0:P1(config)#inter gi0/0/0/1
RP/0/0/CPU0:P1(config-if)#mtu 9000
RP/0/0/CPU0:P1(config-if)#exit
RP/0/0/CPU0:P1(config)#commit

```

**Gráfico 10-2.** Configuración MPLS en router P

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Una vez configurado el escenario con MPLS, mediante el comando **#show mpls forwarding** se evidencia la tabla creada por el protocolo MPLS con las debidas etiquetas asignadas por cada router, como se evidencia en el Gráfico 11-2:

```

PE1 x
User Access Verification
Username:
Username: PE1
Password:

RP/0/0/CPU0:PE1#show mpls forwarding
Mon Jul 6 05:03:18.236 UTC
Local  Outgoing  Prefix      Outgoing    Next Hop    Bytes
Label  Label      or ID       Interface   Interface   Switched
-----
16000  Pop         2.2.2.2/32  Gi0/0/0/1  10.10.1.2   1554710
16001  20000      5.5.5.5/32  Gi0/0/0/1  10.10.1.2   0
16002  20001      7.7.7.7/32  Gi0/0/0/1  10.10.1.2   0
16003  Pop         10.10.5.0/24 Gi0/0/0/1  10.10.1.2   0
16004  Pop         10.10.3.0/24 Gi0/0/0/1  10.10.1.2   0
16005  20002      10.10.9.0/24 Gi0/0/0/1  10.10.1.2   0
16006  20011      200.10.10.12/30 Gi0/0/0/1  10.10.1.2   0
16007  20003      200.10.10.8/30 Gi0/0/0/1  10.10.1.2   0
16008  20004      10.10.10.0/24 Gi0/0/0/1  10.10.1.2   0
16009  20005      10.10.12.0/24 Gi0/0/0/1  10.10.1.2   0
16010  Unlabelled 192.168.20.0/24 Gi0/0/0/0  200.10.10.5  0
16011  20020      192.168.30.0/24 Gi0/0/0/1  10.10.1.2   0
16012  20006      192.168.40.0/24 Gi0/0/0/1  10.10.1.2   0
16013  Unlabelled 192.168.10.0/24 Gi0/0/0/3  200.10.10.1  0
16014  Unlabelled 3.3.3.3/32   Gi0/0/0/2  10.10.2.2   0
16015  20015      4.4.4.4/32   Gi0/0/0/1  10.10.1.2   0
        Unlabelled 4.4.4.4/32   Gi0/0/0/2  10.10.2.2   1601810
16016  Unlabelled 10.10.4.0/24 Gi0/0/0/2  10.10.2.2   0
16017  20016      10.10.7.0/24 Gi0/0/0/1  10.10.1.2   0
        Unlabelled 10.10.7.0/24 Gi0/0/0/2  10.10.2.2   0
--More--
Ready                               Telnet: 192.168.227.128 33

```

**Gráfico 11-2.** Tabla de forwarding equipos PE

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

También podemos ver la tabla de forwarding en los equipos P, como se muestra en el Gráfico 12-2:

```

RP/0/0/CPU0:P3#show mpls forwarding
Mon Jul 6 04:11:53.110 UTC
Local  Outgoing  Prefix      Outgoing    Next Hop    Bytes
Label  Label      or ID       Interface   Interface   Switched
-----
24000  Unlabelled 3.3.3.3/32   Gi0/0/0/1  10.10.4.2   0
24001  Unlabelled 10.10.2.0/24 Gi0/0/0/1  10.10.4.2   0
24002  Unlabelled 10.10.6.0/24 Gi0/0/0/1  10.10.4.2   0
24003  Pop         5.5.5.5/32   Gi0/0/0/3  10.10.7.2   1873788
24004  26005      7.7.7.7/32   Gi0/0/0/3  10.10.7.2   2027654
24005  Pop         10.10.9.0/24 Gi0/0/0/3  10.10.7.2   0
24006  26006      200.10.10.8/30 Gi0/0/0/3  10.10.7.2   0
24007  26007      10.10.10.0/24 Gi0/0/0/3  10.10.7.2   0
24008  26009      192.168.40.0/24 Gi0/0/0/3  10.10.7.2   0
24009  Unlabelled 1.1.1.1/32   Gi0/0/0/1  10.10.4.2   0
        20007      1.1.1.1/32   Gi0/0/0/2  10.10.3.2   1724614
24010  Pop         2.2.2.2/32   Gi0/0/0/2  10.10.3.2   1873674
24011  Pop         10.10.1.0/24 Gi0/0/0/2  10.10.3.2   0
24012  Pop         10.10.5.0/24 Gi0/0/0/2  10.10.3.2   0
        Pop         10.10.5.0/24 Gi0/0/0/3  10.10.7.2   0
24013  26013      200.10.10.12/30 Gi0/0/0/3  10.10.7.2   0
24014  Unlabelled 200.10.10.0/30 Gi0/0/0/1  10.10.4.2   0
        20009      200.10.10.0/30 Gi0/0/0/2  10.10.3.2   0
24015  Unlabelled 200.10.10.4/30 Gi0/0/0/1  10.10.4.2   0
        20008      200.10.10.4/30 Gi0/0/0/2  10.10.3.2   0
24016  Unlabelled 192.168.10.0/24 Gi0/0/0/1  10.10.4.2   0
        20012      192.168.10.0/24 Gi0/0/0/2  10.10.3.2   0
24017  Unlabelled 192.168.20.0/24 Gi0/0/0/1  10.10.4.2   0
        20013      192.168.20.0/24 Gi0/0/0/2  10.10.3.2   0
24018  26020      192.168.30.0/24 Gi0/0/0/3  10.10.7.2   0
RP/0/0/CPU0:P3#

```

**Gráfico 12-2.** Tabla de forwarding equipos P1

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019



## 2.5.2. Evidencia de túneles MPLS por defecto

Antes de empezar con la configuración de los túneles de ingeniería de tráfico o segment routing, para las respectivas comparaciones, es necesario primero evidenciar las rutas que tiene por defecto cada host al comunicarse con los demás equipos de la red. Estas pruebas las realizamos con el comando **#traceroute** desde los equipos de comunicación A, y se observa en los gráficos presentadas a continuación:

En el Gráfico 13-2 se muestra el camino que toman los paquetes al ser enviados desde A1 hasta A3:

```
RP/0/0/CPU0:A1#trace 192.168.30.10
Mon Jun 22 17:32:06.790 UTC

Type escape sequence to abort.
Tracing the route to 192.168.30.10

 0 200.10.10.2 0 msec 0 msec 0 msec
 1 200.10.10.2 0 msec 0 msec 0 msec
 2 10.10.1.2 [MPLS: Label 20018 Exp 0] 9 msec 9 msec 9 msec
 3 10.10.5.2 [MPLS: Label 26018 Exp 0] 9 msec 9 msec 0 msec
 4 10.10.9.2 [MPLS: Label 18001 Exp 0] 9 msec 9 msec 9 msec
 5 200.10.10.13 9 msec 9 msec 9 msec
 6 192.168.30.10 9 msec 9 msec 9 msec
RP/0/0/CPU0:A1#
```

**Gráfico 13-2.** Tráfico enviado desde A1 hasta A3

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 14-2 se aprecia el camino que toman los paquetes al ser enviados desde A1 hasta A4:

```
RP/0/0/CPU0:A1#trace 192.168.40.10
Mon Jun 22 17:32:24.139 UTC

Type escape sequence to abort.
Tracing the route to 192.168.40.10

 0 200.10.10.2 0 msec 0 msec 0 msec
 1 200.10.10.2 0 msec 0 msec 0 msec
 2 10.10.1.2 [MPLS: Label 20019 Exp 0] 9 msec 9 msec 9 msec
 3 10.10.5.2 [MPLS: Label 26019 Exp 0] 9 msec 9 msec 0 msec
 4 10.10.9.2 [MPLS: Label 18000 Exp 0] 0 msec 9 msec 9 msec
 5 200.10.10.9 9 msec 0 msec 9 msec
 6 192.168.40.10 3009 msec 9 msec 9 msec
RP/0/0/CPU0:A1#
RP/0/0/CPU0:A1#
```

**Gráfico 14-2.** Tráfico enviado desde A1 hasta A4

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 15-2 se muestra el camino que toman los paquetes al ser enviados desde A2 hasta A3:

```
RP/0/0/CPU0:A2#trace 192.168.30.10
Mon Jun 22 21:03:31.371 UTC
```

```
Type escape sequence to abort.
Tracing the route to 192.168.30.10
```

```
 1 200.10.10.6 9 msec 0 msec 0 msec
 2 10.10.1.2 [MPLS: Label 20018 Exp 0] 9 msec 9 msec 9 msec
 3 10.10.5.2 [MPLS: Label 26018 Exp 0] 9 msec 9 msec 9 msec
 4 10.10.9.2 [MPLS: Label 18001 Exp 0] 9 msec 9 msec 0 msec
 5 200.10.10.13 9 msec 9 msec 9 msec
 6 192.168.30.10 9 msec 9 msec 9 msec
```

### Gráfico 15-2. Tráfico enviado desde A2 hasta A3

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 16-2 se observa el camino que toman los paquetes al ser enviados desde A2 hasta A4:

```
RP/0/0/CPU0:A2#trace 192.168.40.10
Mon Jun 22 21:03:36.980 UTC

Type escape sequence to abort.
Tracing the route to 192.168.40.10

 1 200.10.10.6 9 msec 0 msec 0 msec
 2 10.10.1.2 [MPLS: Label 20019 Exp 0] 9 msec 9 msec 9 msec
 3 10.10.5.2 [MPLS: Label 26019 Exp 0] 9 msec 9 msec 9 msec
 4 10.10.9.2 [MPLS: Label 18000 Exp 0] 9 msec 9 msec 9 msec
 5 200.10.10.9 9 msec 9 msec 9 msec
 6 192.168.40.10 9 msec 9 msec 9 msec
RP/0/0/CPU0:A2#
```

### Gráfico 16-2. Tráfico enviado desde A2 hasta A4

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 17-2 se muestra el camino que toman los paquetes al ser enviados desde A3 hasta A1:

```
RP/0/0/CPU0:A3#trace 192.168.10.10
Mon Jun 22 17:36:45.436 UTC

Type escape sequence to abort.
Tracing the route to 192.168.10.10

 1 200.10.10.14 0 msec 0 msec 0 msec
 2 10.10.9.1 [MPLS: Label 26012 Exp 0] 9 msec 0 msec 9 msec
 3 10.10.5.1 [MPLS: Label 20004 Exp 0] 0 msec 9 msec 0 msec
 4 10.10.1.1 [MPLS: Label 24000 Exp 0] 9 msec 9 msec 9 msec
 5 200.10.10.1 9 msec 0 msec 0 msec
 6 192.168.10.10 3009 msec 9 msec 9 msec
```

### Gráfico 17-2. Tráfico enviado desde A3 hasta A1

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 18-2 se evidencia el camino que toman los paquetes al ser enviados desde A3 hasta A2:

```
RP/0/0/CPU0:A3#trace 192.168.20.10
Mon Jun 22 17:37:02.025 UTC
```

```
Type escape sequence to abort.
Tracing the route to 192.168.20.10
```

```
 1 200.10.10.14 0 msec 0 msec 0 msec
 2 10.10.9.1 [MPLS: Label 26013 Exp 0] 9 msec 9 msec 0 msec
 3 10.10.5.1 [MPLS: Label 20005 Exp 0] 9 msec 0 msec 9 msec
 4 10.10.1.1 [MPLS: Label 24018 Exp 0] 9 msec 9 msec 9 msec
 5 200.10.10.5 9 msec 0 msec 9 msec
 6 192.168.20.10 3009 msec 9 msec 9 msec
```

### Gráfico 18-2. Tráfico enviado desde A3 hasta A2

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 19-2 se muestra el camino que toman los paquetes al ser enviados desde A4 hasta A1:

```
RP/0/0/CPU0:A4#trace 192.168.10.10
Mon Jun 22 17:30:54.695 UTC
```

```
Type escape sequence to abort.
Tracing the route to 192.168.10.10
```

```
 1 200.10.10.10 0 msec 0 msec 0 msec
 2 10.10.9.1 [MPLS: Label 26012 Exp 0] 9 msec 9 msec 9 msec
 3 10.10.5.1 [MPLS: Label 20004 Exp 0] 19 msec 9 msec 0 msec
 4 10.10.1.1 [MPLS: Label 24000 Exp 0] 9 msec 9 msec 0 msec
 5 200.10.10.1 9 msec 9 msec 9 msec
 6 192.168.10.10 9 msec 9 msec 9 msec
```

```
RP/0/0/CPU0:A4#
```

### Gráfico 19-2. Tráfico enviado desde A4 hasta A1

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

En el Gráfico 20-2 se observa el camino que toman los paquetes al ser enviados desde A4 hasta A2:

```
RP/0/0/CPU0:A4#trace 192.168.20.10
Mon Jun 22 17:30:59.655 UTC
```

```
Type escape sequence to abort.
Tracing the route to 192.168.20.10
```

```
 1 200.10.10.10 0 msec 0 msec 0 msec
 2 10.10.9.1 [MPLS: Label 26013 Exp 0] 9 msec 9 msec 9 msec
 3 10.10.5.1 [MPLS: Label 20005 Exp 0] 0 msec 9 msec 9 msec
 4 10.10.1.1 [MPLS: Label 24018 Exp 0] 9 msec 9 msec 9 msec
 5 200.10.10.5 9 msec 9 msec 9 msec
 6 192.168.20.10 9 msec 9 msec 9 msec
```

```
RP/0/0/CPU0:A4#
```

### Gráfico 20-2. Tráfico enviado desde A4 hasta A2

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

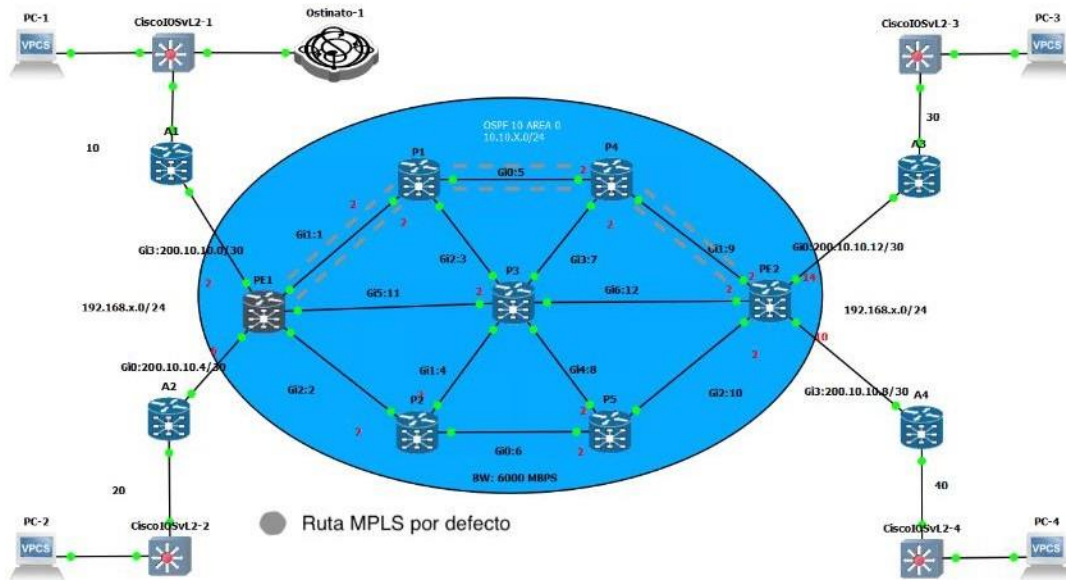
En la Tabla 4-2 se puede apreciar un resumen de las rutas que toma cada dispositivo A, para comunicarse en la red de Core MPLS:

**Tabla 4-2:** Ruta del tráfico MPLS

| ORIGEN | DESTINO | RECORRIDO     |
|--------|---------|---------------|
| A1     | A3      | PE1-P1-P4-PE2 |
| A1     | A4      | PE1-P1-P4-PE2 |
| A2     | A3      | PE1-P1-P4-PE2 |
| A2     | A4      | PE1-P1-P4-PE2 |
| A3     | A1      | PE2-P4-P1-PE1 |
| A3     | A2      | PE2-P4-P1-PE1 |
| A4     | A1      | PE2-P4-P1-PE1 |
| A4     | A2      | PE2-P4-P1-PE1 |

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2019

En resumen, podemos observar que todo el tráfico cursado por la red atraviesa un solo camino, el cual para un mejor entendimiento lo presentamos en el Gráfico 21-2 a continuación:



**Gráfico 21-2.** Ruta para el paso de tráfico

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2019

Esta configuración como bien se ha mencionado en el Capítulo I, ocasiona diversos inconvenientes principalmente citamos dos, así tenemos:

- Cuellos de botella en el paso de información en horas pico.
- Desperdicio de recursos en el Core

### 2.5.3. Designación de túneles para ingeniería de tráfico

“La Ingeniería de Tráfico ofrece varios mecanismos para optimizar el rendimiento, modelado, medición, caracterización y el control de tráfico en una red para obtener objetivos específicos de rendimiento y ofrecer servicios competitivos a los clientes” (Hernández, T. 2015, p. 1).

“El enrutamiento por segmentos proporciona un control completo sobre las rutas de reenvío mediante la combinación de instrucciones de red simples. No requiere ningún protocolo adicional. De hecho, en algunos casos elimina los protocolos innecesarios que simplifican su red” (Cisco Systems, 2020, p. 1).

Para la evaluación de desempeño de Segment Routing frente a la ingeniería de tráfico convencional, se hará uso de los túneles, los cuales se listan a continuación:

**Tabla 5-2:** Asignación de túneles para comparativa

| ORIGEN | DESTINO | RECORRIDO     | TÚNEL |
|--------|---------|---------------|-------|
| A1     | A3      | PE1-P3-PE2    | T1    |
| A1     | A4      | PE1-P2-P5-PE2 | T2    |
| A2     | A3      | PE1-P1-P4-PE2 | T3    |
| A2     | A4      | PE1-P3-PE2    | T4    |

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2020

Como se evidencia en la Tabla 5-2, se hará uso de cuatro túneles los cuales se encargarán de comunicar las redes LAN de nuestro escenario. Estos túneles han sido designados a fin de



equilibrar el paso de la información y de esta forma poder evaluar las principales ventajas que Segment Routing nos presenta ante la ingeniería de tráfico con LDP.

Se evalúo el desempeño de retardos, latencias, jitter, perdidas de paquetes, así como también el tiempo de envío de la información, tanto para CSPF como para Segment Routing; los resultados se evidenciarán en el Capítulo III.

## CAPITULO III

### 3. MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En el presente capítulo se describe los resultados obtenidos después de haber realizado la comparativa entre el tráfico que cruza por la red usando Segment Routing o Ingeniería de Tráfico, además se discutirá las principales ventajas que SR posee y porque los grandes fabricantes y proveedores de internet están empezando a usar dicha tecnología.

#### 3.1. Implementación de túneles LDP con Ingeniería de Tráfico

##### 3.1.1. Construcción de Túneles

Como se evidenció en el Capítulo anterior, se procedió a la configuración de los túneles especificados en los equipos Cisco XR, esta configuración se puede apreciar en el Gráfico 1-3 presentado a continuación:

```
RP/0/0/CPU0:PE1(config)#explicit-path name A2_A4
RP/0/0/CPU0:PE1(config-expl-path)#index 1 next-address strict ipv4 unicast 4.4$
RP/0/0/CPU0:PE1(config-expl-path)#index 2 next-address strict ipv4 unicast 7.7$
RP/0/0/CPU0:PE1(config-expl-path)#exit
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#interface tunnel-te4
RP/0/0/CPU0:PE1(config-if)# bandwidth 6000
RP/0/0/CPU0:PE1(config-if)# ipv4 unnumbered Loopback0
RP/0/0/CPU0:PE1(config-if)# destination 7.7.7.7
RP/0/0/CPU0:PE1(config-if)# path-option 1 explicit name A2_A4
RP/0/0/CPU0:PE1(config-if)#exit
RP/0/0/CPU0:PE1(config)#ip access-list A2_A4
RP/0/0/CPU0:PE1(config-ipv4-acl)#permit ip 192.168.20.0 0.0.0.255 192.168.40.0$
RP/0/0/CPU0:PE1(config-ipv4-acl)#exit
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#route-policy 40
RP/0/0/CPU0:PE1(config-rpl)# set attribute-set tunel-te4 4
RP/0/0/CPU0:PE1(config-rpl)#end-policy
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#
RP/0/0/CPU0:PE1(config)#comm
```

**Gráfico 1-3.** Configuración Túneles para MPLS TE

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

### 3.1.2. Túneles para MPLS-TE

Dentro de esta configuración se puede verificar que los túneles se hayan configurado con éxito mediante el comando **#show mpls interface**, como se evidencia en el Gráfico 2-3 que se presenta a continuación:

```
RP/0/0/CPU0:P1#sh mpls interface
Thu Jul 16 17:59:29.165 UTC
Interface                LDP      Tunnel   Static   Enabled
-----
GigabitEthernet0/0/0/0  Yes      Yes      No       Yes
GigabitEthernet0/0/0/1  Yes      Yes      No       Yes
GigabitEthernet0/0/0/2  Yes      No       No       Yes
RP/0/0/CPU0:P1#
RP/0/0/CPU0:P1#
```

**Gráfico 2-3.** Túneles para MPLS TE en equipos P

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

En la gráfica se puede evidenciar los túneles creados en el router P1, como se aprecia los túneles solo están levantados en las interfaces que serán utilizadas para los túneles de ingeniería de tráfico.

Sin embargo, con el comando **#sh int brief** también podemos apreciar todos los túneles creados en nuestro router de borde para cada uno de los túneles propuestos en el escenario, esta configuración se muestra en el Gráfico 3-3:

```
RP/0/0/CPU0:PE1#sh int brief
Thu Jul 16 18:13:29.217 UTC

```

| Intf Name    | Intf State | LineP State | Encap Type | MTU (byte) | Bw (Kbps) |
|--------------|------------|-------------|------------|------------|-----------|
| Lo0          | up         | up          | Loopback   | 1500       | 0         |
| Null0        | up         | up          | Null       | 1500       | 0         |
| tt1          | up         | up          | TUNNEL     | 1500       | 6000      |
| tt3          | up         | up          | TUNNEL     | 1500       | 6000      |
| tt4          | up         | up          | TUNNEL     | 1500       | 6000      |
| Mg0/0/CPU0/0 | admin-down | admin-down  | ARPA       | 1514       | 1000000   |
| Gi0/0/0/0    | up         | up          | ARPA       | 1514       | 1000000   |
| Gi0/0/0/1    | up         | up          | ARPA       | 9000       | 1000000   |
| Gi0/0/0/2    | up         | up          | ARPA       | 9000       | 1000000   |
| Gi0/0/0/3    | up         | up          | ARPA       | 1514       | 1000000   |
| Gi0/0/0/4    | admin-down | admin-down  | ARPA       | 1514       | 1000000   |
| Gi0/0/0/5    | up         | up          | ARPA       | 9000       | 1000000   |
| Gi0/0/0/6    | admin-down | admin-down  | ARPA       | 1514       | 1000000   |
| Gi0/0/0/7    | admin-down | admin-down  | ARPA       | 1514       | 1000000   |

**Gráfico 3-3.** Túneles para MPLS TE en equipos PE

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Se puede observar el ancho de banda configurado en los túneles creados para la técnica de ingeniería de tráfico mediante el comando **#sh ip rsvp interface**, esta comprobación se observa en el Gráfico 4-3:

```

% Invalid input detected at ... marker
RP/0/0/CPU0:PE1#Sh ip rsvp interface
Thu Jul 16 22:04:29.578 UTC

*: RDM: Default I/F B/w % : 75% [default] (max resv/bc0), 0% [default] (bc1)
Interface                MaxBW (bps)  MaxFlow (bps)  Allocated (bps)  MaxSub (bps)
-----
GigabitEthernet0/0/0/1    6M           6M             0 ( 0%)          0
GigabitEthernet0/0/0/2    6M           6M             0 ( 0%)          0
GigabitEthernet0/0/0/5    6M           6M             0 ( 0%)          0
RP/0/0/CPU0:PE1#
RP/0/0/CPU0:PE1#

```

**Gráfico 4-3.** Túneles para MPLS TE en equipos PE

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Como se aprecia en la gráfica, el ancho de banda para las pruebas es de 6 Mbps y este ancho de banda está configurado solo en las interfaces configuradas para MPLS-TE.

### 3.1.3. Tráfico túneles MPLS-TE

Podemos evidenciar el correcto funcionamiento de los túneles generados, para ello utilizamos el comando **#traceroute**. A continuación, se pone en evidencia el envío de la información por cada uno de los túneles de MPLS TE

#### 3.1.3.1. Túnel 1

Acorde a lo expuesto en el capítulo anterior, se usará la técnica de MPLS-TE, para que el Túnel 1 se pueda comunicar con los equipos A1 y A3, pasando por los enrutadores PE1, P3 y PE2. Esta comprobación la podemos realizar mediante el comando **#trace** y su ejemplo se aprecia en el Gráfico 5-3 mostrado a continuación:

```

RP/0/0/CPU0:A1#TRACERoute 192.168.30.10
Thu Jul 16 18:24:11.893 UTC

Type escape sequence to abort.
Tracing the route to 192.168.30.10

 1 200.10.10.2 0 msec  0 msec  0 msec
 2 10.10.11.2 [MPLS: Label 24019 Exp 0] 0 msec  0 msec  0 msec
 3 10.10.12.2 [MPLS: Label 18019 Exp 0] 0 msec  0 msec  0 msec
 4 200.10.10.13 0 msec  0 msec  0 msec
 5 192.168.30.10 3009 msec  9 msec  9 msec

```

**Gráfico 5-3.** Verificación Túnel 1

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

*3.1.3.2. Túnel 2*

El túnel 2 mediante MPLS-TE sería el encargado de comunicar los equipos de frontera A1 y A4, este túnel se comunica por el core MPLS a través de los equipos PE1-P2-P5-PE5, dicho resultado lo apreciamos en el Gráfico 6-3:

```

RP/0/0/CPU0:A1#trace 192.168.40.10
Thu Jul 16 22:56:24.484 UTC

Type escape sequence to abort.
Tracing the route to 192.168.40.10

 1 200.10.10.2 9 msec  0 msec  0 msec
 2 10.10.11.2 [MPLS: Label 24018 Exp 0] 0 msec  0 msec  0 msec
 3 10.10.12.2 [MPLS: Label 18018 Exp 0] 0 msec  0 msec  0 msec
 4 200.10.10.9 0 msec  0 msec  0 msec
 5 192.168.40.10 9 msec  0 msec  0 msec
RP/0/0/CPU0:A1#
RP/0/0/CPU0:A1#

```

**Gráfico 6-3.** Verificación Túnel 2

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

*3.1.3.3. Túnel 3*

El túnel 3 comunica los equipos A2 con A3 y se comunican a través del core por los routers PE1-P1-P4-PE5. El resultado de esta configuración la evidenciamos en el Gráfico 7-3 detallado a continuación:

```

RP/0/0/CPU0:A2#TRACE 192.168.30.10
Thu Jul 16 22:20:25.302 UTC

Type escape sequence to abort.
Tracing the route to 192.168.30.10

 1 200.10.10.6 0 msec  0 msec  0 msec
 2 10.10.1.2 [MPLS: Label 20021 Exp 0] 9 msec  9 msec  0 msec
 3 10.10.5.2 [MPLS: Label 26021 Exp 0] 9 msec  0 msec  9 msec
 4 10.10.9.2 [MPLS: Label 18019 Exp 0] 9 msec  19 msec 19 msec
 5 200.10.10.13 9 msec  9 msec  9 msec
 6 *
    192.168.30.10 9 msec  9 msec
RP/0/0/CPU0:A2#

```

### Gráfico 7-3. Verificación Túnel 3

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

#### 3.1.3.4. Túnel 4

El túnel numero 4 es el encargado de comunicar el equipo A4 con A2. Este túnel está formado por los dispositivos de comunicación PE1-P3-PE2. La configuración la podemos apreciar en el Gráfico 8-3 mostrado a continuación:

```

    192.168.30.10 9 msec  9 msec
RP/0/0/CPU0:A2#TRACE 192.168.40.10
Thu Jul 16 22:21:25.608 UTC

Type escape sequence to abort.
Tracing the route to 192.168.40.10

 1 200.10.10.6 0 msec  0 msec  0 msec
 2 10.10.11.2 [MPLS: Label 24018 Exp 0] 9 msec  0 msec  0 msec
 3 10.10.12.2 [MPLS: Label 18018 Exp 0] 0 msec  0 msec  0 msec
 4 200.10.10.9 9 msec  0 msec  0 msec
 5 192.168.40.10 3009 msec  9 msec  9 msec
RP/0/0/CPU0:A2#

```

### Gráfico 8-3. Verificación Túnel 3

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Estos resultados son importantes demostrarlos, dado que cada uno de los túneles han sido diseñados para la obtención del ultimo resultado del presente trabajo de investigación el cual es evaluar MPLS-TE frente a Segment Routing.

## 3.2. Configuración Segment Routing

A pesar de ser una tecnología casi reciente, la documentación acerca de las configuraciones que existe en la actualidad es muy limitada, además se debe considerar que los grandes proveedores mantienen Segment Routing en constantes actualizaciones para mejorar el protocolo. Sin

embargo, una de las ventajas más grandes de segment routing es su facilidad de implementación como se aprecia en el Gráfico 9-3:

```

RP/0/0/CPU0:PE1(config)#router ospf 10
RP/0/0/CPU0:PE1(config-ospf)#segment-routing mpls
RP/0/0/CPU0:PE1(config-ospf)#router-id 1.1.1.1
RP/0/0/CPU0:PE1(config-ospf)#area 0
RP/0/0/CPU0:PE1(config-ospf-ar)#mpls traffic-eng
RP/0/0/CPU0:PE1(config-ospf-ar)#mpls traffic-eng router-id Loopback0
RP/0/0/CPU0:PE1(config-ospf)#segment-routing mpls
RP/0/0/CPU0:PE1(config-ospf)#inter lo0
RP/0/0/CPU0:PE1(config-if)#int gi0/0/0/0
RP/0/0/CPU0:PE1(config-if)#int gi0/0/0/1
RP/0/0/CPU0:PE1(config-if)#int gi0/0/0/2
RP/0/0/CPU0:PE1(config-if)#int gi0/0/0/3
RP/0/0/CPU0:PE1(config-if)#int gi0/0/0/5
RP/0/0/CPU0:PE1(config-if)#exit

```

### Gráfico 9-3. Configuración Segment Routing

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

En caso de los equipos Cisco, dicha configuración varía un poco dependiendo de la versión de IOS del router, ya sea XR o XE, sin embargo, el principio es el mismo y este es declarar segment routing dentro del protocolo de enrutamiento OSPF o IS IS.

Para verificar la correcta configuración del protocolo podemos usar el comando *#show mpls forwarding*, como se evidencia en el Gráfico 10-3 presentado a continuación:

```

RP/0/0/CPU0:PE2(config-ospf-ar)#END
RP/0/0/CPU0:PE2#SH MPLS FORWARDING
Sat Jul 18 19:43:54.522 UTC

```

| Local Label | Outgoing Label | Prefix or ID   | Outgoing Interface | Next Hop     | Bytes Switched |
|-------------|----------------|----------------|--------------------|--------------|----------------|
| 24000       | Pop            | SR Adj (idx 0) | Gi0/0/0/0          | 200.10.10.13 | 0              |
| 24001       | Pop            | SR Adj (idx 0) | Gi0/0/0/0          | 200.10.10.13 | 0              |
| 24002       | Pop            | SR Adj (idx 0) | Gi0/0/0/1          | 10.10.9.1    | 0              |
| 24003       | Pop            | SR Adj (idx 0) | Gi0/0/0/1          | 10.10.9.1    | 0              |
| 24004       | Pop            | SR Adj (idx 0) | Gi0/0/0/2          | 10.10.10.1   | 0              |
| 24005       | Pop            | SR Adj (idx 0) | Gi0/0/0/2          | 10.10.10.1   | 0              |
| 24006       | Pop            | SR Adj (idx 0) | Gi0/0/0/3          | 200.10.10.9  | 0              |
| 24007       | Pop            | SR Adj (idx 0) | Gi0/0/0/3          | 200.10.10.9  | 0              |
| 24008       | Pop            | SR Adj (idx 0) | Gi0/0/0/6          | 10.10.12.1   | 0              |
| 24009       | Pop            | SR Adj (idx 0) | Gi0/0/0/6          | 10.10.12.1   | 0              |

```

RP/0/0/CPU0:PE2#

```

### Gráfico 10-3. Configuración Segment Routing

Realizado por : Navarrete, Sebastián & Rivera Patricio. 2019

Además, debemos verificar el rango de las etiquetas que son asignadas por Segment routing, como bien dice la teoría las etiquetas pueden ir desde 16000 hasta 24000, esto es comprobado gracias al comando *#sh mpls label range*, y su afirmación se demuestra en el Gráfico 11-3 mostrado a continuación:

```

PE1#sh mpls label range
Downstream Generic label region: Min/Max label: 16/1048575
Range for Reserved labels:
  1. Range ID: 0 Owner: SR-APP
    Min/Max label: 16000/23999
    Clients Permitted: ISIS OSPF
                       BGP-Segment-Routing TE Policy
    Clients In Use: OSPF
    Dynamic Label Clients In Use: OSPF
    Checkpoint Labels unclaimed: FALSE

```

### Gráfico 11-3. Configuración Segment Routing

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Además, se puede comprobar la correcta integración del IGP con el protocolo Segment routing como se aprecia en el Gráfico 12-3:

```

PE1#show ip ospf 10 segment-routing
      OSPF Router with ID (1.1.1.1) (Process ID 10)
Global segment-routing state: Enabled
Segment routing enabled:
  Area 0 Base Topology name Forwarding Strict SPF
  AS external Base MPLS MPLS Not capable
  AS external Base MPLS MPLS Not applicable
SR Attributes
  Prefer non-SR (LDP) Labels
  Do not advertise Explicit Null
Global Block (SRGB):
  Range: 16000 - 23999
  State: Created
Local Block (SRLB):
  Range: 15000 - 15999
  State: Created
Registered with SR App, client handle: 2
--More--
*Jul 19 03:30:26.673: %SYS-5-CONFIG_I: Configured from console SR algo 0 connected map notifications active (handle 0x0)
  SR algo 0 Active policy map notifications active (handle 0x2), bitmask 0xc
  SR algo 1 Connected map notifications active (handle 0x1), bitmask 0x1
  SR algo 1 Active policy map notifications active (handle 0x3), bitmask 0xc
Registered with MPLS, client-id: 100
Max labels: platform 16, available 13
Max labels pushed by OSPF: uloop tunnels 10, TI-LFA tunnels 10
mfi label reservation ack not pending
Bind Retry timer not running
Adj Label Bind Retry timer not running
sr-app locks requested: srgb 0, srlb 0
SR TEAPP Information:
PE1#
PE1#
PE1#

```

### Gráfico 12-3. Configuración Segment Routing

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Segment routing también nos permite configurar túneles para ingeniería de tráfico, los comandos de configuración los podremos apreciar en los anexos del presente trabajo de investigación. Sin embargo, en el Gráfico 13-3 se puede evidenciar la confirmación de estos:



```

PE1#sh mpls traffic-eng tun
PE1#sh mpls traffic-eng tunnels

P2P TUNNELS/LSPs:

Name: PE1_t1                               (Tunnel) Destination: 7.7.7.7
Status:
  Admin: up          Oper: up | Path: not valid  Signalling: Down
  path option 10, (SEGMENT-ROUTING) type explicit IP_PATH1

Config Parameters:
  Bandwidth: 0          kbps (Global) Priority: 6 6  Affinity: 0x0/0xFFFF
  Metric Type: IGP (interface)
  Path Selection:
  Protection: any (default)
  Path-selection Tiebreaker:
  Global: not set  Tunnel specific: not set  Effective: min-fill (default)
  Hop Limit: disabled
  Cost Limit: disabled
  Path-invalidation timeout: 10000 msec (default), Action: Tear
  AutoRoute: enabled  LockDown: disabled Loadshare: 10 [0]
  auto-bw: disabled
  Fault-OAM: disabled, wrap-Protection: disabled, wrap-Capable: No

History:
  Tunnel:
  Time since created: 2 minutes, 48 seconds
  Number of LSP IDs (Tun_Instances) used: 0

P2MP TUNNELS:

P2MP SUB-LSPS:
PE1#
PE1#

```

### Gráfico 13-3. Configuración Segment Routing

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

Finalmente, podemos comprobar el uso del túnel para ingeniería de tráfico mediante el envío de tráfico a la lo0, esta comprobación se evidencia en el Gráfico 14-3 mostrado a continuación:

```

-----
PE1#trace 192.168.30.1
Type escape sequence to abort.
Tracing the route to 192.168.30.1
VRF info: (vrf in name/id, vrf out name/id)
 0 10.10.11.2 [MPLS: Label 16100 Exp 0] 4 msec 3 msec 3 msec
 1 10.10.12.2 2 msec 4 msec 4 msec
 2 200.10.10.13 9 msec 2 msec *
PE1#

```

### Gráfico 14-3. Asignación de etiqueta única por Segment Routing

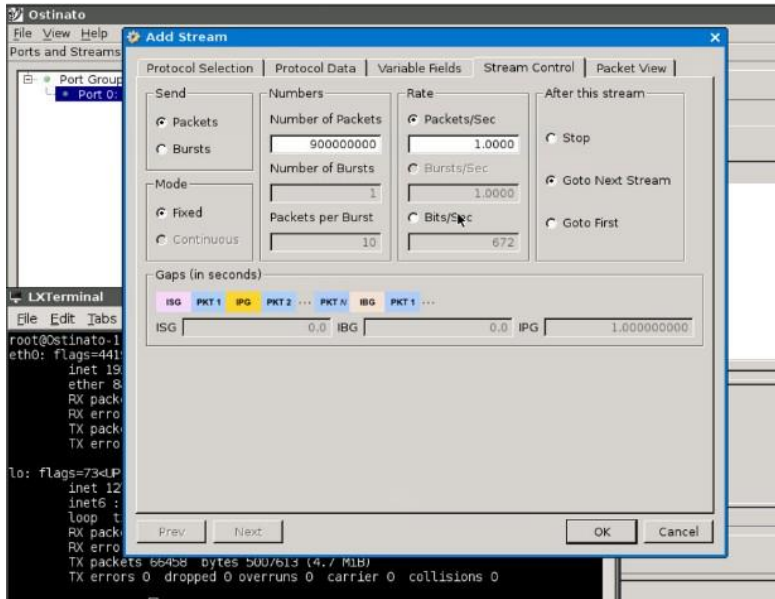
Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

## 3.3. Pruebas de rendimiento

Cabe mencionar que, al tener cuatro túneles configurados con las mismas características de ancho de banda, para la prueba de rendimiento se tomó una muestra aleatoria de un túnel LDP y se evaluó su comportamiento ante la inyección de tráfico

Para las pruebas en los diferentes túneles se procedió con la inyección de paquetes UDP con diferentes tamaños de longitud de la trama evaluando desde la mínima que es 64 bytes hasta llegar a 1500 que es la máxima, a fin de comprobar la eficacia de los túneles configurados, mediante

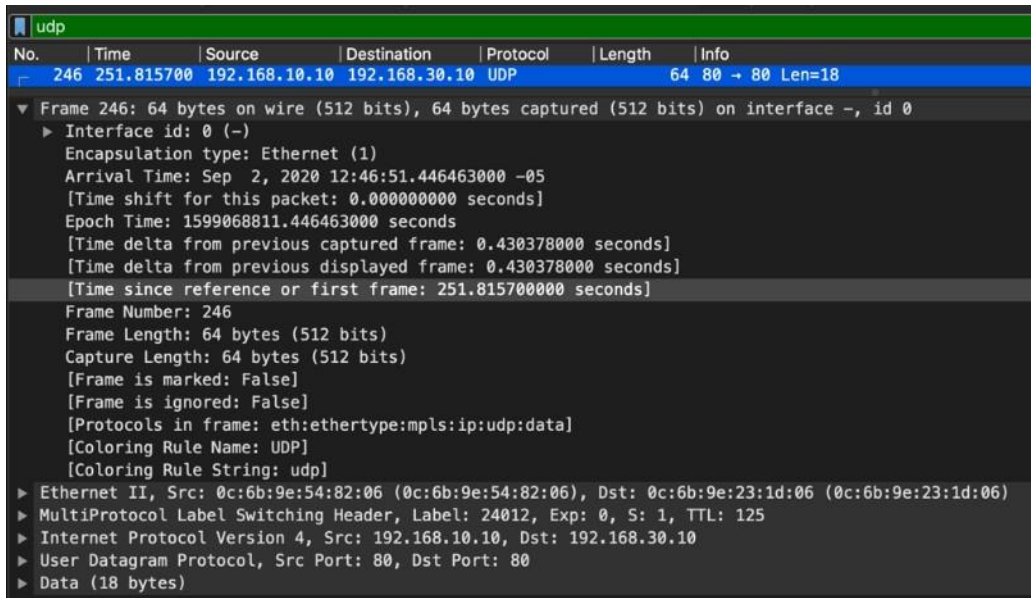
MPLS-TE con LDP y SEGMENT ROUTING TE. Dicha inyección se realizó con Ostinato; la configuración del software la comprobamos en el Gráfico 15-3 presentado a continuación:



**Gráfico 15-3.** Configuración Ostinato

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

El tráfico enviado lo podemos analizar mediante Wireshark como se evidencia en las Gráficas 16-3, 17-3 y 18-3:



**Gráfico 16-3.** Captura MPLS-LDP con 64 bytes de longitud

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

| No. | Time       | Source        | Destination   | Protocol | Length      | Info    |
|-----|------------|---------------|---------------|----------|-------------|---------|
| 206 | 111.403419 | 192.168.10.10 | 192.168.30.10 | UDP      | 750 80 → 80 | Len=704 |

```

▼ Frame 206: 750 bytes on wire (6000 bits), 750 bytes captured (6000 bits) on interface -, id 0
  ▶ Interface id: 0 (-)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 2, 2020 12:50:50.467887000 -05
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1599069050.467887000 seconds
    [Time delta from previous captured frame: 0.108479000 seconds]
    [Time delta from previous displayed frame: 1.000453000 seconds]
    [Time since reference or first frame: 111.403419000 seconds]
  Frame Number: 206
  Frame Length: 750 bytes (6000 bits)
  Capture Length: 750 bytes (6000 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:mpls:ip:udp:data]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
  ▶ Ethernet II, Src: 0c:6b:9e:54:82:06 (0c:6b:9e:54:82:06), Dst: 0c:6b:9e:23:1d:06 (0c:6b:9e:23:1d:06)
  ▶ MultiProtocol Label Switching Header, Label: 24012, Exp: 0, S: 1, TTL: 125
  ▶ Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.30.10
  ▶ User Datagram Protocol, Src Port: 80, Dst Port: 80
  ▶ Data (704 bytes)

```

**Gráfico 17-3.** Captura MPLS-LDP con 750 bytes de longitud

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

| No. | Time      | Source        | Destination   | Protocol | Length       | Info     |
|-----|-----------|---------------|---------------|----------|--------------|----------|
| 30  | 33.832397 | 192.168.10.10 | 192.168.30.10 | UDP      | 1518 80 → 80 | Len=1472 |

```

▼ Frame 30: 1518 bytes on wire (12144 bits), 1518 bytes captured (12144 bits) on interface -, id 0
  ▶ Interface id: 0 (-)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 2, 2020 12:53:02.765479000 -05
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1599069182.765479000 seconds
    [Time delta from previous captured frame: 0.474390000 seconds]
    [Time delta from previous displayed frame: 0.474390000 seconds]
    [Time since reference or first frame: 33.832397000 seconds]
  Frame Number: 30
  Frame Length: 1518 bytes (12144 bits)
  Capture Length: 1518 bytes (12144 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:mpls:ip:udp:data]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
  ▶ Ethernet II, Src: 0c:6b:9e:54:82:06 (0c:6b:9e:54:82:06), Dst: 0c:6b:9e:23:1d:06 (0c:6b:9e:23:1d:06)
  ▶ MultiProtocol Label Switching Header, Label: 24012, Exp: 0, S: 1, TTL: 125
  ▶ Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.30.10
  ▶ User Datagram Protocol, Src Port: 80, Dst Port: 80
  ▶ Data (1472 bytes)

```

**Gráfico 18-3.** Captura MPLS-LDP con 1500 bytes de longitud

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

De igual forma se capturó el tráfico en el escenario con Segment Routing, como se evidencia en las gráficas 19-3, 20-3 y 21-3 mostradas a continuación:

```

231 115.078746 192.168.10.10 192.168.30.10 UDP 64 80 → 80 Len=18
▼ Frame 231: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
▶ Interface id: 0 (-)
  Encapsulation type: Ethernet (1)
  Arrival Time: Sep 3, 2020 12:50:19.220787000 -05
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1599155419.220787000 seconds
  [Time delta from previous captured frame: 1.000281000 seconds]
  [Time delta from previous displayed frame: 1.000281000 seconds]
  [Time since reference or first frame: 115.078746000 seconds]
  Frame Number: 231
  Frame Length: 64 bytes (512 bits)
  Capture Length: 64 bytes (512 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:mpls:ip:udp:data]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
▶ Ethernet II, Src: 0c:be:40:ee:fe:06 (0c:be:40:ee:fe:06), Dst: 0c:be:40:29:73:06 (0c:be:40:29:73:06)
▶ MultiProtocol Label Switching Header, Label: 16100, Exp: 0, S: 1, TTL: 125
▶ Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.30.10
▶ User Datagram Protocol, Src Port: 80, Dst Port: 80
▶ Data (18 bytes)

```

**Gráfico 19-3.** Captura MPLS-SR con 64 bytes de longitud

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

```

217 101.140253 192.168.10.10 192.168.30.10 UDP 750 80 → 80 Len=704
▼ Frame 217: 750 bytes on wire (6000 bits), 750 bytes captured (6000 bits) on interface -, id 0
▶ Interface id: 0 (-)
  Encapsulation type: Ethernet (1)
  Arrival Time: Sep 3, 2020 12:54:54.989760000 -05
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1599155694.989760000 seconds
  [Time delta from previous captured frame: 0.998058000 seconds]
  [Time delta from previous displayed frame: 0.998058000 seconds]
  [Time since reference or first frame: 101.140253000 seconds]
  Frame Number: 217
  Frame Length: 750 bytes (6000 bits)
  Capture Length: 750 bytes (6000 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:mpls:ip:udp:data]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
▶ Ethernet II, Src: 0c:be:40:ee:fe:06 (0c:be:40:ee:fe:06), Dst: 0c:be:40:29:73:06 (0c:be:40:29:73:06)
▶ MultiProtocol Label Switching Header, Label: 16100, Exp: 0, S: 1, TTL: 125
▶ Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.30.10
▶ User Datagram Protocol, Src Port: 80, Dst Port: 80
▶ Data (704 bytes)

```

**Gráfico 20-3.** Captura MPLS-SR con 750 bytes de longitud

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

```

6 23.717043 192.168.10.10 192.168.30.10 UDP 1500 80 → 80 Len=1454
▼ Frame 6: 1500 bytes on wire (12000 bits), 1500 bytes captured (12000 bits) on interface -, id 0
  ▶ Interface id: 0 (-)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 3, 2020 12:56:20.342239000 -05
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1599155780.342239000 seconds
    [Time delta from previous captured frame: 5.671667000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 23.717043000 seconds]
    Frame Number: 6
    Frame Length: 1500 bytes (12000 bits)
    Capture Length: 1500 bytes (12000 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:mpls:ip:udp:data]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
  ▶ Ethernet II, Src: 0c:be:40:ee:fe:06 (0c:be:40:ee:fe:06), Dst: 0c:be:40:29:73:06 (0c:be:40:29:73:06)
  ▶ MultiProtocol Label Switching Header, Label: 16100, Exp: 0, S: 1, TTL: 125
  ▶ Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.30.10
  ▶ User Datagram Protocol, Src Port: 80, Dst Port: 80
  ▶ Data (1454 bytes)

```

**Gráfico 21-3.** Captura MPLS-SR con 1500 bytes de longitud

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Para un mejor entendimiento se presenta la Tabla 1-3, que contiene los valores analizados en los escenarios propuestos:

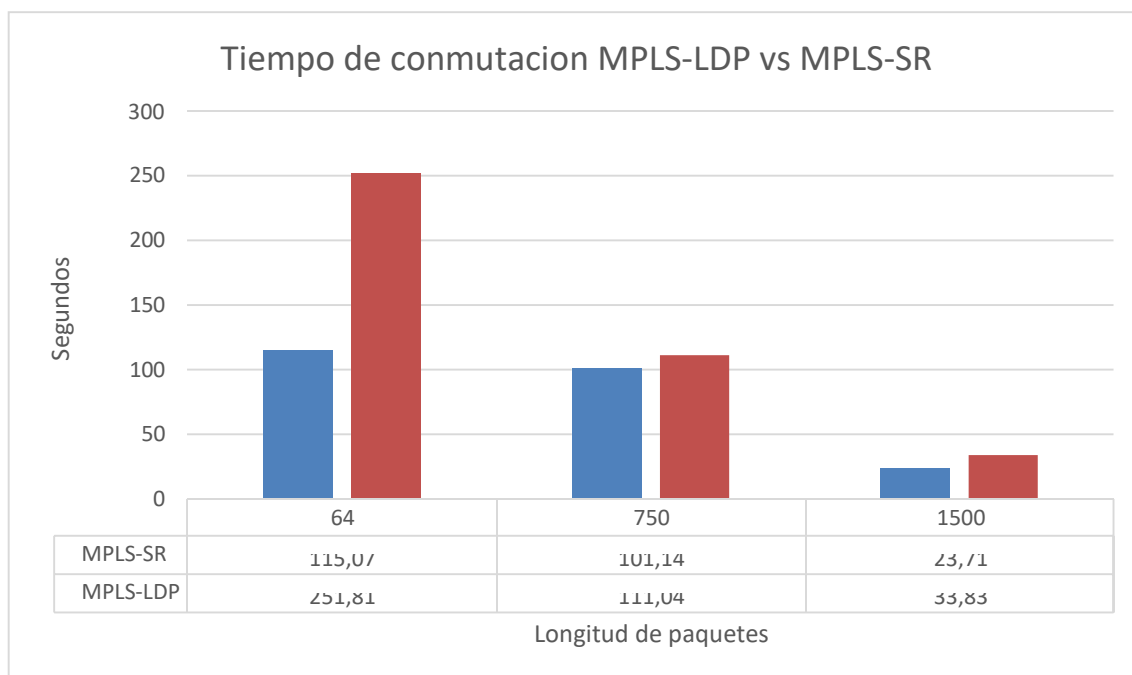
**Tabla 1-3:** Asignación de túneles para comparativa

| LONGITUD DE TRAMA | TIPO DE PAQUETE | CANTIDAD PAQUETES | TIEMPO DE CONMUTACION MPLS-SR | TIEMPO DE CONMUTACION MPLS-LDP |
|-------------------|-----------------|-------------------|-------------------------------|--------------------------------|
| 64                | UDP             | 900000000         | 115,07                        | 251,81                         |
| 750               | UDP             | 900000000         | 101,14                        | 111,04                         |
| 1500              | UDP             | 900000000         | 23,71                         | 33,83                          |

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2019

Además, podemos evidenciar el Gráfico 22-3 en el cual se muestra de forma estadística los resultados obtenidos en Wireshark.





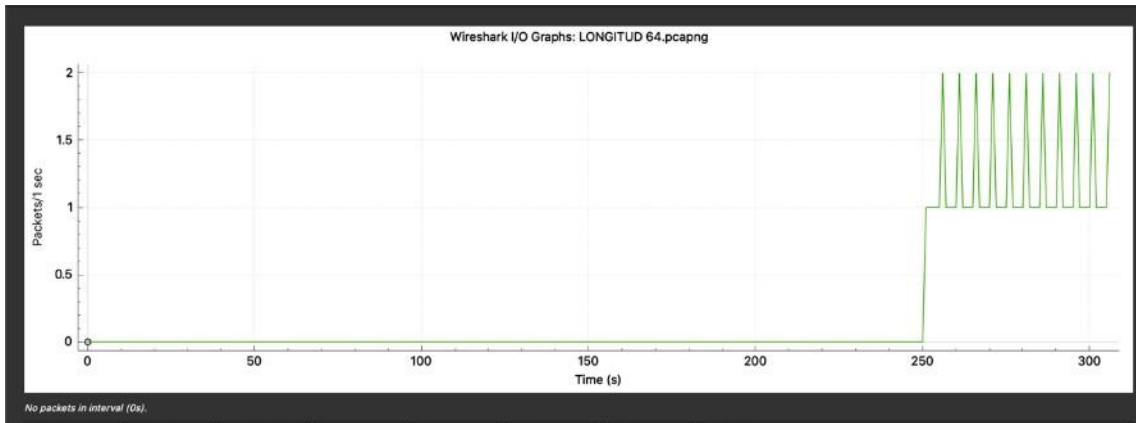
**Gráfico 22-3.** Gráfico comparativo tiempos de conmutación

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Como se muestra en las tablas resumen, Segment Routing, presenta un alto rendimiento en tiempo de conmutación frente a MPLS con Ingeniería de Tráfico convencional, esto es gracias al tiempo de respuesta en el enrutamiento por segmentos.

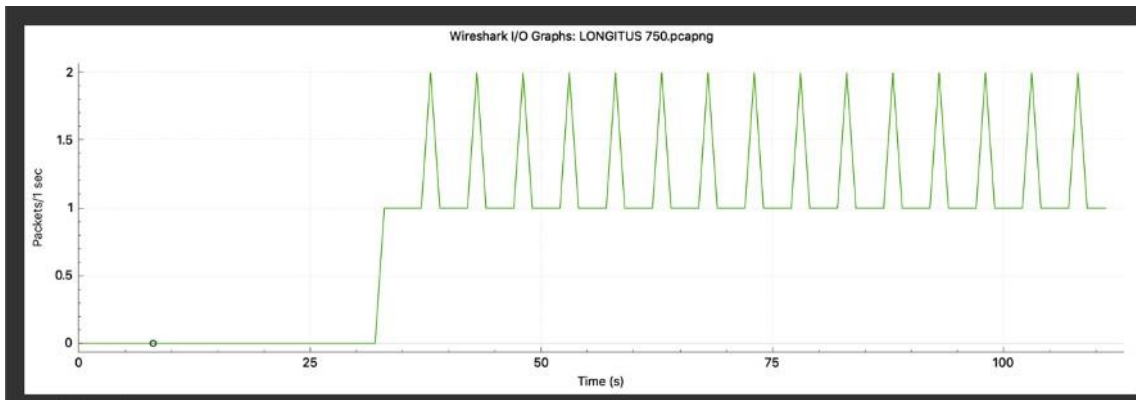
### 3.4. Pruebas de interferencias latencia y Jitter

Gracias a la herramienta I/O Graphic de Wireshark podemos realizar un análisis del comportamiento de la transmisión de los paquetes y a su vez las perdidas en ms con las diferentes longitudes de cabeceras en el escenario. Las estadísticas se pueden evidenciar en los Gráficos 23-3. 24-3 y 254-3 mostrado a continuación:



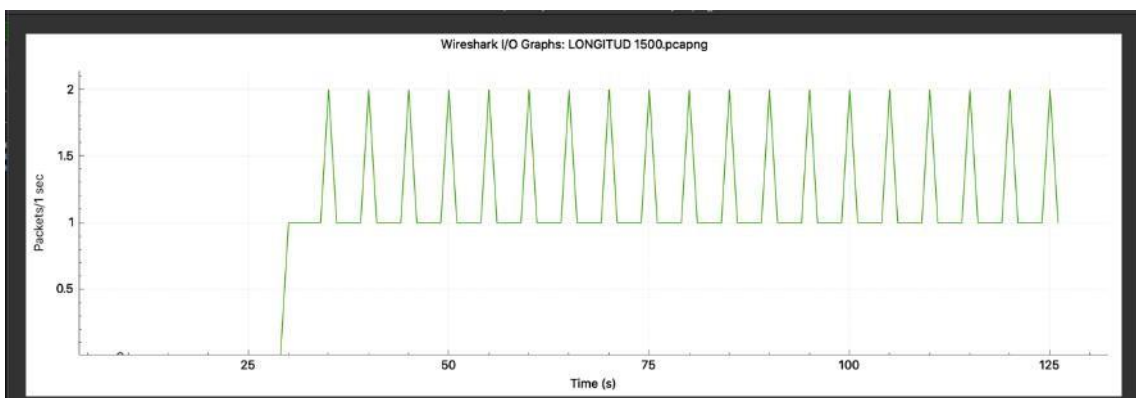
**Gráfico 23-3.** Interferencias en MPLS-LDP con longitud de trama de 64 bytes

Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019



**Gráfico 24-3.** Interferencias en MPLS-LDP con longitud de trama de 750 bytes

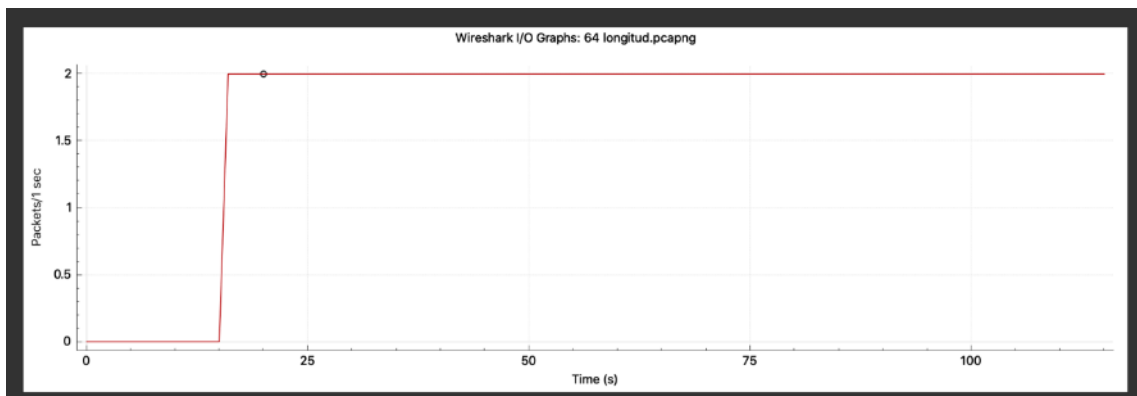
Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019



**Gráfico 25-3.** Interferencias en MPLS-LDP con longitud de trama de 1500 bytes

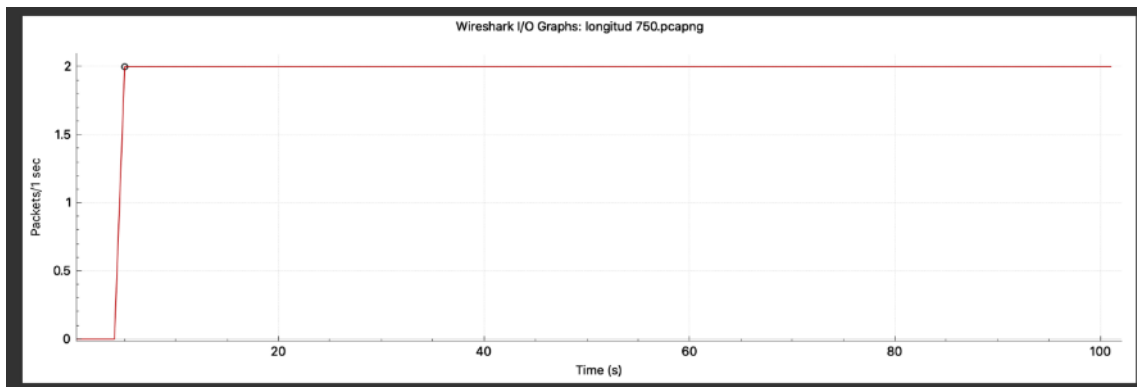
Realizado por: Navarrete, Sebastián & Rivera Patricio. 2019

De igual forma se puede apreciar las interferencias en el escenario configurado con Segment Routing, dichas interferencias se muestran en los Gráficos 26-3, 27-3 y 28-3, así tenemos:



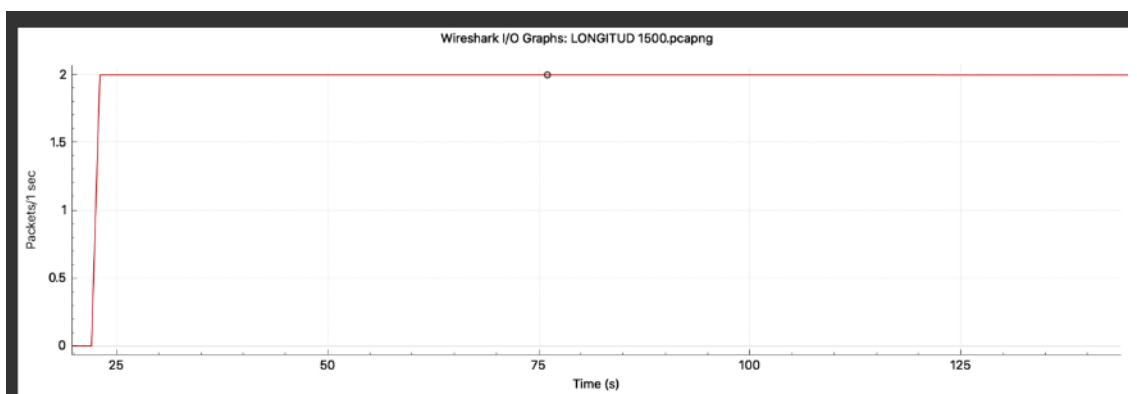
**Gráfico 26-3.** Interferencias en MPLS-SR con longitud de trama de 64 bytes

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019



**Gráfico 27-3.** Interferencias en MPLS-SR con longitud de trama de 750 bytes

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019



**Gráfico 28-3.** Interferencias en MPLS-SR con longitud de trama de 1500 bytes

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019



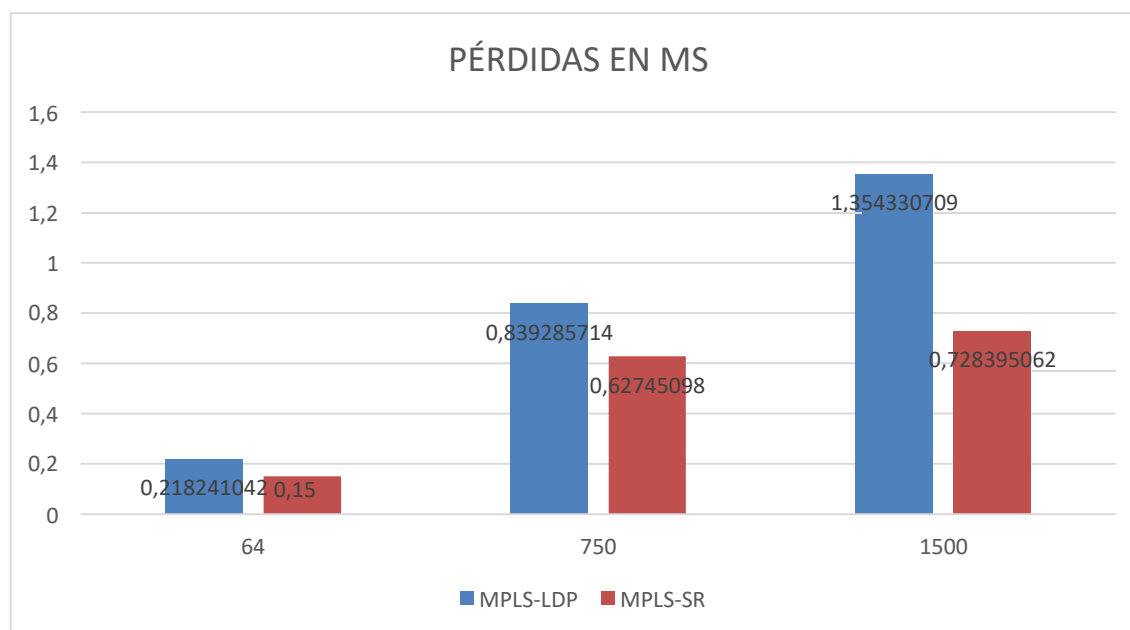
Acorde a lo evidenciado podemos evidenciar que, Segment Routing presenta una notable disminución a las interferencias generadas por la latencia y Jitter del escenario propuesto. Con la opción copy de Wireshark se exportan los resultados a una tabla de Excel, para un mejor entendimiento como se puede apreciar en la Tabla 2-3 mostrada a continuación:

**Tabla 2-3:** Asignación de túneles para comparativa

| LONGITUD DE TRAMA | TIPO DE PAQUETE | CANTIDAD PAQUETES | PERDIDA EN MS MPLS LDP | PERDIDA EN MS MPLS SR |
|-------------------|-----------------|-------------------|------------------------|-----------------------|
| 64                | UDP             | 900000000         | 0,218241042            | 0,15                  |
| 750               | UDP             | 900000000         | 0,839285714            | 0,62745098            |
| 1500              | UDP             | 900000000         | 1,354330709            | 0,728395062           |

**Realizado por:** Navarrete, Sebastián & Rivera Patricio, 2019

De igual forma, en el Gráfico 29-3 se evidencia de forma estadística la diferencia la disminución considerable de interferencias en el escenario configurado con Segment Routing.



**Gráfico 29-3.** Interferencias MPLS-LDP y MPLS-SR

**Realizado por:** Navarrete, Sebastián & Rivera Patricio. 2019

Como hemos visto en la investigación y según afirma la teoría Segment Routing, presenta una considerable mejora ante la ingeniería de tráfico convencional LDP, este perfeccionamiento en la

comunicación es evidente tanto en velocidad de transmisión, al igual que una considerable disminución en las interferencias y paquetes perdidos.

Segment Routing, al ser una tecnología nueva y por ende aun en desarrollo se proyecta a futuro reemplazar las técnicas de ingeniería de tráfico convencionales a fin de elevar la calidad de servicio en las redes de datos.

## CONCLUSIONES

- Las diversas técnicas de ingeniería de tráfico permiten identificar, estructurar metas y prioridades en términos de mejora para garantizar la calidad de servicio, dado que, en la actualidad uno de los mayores inconvenientes es a nivel de redes WAN y el no saber cómo optimizar el ancho de banda en la transmisión de información.
- Segment Routing a diferencia de MPLS se encarga de dividir la red en tres segmentos y asigna un identificador a cada uno de ellos. Este aspecto es anunciado por el protocolo de enrutamiento estándar, lo que permite eliminar la necesidad de ejecutar protocolos de distribución de etiquetas adicionales. Además de permitirnos reducir el tiempo de transmisión, esta técnica también disminuye el consumo de recurso de hardware para el procesamiento de la información.
- Con el estudio de la tecnología Segment Routing, la transmisión en redes WAN incrementa la velocidad de transmisión en aproximadamente un 10 % en relación con la tecnología MPLS tradicional.
- Al realizar el análisis de la eficiencia que tiene Segment Routing en la transmisión de paquetes, se evidenció una notable disminución de pérdidas e interferencias lo que garantiza ser una tecnología muy confiable para la transmisión de datos.
- En la actualidad Segment Routing aún representa una tecnología nueva para muchos fabricantes de equipos de comunicación. Esto provoca que la documentación sea limitada y disminuye la posibilidad de realizar más pruebas frente al comportamiento de dicha tecnología.

## RECOMENDACIONES

- Es aconsejable realizar una investigación más amplia para Segment Routing en los equipos de comunicación más utilizados en la actualidad (Huawei, Juniper, Avaya, HP).
- Como estudiantes familiarizarnos con esta tecnología, dado que Segment Routing se proyecta como una tecnología que podría reemplazar al MPLS convencional que es usado con más frecuencia.
- Se recomienda realizar más investigaciones del comportamiento de Segment Routing, pero con el protocolo de Internet Versión (IPv6), a fin de estudiar su comportamiento y eficiencia con el protocolo mencionado.
- Investigar más métodos de prueba que nos permita evaluar el rendimiento de Segment Routing, y poder analizar el nivel de seguridad que esta tecnología posee; dado que, al ser una tecnología reciente aún se desconoce el nivel de seguridad efectivo.

## **BIBLIOGRAFÍA**

Apolo, C., y Coral, Y. (2017). *Análisis y simulación de tráfico de la red de datos de las fuerzas armadas con tecnologías MPLS (Tesis de pregrado)*. Recuperada de <https://dspace.ups.edu.ec/bitstream/123456789/14289/1/UPS%20-%20ST003131.pdf>

Arista. (2019). *MPLS Segment Routing*. Recuperado de [https://www.arista.com/assets/data/pdf/Whitepapers/MPLSSegmentRouting\\_Whitepaper.pdf](https://www.arista.com/assets/data/pdf/Whitepapers/MPLSSegmentRouting_Whitepaper.pdf)

Bhatia, R., Hao, F., Kodialam, M., y Lakshman, T. (2015). *Optimized Network Traffic Engineering using Segment Routing*. *IEEE Conference on Computer Communications (INFOCOM)*. 657-665. doi: 10.1109/INFOCOM.2015.7218434.

Cisco Systems. (2020). *Segment Routing*. Recuperado de <https://www.segment-routing.net/>

Cisco. (2018). *Pronósticos de tráfico de Internet móvil y fijo de VNI*. Recuperado de <https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html#~cloud-forecast>

Cisco Networking Academy. (2019). *Cisco*. Recuperado de <https://www.netacad.com/es/courses/packet-tracer>

Crow, W. (2016). *Análisis de calidad de servicio en transferencia de voz y video en una red de tecnología MPLS (Multi-Protocol Label Switching) (Tesis de pregrado)*. Recuperada de <http://dspace.esPOCH.edu.ec/bitstream/123456789/6428/1/98T00124.pdf>

Chávez, M. (2003). *Análisis y modelado de mecanismos para la interoperabilidad de MPLS y DiffServ (Tesis de pregrado)*. Recuperada de <https://cicese.repositorioinstitucional.mx/jspui/handle/1007/2685>

Chuy, K. (2019). *Propuesta para la creación y diseño del laboratorio de telecomunicaciones y redes locales II de la Escuela de Mecánica Eléctrica Facultad de*

*Ingeniería, Universidad de San Carlos de Guatemala (Tesis de pregrado)*. Recuperada de

<http://www.repositorio.usac.edu.gt/14203/1/Kenie%20Gary%20Chuy%20Azurdia.pdf>

Delfino, A., Rivero, S., & San Martín, M. (2006). *Ingeniería de Tráfico en Redes MPLS*. Montevideo (Tesis de pregrado). Recuperada de [https://ie.fing.edu.uy/investigacion/grupos/artes-old/fce/net-te/Ingenieria\\_de\\_Trafico\\_en\\_Redres\\_MPLS.pdf](https://ie.fing.edu.uy/investigacion/grupos/artes-old/fce/net-te/Ingenieria_de_Trafico_en_Redres_MPLS.pdf)

EVE-NG Ltd. (2019). *EVE* . Recuperado de <https://www.eve-ng.net/>

Filsfils, C., Kumar, N., Pignataro, C., Cardona, J. C., & Francois, P. (2015). The Segment Routing Architecture. *IEEE*. Recuperado de [https://eprints.networks.imdea.org/1306/1/the\\_segment-routing\\_architecture\\_2015.pdf](https://eprints.networks.imdea.org/1306/1/the_segment-routing_architecture_2015.pdf)

García, E. (2016). *El Ciclo de Deming: La gestión y mejora de procesos*

Recuperado de <https://equipo.altran.es/el-ciclo-de-deming-la-gestion-y-mejora-de-procesos/>

García, A. (2018). *Estudio de la inclusión del sistema PCE en redes GMPLS (Tesis de pregrado)*. Recuperada de <file:///C:/Users/User/Downloads/Proyecto%20PCE.pdf>

Hernández, T. (2015). *Estudio de la ingeniería de tráfico en redes mpls mediante casos de uso práctico con la herramienta VNX (Tesis de posgrado)*. Recuperada de [http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2014-2015/TFM\\_Tatiana\\_Hernandez\\_Camacho\\_2015.pdf](http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2014-2015/TFM_Tatiana_Hernandez_Camacho_2015.pdf)

Huawei. (2019). *Segment Routing*. Recuperado de <https://support.huawei.com/enterprise/en/doc/EDOC1100092117>

Lagua, A. S., y Lugmania, J. C. (2017). *Evaluación de rendimiento e interoperabilidad de una plataforma HW/SW de routing Opensource (Tesis de pregrado)*. Recuperada de <http://dspace.epoch.edu.ec/bitstream/123456789/8435/1/98T00182.pdf>

Luc De Ghein. (2016). *MPLS Fundamentals* (11<sup>a</sup> ed.). Indianapolis: Cisco Press.

Optical Networks. (2019). *¿Qué es la red MPLS y cómo funciona?* Recuperado de <https://www.optical.pe/que-es-una-red-mpls/>

Ostoa, R. (2020). *Análisis comparativo para la selección del protocolo MPLS y SPB para implementar una red backbone de un ISP (T) (Tesis de pregrado)*. Recuperada de <https://core.ac.uk/download/pdf/288312001.pdf> ALTRAN . (2019). *EQUIPO ALTRAN* . Recuperado el 2019, de <https://equipo.altran.es/el-ciclo-de-deming-la-gestion-y-mejora-de-procesos/>

Apolo, C. D., & Coral, Y. X. (2017). *ANÁLISIS Y SIMULACIÓN DE TRÁFICO DE LA RED DE DATOS DE LAS FUERZAS ARMADAS CON TECNOLOGÍAS MPLS*. Quito.

ARISTA. (2019). *MPLS Segment Routing*.

Bhatia, R., Hao, F., Kodialam, M., & Lakshman, T. (2015). *Optimized Network Traffic Engineering using Segment Routing*. *IEEE*.

CISCO. (2018). *Pronósticos de tráfico de Internet móvil y fijo de VNI*. Recuperado el 2019, de <https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html#~cloud-forecast>

Cisco Networking Academy. (2019). *CISCO*. Recuperado el 2019, de <https://www.netacad.com/es/courses/packet-tracer>

Cisco Systems. (2020). *Segment Routing*. Recuperado el 2020, de <https://www.segment-routing.net/>

Delfino, A., Rivero, S., & San Martin, M. (2006). *Ingeniería de Tráfico en Redes MPLS*. Montevideo.

EVE-NG Ltd. (2019). *EVE* . Recuperado el 2020, de <https://www.eve-ng.net/>

Filsfils, C., Kumar Nainar, N., Pignataro, C., Cardona, J. C., & Francois, P. (2015). *The Segment Routing Architecture*. *IEEE*.

García, A. d. (2018). *Estudio de la inclusión del sistema PCE en redes GMPLS*. Barcelona.

Hernandez, T. (2015). *ESTUDIO DE LA INGENIERÍA DETRÁFICO EN REDES MPLS MEDIANTE CASOS DE USO PRÁCTICO CON LA HERRAMIENTA VNX*. Madrid.

Hernández, T. (2015). *ESTUDIO DE LA INGENIERÍA DETRÁFICO EN REDES MPLS MEDIANTE CASOS DE USO PRÁCTICO CON LA HERRAMIENTA VNX*. Madrid.

Huawei. (2019). *Huawei Technologies*. Recuperado el 2019, de <https://support.huawei.com/enterprise/en/doc/EDOC1100092117>

La Sociedad de Internet. (2002). *Descripción general y principios de ingeniería de tráfico de Internet*. Recuperado el 2019, de <https://tools.ietf.org/html/rfc3272>

Lagua, A. S., & Lugmania, J. C. (2017). *EVALUACIÓN DE RENDIMIENTO E INTEROPERABILIDAD DE UNA PLATAFORMA HW/SW DE ROUTING OPENSOURCE*. Riobamba.

Lloyd , R. (2001). *Medición de la Calidad del Servicio*. Mexico.

Luc De Ghein, CCIE. (2016). *MPLS Fundamentals* (11 edición ed.). Indianapolis : Cisco Press.

Networks, O. (2019). *Optical Networks*. Recuperado el 2019, de <https://www.optical.pe/que-es-una-red-mpls/>

NFON. (2019). *NFON Cloud Telephony System*. Recuperado el 2019, de <https://www.nfon.com/es/servicio/base-de-conocimiento/base-de-conocimiento-destacar/redes-de-telecomunicaciones>

Packet . (2020). *Packet*. Recuperado el 2020, de <https://www.packet.com/resources/guides/cisco-virl-on-packet/>

Peralta, J. D. (2016). *SOPORTE DE CALIDAD DE SERVICIOS EN REDES MPLS.DETALLE: MECANISMO DE QOS EN MPLS. EJEMPLO DE APLICACIÓN*. Lima.

Srivats, P. (2019). *Generador de paquetes Ostinato* . Recuperado el 2019, de <https://ostinato.org/>

We Are Social Ltd. (2019). *We are Social*. Recuperado el 2019, de <https://wearesocial.com/global-digital-report-2019>

WireShark. (2019). *WireShark*. Recuperado el 2019, de <https://www.wireshark.org/>



## ANEXOS

### Anexo A: Manual de configuración

|                   |                                    |                       |
|-------------------|------------------------------------|-----------------------|
| <b>ROUTER PE1</b> | conf t                             | Configurar Terminal   |
|                   | hostname PE1                       | Asignar nombre        |
|                   | commit                             | Guardad               |
|                   | int lo0                            | Interface de Loopback |
|                   | ip add 1.1.1.1 255.255.255.255     | Asignación IP         |
|                   | exit                               | Salir                 |
|                   | int gi0/0/0/0                      | Configurar Interfaz   |
|                   | ip add 200.10.10.6 255.255.255.252 | Asignación IP         |
|                   | no shut                            | Activar puerto        |
|                   | exit                               | Salir                 |
|                   | int gi0/0/0/1                      | Configurar Interfaz   |
|                   | ip add 10.10.1.1 255.255.255.0     | Asignación IP         |
|                   | no shut                            | Activar puerto        |
|                   | exit                               | Salir                 |
|                   | int gi0/0/0/2                      | Configurar Interfaz   |
|                   | ip add 10.10.2.1 255.255.255.0     | Asignación IP         |
|                   | no shut                            | Activar puerto        |
|                   | exit                               | Salir                 |
|                   | int gi0/0/0/3                      | Configurar Interfaz   |
|                   | ip add 200.10.10.2 255.255.255.252 | Asignación IP         |
|                   | no shut                            | Activar puerto        |
|                   | exit                               | Salir                 |
|                   | int gi0/0/0/5                      | Configurar Interfaz   |
|                   | ip add 10.10.11.1 255.255.255.0    | Asignación IP         |
|                   | no shut                            | Activar puerto        |
|                   | exit                               | Salir                 |
|                   | commit                             | Guardar               |
|                   | router ospf 10                     | Configurar OSPF       |
|                   | router-id 1.1.1.1                  | Id de Router          |
|                   | area 0                             | Declaracion de area   |
|                   | inter lo0                          | Asignar puerto a OSPF |
|                   | int gi0/0/0/0                      | Asignar puerto a OSPF |
|                   | int gi0/0/0/1                      | Asignar puerto a OSPF |
| int gi0/0/0/2     | Asignar puerto a OSPF              |                       |
| int gi0/0/0/3     | Asignar puerto a OSPF              |                       |
| int gi0/0/0/5     | Asignar puerto a OSPF              |                       |
| exit              | Salir                              |                       |
| commit            | Guardar                            |                       |

|                   |                                       |  |
|-------------------|---------------------------------------|--|
|                   | segment-routing mpls                  | Activar segment routing                              |
|                   | connected-prefix-sid-map              | Asignación de SID                                    |
|                   | address-family ipv4                   | Relacionar SR a IPv4                                 |
|                   | 1.1.1.1/32 100 range 1                | Declarar Interfaz de Loopback                        |
|                   | exit-address-family                   | Salir  |
|                   | ROUTER OSPF 10                        | Configurar OSPF                                      |
|                   | AREA 0                                | Declaracion de area                                  |
|                   | SEGMENT ROUTING MPLS                  | Activar Segment routing                              |
|                   | SEGMENT ROUTING MPLS FORWARDING       | Activar Segment routing forwarding para el protocolo |
|                   | exit                                  | exit   |
|                   | interface tunnel-te11                 | Id de tunel  |
|                   | ipv4 unnumbered loopback0             | Interface de túnel                                   |
|                   | autoroute destination 3.3.3.3         | Declarar siguiente salto                             |
|                   | destination 7.7.7.7                   | Declarar destino de túnel                            |
|                   | path-option 1 dynamic segment-routing | Declarar la prioridad dinámica                       |
|                   | path-protection                       | Activar camino de proteccion                         |
|                   | exit                                  | Salir  |
|                   | commit                                | Guardar  |
|                   | interface tunnel-te11                 | Interface de túnel                                   |
|                   | exit                                  | Salir  |
|                   | commit                                | Guardar  |
|                   |                                       |  |
| <b>ROUTER PE2</b> | conf t                                | Configurar Terminal                                  |
|                   | hostname PE2                          | Asignar nombre                                       |
|                   | commit                                | Guardad  |
|                   | int lo0                               | Interface de Loopback                                |
|                   | ip add 7.7.7.7 255.255.255.255        | Asignación IP  |
|                   | exit                                  | Salit  |
|                   | int gi0/0/0/0                         | Configurar Interfaz                                  |
|                   | ip add 200.10.10.14 255.255.255.252   | Asignación IP  |
|                   | no shut                               | Activar puerto                                       |
|                   | exit                                  | Salir  |
|                   | int gi0/0/0/1                         | Configurar Interfaz                                  |
|                   | ip add 10.10.9.2 255.255.255.0        | Asignación IP  |
|                   | no shut                               | Activar puerto                                       |
|                   | exit                                  | Salir  |
|                   | int gi0/0/0/2                         | Configurar Interfaz                                  |
|                   | ip add 10.10.10.2 255.255.255.0       | Asignación IP  |
|                   | no shut                               | Activar puerto                                       |
|                   | exit                                  | Salir  |
|                   | int gi0/0/0/6                         | Configurar Interfaz                                  |

|               |                                       |  |
|---------------|---------------------------------------|--|
|               | ip add 10.10.12.2 255.255.255.0       | Asignación IP  |
|               | no shut                               | Activar puerto                                       |
|               | exit                                  | Salir  |
|               | int gi0/0/0/3                         | Configurar Interfaz                                  |
|               | ip add 200.10.10.10 255.255.255.252   | Asignación IP  |
|               | no shut                               | Activar puerto                                       |
|               | exit                                  | Salir  |
|               | commit                                | Guardar  |
|               | router ospf 10                        | Configurar OSPF                                      |
|               | router-id 7.7.7.7                     | Id de Router   |
|               | area 0                                | Declaracion de area                                  |
|               | inter lo0                             | Asignar puerto a OSPF                                |
|               | int gi0/0/0/0                         | Asignar puerto a OSPF                                |
|               | int gi0/0/0/1                         | Asignar puerto a OSPF                                |
|               | int gi0/0/0/2                         | Asignar puerto a OSPF                                |
|               | int gi0/0/0/3                         | Asignar puerto a OSPF                                |
|               | int gi0/0/0/6                         | Asignar puerto a OSPF                                |
|               | exit                                  | Salir  |
|               | commit                                | Guardar  |
|               | segment-routing mpls                  | Activar segment routing                              |
|               | connected-prefix-sid-map              | Asignación de SID                                    |
|               | address-family ipv4                   | Relacionar SR a IPv4                                 |
|               | 7.7.7.7/32 100 range 1                | Declarar Interfaz de Loopback                        |
|               | exit-address-family                   | Salir  |
|               | ROUTER OSPF 10                        | Configurar OSPF                                      |
|               | AREA 0                                | Declaracion de area                                  |
|               | SEGMENT ROUTING MPLS                  | Activar Segment routing                              |
|               | SEGMENT ROUTING MPLS FORWARDING       | Activar Segment routing forwarding para el protocolo |
|               | exit                                  | exit   |
|               | interface tunnel-te11                 | Id de tunel  |
|               | ipv4 unnumbered loopback0             | Interface de túnel                                   |
|               | destination 1.1.1.1                   | Declarar destino de túnel                            |
|               | autoroute destination 3.3.3.3         | Declarar siguiente salto                             |
|               | path-option 1 dynamic segment-routing | Declarar la prioridad dinámica                       |
|               | path-protection                       | Activar camino de proteccion                         |
|               | exit                                  | Salir  |
|               | commit                                | Guardar  |
|               |                                       |  |
| <b>ROUTER</b> | conf t                                | Configurar Terminal                                  |
|               | hostname P1                           | Asignar nombre                                       |
|               | commit                                | Guardar  |

|                                 |  |
|---------------------------------|--|
| int lo0                         | Interface de Loopback                                |
| ip add 2.2.2.2 255.255.255.255  | Asignación IP  |
| exit                            | Salir  |
| int gi0/0/0/0                   | Configurar Interfaz                                  |
| ip add 10.10.5.1 255.255.255.0  | Asignación IP  |
| no shut                         | Activar puerto                                       |
| exit                            | Salir  |
| int gi0/0/0/1                   | Configurar Interfaz                                  |
| ip add 10.10.1.2 255.255.255.0  | Asignación IP  |
| no shut                         | Activar puerto                                       |
| exit                            | Salir  |
| int gi0/0/0/2                   | Configurar Interfaz                                  |
| ip add 10.10.3.2 255.255.255.0  | Asignación IP  |
| no shut                         | Activar puerto                                       |
| exit                            | Salir  |
| commit                          | Guardar  |
| router ospf 10                  | Configurar OSPF                                      |
| router-id 2.2.2.2               | Id de Router   |
| area 0                          | Declaracion de area                                  |
| inter lo0                       | Asignar puerto a OSPF                                |
| int gi0/0/0/0                   | Asignar puerto a OSPF                                |
| int gi0/0/0/1                   | Asignar puerto a OSPF                                |
| int gi0/0/0/2                   | Asignar puerto a OSPF                                |
| exit                            | Salir  |
| commit                          | Guardar  |
| segment-routing mpls            | Activar segment routing                              |
| connected-prefix-sid-map        | Asignación de SID                                    |
| address-family ipv4             | Relacionar SR a IPv4                                 |
| 2.2.2.2/32 100 range 1          | Declarar Interfaz de Loopback                        |
| exit-address-family             | Salir  |
| commit                          | Guardar  |
| ROUTER OSPF 10                  | Configurar OSPF                                      |
| AREA 0                          | Declaracion de area                                  |
| SEGMENT ROUTING MPLS            | Activar Segment routing                              |
| SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |
| exit                            | exit   |
| ROUTER OSPF 10                  | Configurar OSPF                                      |
| AREA 0                          | Declaracion de area                                  |
| SEGMENT ROUTING MPLS            | Activar Segment routing                              |
| SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |

|                                 |  |                               |
|---------------------------------|--|-------------------------------|
|                                 | exit   | exit                          |
|                                 |  |                               |
| <b>ROUTER P2</b>                | conf t   | Configurar Terminal           |
|                                 | hostname P2  | Asignar nombre                |
|                                 | commit   | Guardar                       |
|                                 | int lo0  | Interface de Loopback         |
|                                 | ip add 3.3.3.3 255.255.255.255                       | Asignación IP                 |
|                                 | exit   | Salir                         |
|                                 | int gi0/0/0/0  | Configurar Interfaz           |
|                                 | ip add 10.10.6.1 255.255.255.0                       | Asignación IP                 |
|                                 | no shut  | Activar puerto                |
|                                 | exit   | Salir                         |
|                                 | int gi0/0/0/1  | Configurar Interfaz           |
|                                 | ip add 10.10.4.2 255.255.255.0                       | Asignación IP                 |
|                                 | no shut  | Activar puerto                |
|                                 | exit   | Salir                         |
|                                 | int gi0/0/0/2  | Configurar Interfaz           |
|                                 | ip add 10.10.2.2 255.255.255.0                       | Asignación IP                 |
|                                 | no shut  | Activar puerto                |
|                                 | exit   | Salir                         |
|                                 | commit   | Guardar                       |
|                                 | router ospf 10                                       | Configurar OSPF               |
|                                 | router-id 3.3.3.3                                    | Id de Router                  |
|                                 | area 0   | Declaracion de area           |
|                                 | inter lo0  | Asignar puerto a OSPF         |
|                                 | int gi0/0/0/0  | Asignar puerto a OSPF         |
|                                 | int gi0/0/0/1  | Asignar puerto a OSPF         |
|                                 | int gi0/0/0/2  | Asignar puerto a OSPF         |
|                                 | exit   | Salir                         |
|                                 | commit   | Guardar                       |
|                                 | segment-routing mpls                                 | Activar segment routing       |
|                                 | connected-prefix-sid-map                             | Asignación de SID             |
|                                 | address-family ipv4                                  | Relacionar SR a IPv4          |
|                                 | 2.2.2.2/32 100 range 1                               | Declarar Interfaz de Loopback |
|                                 | exit-address-family                                  | Salir                         |
|                                 | commit   | Guardar                       |
| ROUTER OSPF 10                  | Configurar OSPF                                      |                               |
| AREA 0                          | Declaracion de area                                  |                               |
| SEGMENT ROUTING MPLS            | Activar Segment routing                              |                               |
| SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |                               |
| exit                            | exit   |                               |

|               |                                 |                       |
|---------------|---------------------------------|-----------------------|
| ROUTER P3     | conf t                          | Configurar Terminal   |
|               | hostname P3                     | Asignar nombre        |
|               | commit                          | Guardar               |
|               | int lo0                         | Interface de Loopback |
|               | ip add 4.4.4.4 255.255.255.255  | Asignación IP         |
|               | exit                            | Salir                 |
|               | int gi0/0/0/1                   | Configurar Interfaz   |
|               | ip add 10.10.4.1 255.255.255.0  | Asignación IP         |
|               | no shut                         | Activar puerto        |
|               | exit                            | Salir                 |
|               | int gi0/0/0/2                   | Configurar Interfaz   |
|               | ip add 10.10.3.1 255.255.255.0  | Asignación IP         |
|               | no shut                         | Activar puerto        |
|               | exit                            | Salir                 |
|               | int gi0/0/0/3                   | Configurar Interfaz   |
|               | ip add 10.10.7.1 255.255.255.0  | Asignación IP         |
|               | no shut                         | Activar puerto        |
|               | exit                            | Salir                 |
|               | int gi0/0/0/4                   | Configurar Interfaz   |
|               | ip add 10.10.8.1 255.255.255.0  | Asignación IP         |
|               | no shut                         | Activar puerto        |
|               | exit                            | Salir                 |
|               | int gi0/0/0/5                   | Configurar Interfaz   |
|               | ip add 10.10.11.2 255.255.255.0 | Asignación IP         |
|               | no shut                         | Activar puerto        |
|               | exit                            | Salir                 |
|               | int gi0/0/0/6                   | Configurar Interfaz   |
|               | ip add 10.10.12.1 255.255.255.0 | Asignación IP         |
|               | no shut                         | Activar puerto        |
|               | exit                            | Salir                 |
|               | commit                          | Guardar               |
|               | router ospf 10                  | Configurar OSPF       |
|               | router-id 4.4.4.4               | Id de Router          |
| area 0        | Declaracion de area             |                       |
| inter lo0     | Asignar puerto a OSPF           |                       |
| int gi0/0/0/1 | Asignar puerto a OSPF           |                       |
| int gi0/0/0/2 | Asignar puerto a OSPF           |                       |
| int gi0/0/0/3 | Asignar puerto a OSPF           |                       |
| int gi0/0/0/4 | Asignar puerto a OSPF           |                       |
| int gi0/0/0/5 | Asignar puerto a OSPF           |                       |
| int gi0/0/0/6 | Asignar puerto a OSPF           |                       |

|                   |                                 |  |
|-------------------|---------------------------------|--|
|                   | exit                            | Salir  |
|                   | commit                          | Guardar  |
|                   | segment-routing mpls            | Activar segment routing                              |
|                   | connected-prefix-sid-map        | Asignación de SID                                    |
|                   | address-family ipv4             | Relacionar SR a IPv4                                 |
|                   | 4.4.4.4/32 100 range 1          | Declarar Interfaz de Loopback                        |
|                   | exit-address-family             | Salir  |
|                   | commit                          | Guardar  |
|                   | ROUTER OSPF 10                  | Configurar OSPF                                      |
|                   | AREA 0                          | Declaracion de area                                  |
|                   | SEGMENT ROUTING MPLS            | Activar Segment routing                              |
|                   | SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |
|                   | exit                            | exit   |
|                   | ROUTER OSPF 10                  | Configurar OSPF                                      |
|                   | AREA 0                          | Declaracion de area                                  |
|                   | SEGMENT ROUTING MPLS            | Activar Segment routing                              |
|                   | SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |
|                   | exit                            | exit   |
|                   |                                 |  |
| <b>ROUTER P4</b>  | conf t                          | Configurar Terminal                                  |
|                   | hostname P4                     | Asignar nombre                                       |
|                   | commit                          | Guardar  |
|                   | int lo0                         | Interface de Loopback                                |
|                   | ip add 5.5.5.5 255.255.255.255  | Asignación IP  |
|                   | exit                            | Salir  |
|                   | int gi0/0/0/0                   | Configurar Interfaz                                  |
|                   | ip add 10.10.5.2 255.255.255.0  | Asignación IP  |
|                   | no shut                         | Activar puerto                                       |
|                   | exit                            | Salir  |
|                   | int gi0/0/0/1                   | Configurar Interfaz                                  |
|                   | ip add 10.10.9.1 255.255.255.0  | Asignación IP  |
|                   | no shut                         | Activar puerto                                       |
|                   | exit                            | Salir  |
|                   | int gi0/0/0/3                   | Configurar Interfaz                                  |
|                   | ip add 10.10.7.2 255.255.255.0  | Asignación IP  |
|                   | no shut                         | Activar puerto                                       |
|                   | exit                            | Salir  |
|                   | commit                          | Guardar  |
|                   | router ospf 10                  | Configurar OSPF                                      |
| router-id 5.5.5.5 | Id de Router                    |  |

|                  |                                 |  |
|------------------|---------------------------------|--|
|                  | area 0                          | Declaracion de area                                  |
|                  | inter lo0                       | Asignar puerto a OSPF                                |
|                  | int gi0/0/0/0                   | Asignar puerto a OSPF                                |
|                  | int gi0/0/0/1                   | Asignar puerto a OSPF                                |
|                  | int gi0/0/0/3                   | Asignar puerto a OSPF                                |
|                  | exit                            | Salir  |
|                  | commit                          | Guardar  |
|                  | segment-routing mpls            | Activar segment routing                              |
|                  | connected-prefix-sid-map        | Asignación de SID                                    |
|                  | address-family ipv4             | Relacionar SR a IPv4                                 |
|                  | 5.5.5.5/32 100 range 1          | Declarar Interfaz de Loopback                        |
|                  | exit-address-family             | Salir  |
|                  | commit                          | Guardar  |
|                  | ROUTER OSPF 10                  | Configurar OSPF                                      |
|                  | AREA 0                          | Declaracion de area                                  |
|                  | SEGMENT ROUTING MPLS            | Activar Segment routing                              |
|                  | SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |
|                  | exit                            | exit   |
|                  | ROUTER OSPF 10                  | Configurar OSPF                                      |
|                  | AREA 0                          | Declaracion de area                                  |
|                  | SEGMENT ROUTING MPLS            | Activar Segment routing                              |
|                  | SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |
|                  | exit                            | Salir  |
|                  |                                 |  |
| <b>ROUTER P5</b> | conf t                          | Configurar Terminal                                  |
|                  | hostname P5                     | Asignar nombre                                       |
|                  | commit                          | Guardar  |
|                  | int lo0                         | Interface de Loopback                                |
|                  | ip add 6.6.6.6 255.255.255.255  | Asignación IP  |
|                  | exit                            | Salir  |
|                  | int gi0/0/0/0                   | Configurar Interfaz                                  |
|                  | ip add 10.10.6.2 255.255.255.0  | Asignación IP  |
|                  | no shut                         | Activar puerto                                       |
|                  | exit                            | Salir  |
|                  | int gi0/0/0/2                   | Configurar Interfaz                                  |
|                  | ip add 10.10.10.1 255.255.255.0 | Asignación IP  |
|                  | no shut                         | Activar puerto                                       |
|                  | exit                            | Salir  |
|                  | int gi0/0/0/4                   | Configurar Interfaz                                  |
|                  | ip add 10.10.8.2 255.255.255.0  | Asignación IP  |



|                                 |  |
|---------------------------------|--|
| no shut                         | Activar puerto                                       |
| exit                            | Salir  |
| router ospf 10                  | Guardar  |
| router-id 6.6.6.6               | Configurar OSPF                                      |
| area 0                          | Id de Router   |
| inter lo0                       | Declaracion de area                                  |
| int gi0/0/0/0                   | Asignar puerto a OSPF                                |
| int gi0/0/0/4                   | Asignar puerto a OSPF                                |
| int gi0/0/0/2                   | Asignar puerto a OSPF                                |
| exit                            | Salir  |
| commit                          | Guardar  |
| segment-routing mpls            | Activar segment routing                              |
| connected-prefix-sid-map        | Asignación de SID                                    |
| address-family ipv4             | Relacionar SR a IPv4                                 |
| 6.6.6.6/32 100 range 1          | Declarar Interfaz de Loopback                        |
| exit-address-family             | Salir  |
| commit                          | Guardar  |
| ROUTER OSPF 10                  | Configurar OSPF                                      |
| AREA 0                          | Declaracion de area                                  |
| SEGMENT ROUTING MPLS            | Activar Segment routing                              |
| SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |
| exit                            | exit   |
| ROUTER OSPF 10                  | Configurar OSPF                                      |
| AREA 0                          | Declaracion de area                                  |
| SEGMENT ROUTING MPLS            | Activar Segment routing                              |
| SEGMENT ROUTING MPLS FORWARDING | Activar Segment routing forwarding para el protocolo |
| exit                            | Salir  |

## Anexo B: Configuraciones en Equipos

```
!!!!!!!!!!!!!!!!!!!! NO root-system username is configured. Need to configure root-system username. !!!!!!!!!!!!!!!!!!!!!  
  
--- Administrative User Dialog ---  
  
Enter root-system username: PE1  
Enter secret: █  
  
Ready Telnet: 192.168.227.128 40, 17 40 Rows, 96 Cols VT100 CAP NUM ..
```

```
MON JUN 22 20:12:02.931 UTC  
RP/0/0/CPU0:PE1(config)#mpls ldp  
RP/0/0/CPU0:PE1(config-ldp)#router-id 1.1.1.1  
RP/0/0/CPU0:PE1(config-ldp)#inter gi0/0/0/0  
RP/0/0/CPU0:PE1(config-ldp-if)#no int gi0/0/0/0  
RP/0/0/CPU0:PE1(config-ldp)#  
RP/0/0/CPU0:PE1(config-ldp)#  
RP/0/0/CPU0:PE1(config-ldp)#inter gi0/0/0/2  
RP/0/0/CPU0:PE1(config-ldp-if)#exit  
RP/0/0/CPU0:PE1(config-ldp)#inter gi0/0/0/1  
RP/0/0/CPU0:PE1(config-ldp-if)#exit  
RP/0/0/CPU0:PE1(config-ldp)#exit  
RP/0/0/CPU0:PE1(config)#commit  
Mon Jun 22 20:13:15.622 UTC  
RP/0/0/CPU0:PE1(config)#  
RP/0/0/CPU0:PE1(config)#  
RP/0/0/CPU0:PE1(config)#inter gi0/0/0/2  
RP/0/0/CPU0:PE1(config-if)#mtu 9000  
RP/0/0/CPU0:PE1(config-if)#exit  
RP/0/0/CPU0:PE1(config)#inter gi0/0/0/1  
RP/0/0/CPU0:PE1(config-if)#mtu 9000  
RP/0/0/CPU0:PE1(config-if)#exit  
RP/0/0/CPU0:PE1(config)#commit  
Mon Jun 22 20:13:24.931 UTC  
RP/0/0/CPU0:PE1(config)#  
RP/0/0/CPU0:PE1(config)#  
RP/0/0/CPU0:PE1(config)#  
RP/0/0/CPU0:PE1(config)#end
```

t DGIG G I #1 DG G I t DG I G  
RP/0/0/CPU0:PEZ(config)#mpJs JabeJ ranqe 18000 19999  
RP/D/D/CPMO : PE2 {CDnfn g)#npl s l dp  
RP/D/D/CPM0 : PE2 {confñ g—l dp)#r outer —1 d 7. 7. 7. 7  
RP/D/D/CPMO : PE2 {confñ g—l dp)#ñ rrcer gñ D/D/D/2  
RP/D/D/CPMD : PE2 corrf ñ g—l dp—1 f)#ex1z  
RP/D/D/CPM0 : PE2 {confñ g—l dp)#1rrcer g1D/D/D/1  
RP/D/D/CPU0 : PE2 (corrFl g-1 dp-1f)#ex1t  
RP/D/D/CPM0 : PE2 (corrFl g-1 dp)#ex1z  
RP/D/D/CPU0 : PE2 (confl g)#coou1Jr  
Mon l un ZZ ZO : Z9 : 41. Z18 UTC  
RP/D/D/CPU0 : PE2 { conf ñ g)#  
R /o/o/cdna:Nez « <r g>+1rrrer g1o/o/o/z  
RP/D/D/CPMO : PE2 {confñ g—ñ f)#ntu 9DDD  
RP/D/D/CPM0 : PE2 {confñ g—1 f)#ex1z  
RP/D/D/CPuO : PE2 { corrf ñ g)#1FTLeF g1D/D/D/1  
RP/D/D/CPMD : PE2 {confñ g—1 f)#nt:u 9DDD  
RP/D/D/CPM0 : PE2 {confñ g—1 f)#ex1z  
RP/D/D/CPMO : PE2 CDFr q §)6CDITIti# 1

RP/D/D/CPUD : PZ { c onfl g)#npl s l abel r ange 22DDD 23999  
R /D/D/c uD : Hz confJ g)#opi s dp  
RP/D/D/CPMD : PZ {confñ g—l dp)#r outer —ñ d Z . 2. 2. Z  
RP/D/D/CPUD : P2 {confñ g—l dp)'ñ rrcer gñ D/D/D/D  
RP/D/D/CPUD : P2 {confñ g—l dp —z f)#E'X0 t:  
RP/D/D/CPUD : P2 {confñ g—l dp)#ñ rrcer gñ D/D/D/2  
RP/D/D/CPUD : P2 {confñ g—l dp —z f)#E'X0 t:  
RP/D/D/CPMD:P2tconfJg—ld )#J rrcRr gJ D/D/D/1  
RP/D/D/CPUD : P2 {confñ g—l dp —z f)#E'X0 t:  
RP/D/D/CPMD : P2 {confñ g—l dp)#exñ t:  
RP/D/D/CPUD : P2 SCO §)6CDITIti? l  
Plan 3 un 22 16 : 47 :19. 4D3 UTC  
RP/D/D/CPUD : PZ COFif g)#  
R /o/o/c uo: z<ronf1p>», rrcer p< o/o/o/o  
R /o/o/c uo: z<ro<r, g—, l)+mu s000  
RP/0/0/CPU0:P2(config-if)#exit  
R /o/o/c uo: Hz ( c onfig)+1rrcer p< o/o/o/z  
R /o/o/c uo: z<ro<r<p—<f )amu s000  
RP/0/0/CPU0:P2(config-if)#exit  
R /o/o/c uo: z<contig>a1ncer g< o/o/o/i  
R /D/D/c uD: z«>><, t—, Y)»mu 9DDD  
RP/0/0/CPU0:P2(config-if)#exit  
RP/D/D/CPUD : PZ SCO §)6CDITIti? l  
Plan 3 un 22 16 : 47 : 21. 1D3 UTC  
RP /D/D/CPUD : PZ € COFif O36

RP/a/0/cPu0 : P 3 (confñ g)#mp1 3 l abel r ange 24000 25999  
RP/D/0/CPU0 : P 3 {confñ g}#mpl 3 l dp  
RP/0/0/CPU0 : P 3 (confñ g—l dp)#r outer —1 d 4. 4. 4. 4  
RP/D/0/CPU0 : P 3 {confñ g—l dp}#ñ nder g 0/0/0/ 3  
RP/0/0/CPU0 : P 3 (confñ g—l dp—ñ f)#ex1z  
RP/D/0/CPU0 : P 3 {confñ g—l dp}#ñ nder g 0/0/0/2  
RP/D/0/CPU0 : P 3 (confñ g—l dp—ñ f)#ex1z  
RP/D/0/CPU0 : P 3 {confñ g—l dp}#ñ nder g 0/0/0/1  
RP/D/0/CPU0 : P 3 {Confñ g—l dp—ñ f)#ex1c  
RP/D/0/CPU0 : P 3 {confñ g—l dp}#ñ nder g 0/0/0/4  
RP/D/0/CPU0 : P 3 Confñ g—l dp—ñ f)#ex1c  
RP/D/0/CPU0 : P 3 {confñ g—l dp} #e xñ c  
RP/D/0/CPU0 : P 3 {confñ g} Scored c  
MOn Jun 22 16 : 5 3 : 20. 304 UTC  
RP/D/0/CPU0 : P 3 {confñ g}#  
RP/a/0/cPu0: P 3(confñ g)#1nder gñ 0/0/0/ 3  
RP/D/0/CPU0 : P 3 {confñ g—1f)#mzu 9D00  
RP/0/0/CPU0 : P 3 (confñ g—1f)#exñ z  
RP/D/0/CPU0 : P 3 {confñ g}#1nder gñ D/0/0/2  
RP/0/0/CPU0 : P 3 (confñ g—1f)#wu 9O00  
RP/D/0/CPU0 : P 3 {confñ g—1f)#exñ c  
RP/D/0/CPU0 : P 3 (confñ g)#1nder gñ D/0/0/1  
RP/D/0/CPU0 : P 3 {confñ g—1f)#mzu 9D00  
RP/D/0/CPU0 : P 3 (confñ g—1f)#exñ z  
RP/D/0/CPU0 : P 3 {confñ g}#1nder gñ D/0/0/4  
RP/D/0/CPU0 : P 3 {confñ g—1f)#rircu 9D00  
RP/D/0/CPU0 : P 3 {confñ g—1f)#exñ c

RP/0/D/CPU0 : P4 {confñ g}#mpl s l abel r ange 2600o z7g99  
RP/0/D/CPU0 : P4 (config) #mpJs Gdp  
RP/0/a/cPu0 : P4 {confñ g—l dp}#r outer —1 d 5. 5. 5. 5  
RP/0/D/CPU0 : P4 {confñ g—l dp}#ñ nder g1 0/0/0/ 3  
RP/0/D/CPU0 : P4 {confñ g—l dp—ñ f)#ex1z  
RP/0/D/CPU0 : P4 {confñ g—l dp}#ñ nder g1 0/0/0/1  
RP/0/D/CPU0 : P4 {confñ g—l dp—ñ f)#ex1z  
RP/0/D/CPU0 : P4 {confñ g—l dp}#ñ nder g1 0/0/0/0  
RP/0/D/CPU0 : P4 {confñ g—l dp—ñ l} #e x1 c  
RP/0/D/CPU0 : P4 {confñ g—l dp} #e xñ c  
RP/0/D/CPU0 : P4 {confñ g} Scored z  
non 3 un 22 16 : 47 : 44. 00 3 UTC  
RP/0/D/CPU0 : P4 {confñ g}#  
RP/0/O/CPU0 : P4 {confñ g}#1nder gñ 0/0/0/3  
R /o/D/c Uo: < (conf 1g-1f) arcu 9000  
RP/0/D/CPU0 : P4 {confñ g—ñ f)#exñ c  
RP/0/D/CPU0 : P4 (config) #inter gi0/0/0/1  
RP/0/a/cPu0 : P4 {confñ g—ñ f)#nzcu 9000  
RP/0/D/CPU0 : P4 {confñ g—ñ f)#exñ z  
RP/0/D/CPU0 : P4 (confñg) #4nter gñ0/0/0/0  
RP/0/D/CPU0 : P4 {confñ g—ñ f)#wcu 9000  
RP/0/D/CPU0 : P4 {confñ g—ñ f)#exñ z  
RP/0/D/CPU0 : P4 {confñ g} Scored c  
non 3 un 22 16 : 47 : 45. 60 3 UTC  
RP/0/0/CPU0 : P4 (confñ) #

```

RP,, 0,, 0,, 'C PU0 : P 5 (confl g).4npl s l abel r ange 2 6000 2 7999
RP,, ' 0,, ' 0,, 'C PLI0 : P 5 (confl g) nipl s l dp
RP,, '0,, '0,, 'CPLI0 : P 5 (confl g—l dp) r ork t er —1 d 6. 6. 6. 6
RP,, '0,, '0,, 'CPLI0 : P 5 (confl g—l dp) 1 nt er g1 0,, '0,, '0,, '0
RP,, '0,, '0,, 'CPLI0 : P 5 (confl g—l dp—lf), 4ext l t
RP,, '0,, '0,, 'CPU0 : P 5 (confl g—l dp) .41 nt er g1 0,, 0,, 0,, 2
RP,, '0,, '0,, 'CPU0 : P 5 (confl g—l dp—lf).text t
RP,, '0,, '0,, 'CPLI0 : P 5 (confl g—l dp) || rat er g1 0,, '0,, '0,, '4
RP,, '0,, '0,, 'CPLI0 : P 5 (confl g—l dp—lf) text t
RP,, '0,, '0,, 'CPLI0 : P 5 (cofafl g—l dp) exit
RP,, 0,, 0,, 'C PLI0 : P 5 (confl g) cont t
fion 3 rn 2 2 16 : 48 : 19 . 74 0 LITC
RP,, 0,, 0,, 'C PU0 : P 5 (confl g).4
RP,, '0,, '0,, 'CPLI0 : P 5 (confl g)é1 rat er g 0,, '0,, '0,, 2
RP,, '0,, '0,, 'cpU0 : 5 (confl g—lf) Unit r 9000
RP,, '0,, '0,, 'C PLI0 : P 5 (cofafl g—lf) text t
RP,, '0,, '0,, 'C PLI0 : P 5 (confl g) 1 nt er g1 0,, '0,, '0,, '0
RP,, 0,, 0,, 'CPU0 : P 5 (confl g—l) , 4nit r 9000
RP,, 0,, 0,, 'C PU0 : P 5 (confl g—lf) .next t
RP,, '0,, '0,, 'CPLI0 : P 5 (confl g) || rat er g1 0,, '0,, '0,, '4
RP,, '0,, '0,, 'cpU0 : 5 (confl g—lf) Unit r 9000
RP,, '0,, '0,, 'C PLI0 : P 5 (cofafl g—lf) text t
RP,, 0,, 0,, 'C PLI0 : P 5 (confl g) cont t

```

PE1 x

User Access Verification

U3er name:  
U3er name: PE1  
Pas'st'.or d:

RP/0/0/CPU0: PPLVs hot\*. 'upl s for \*. 'ar dang  
hon l r i 8 D3: D3: LB. 2 38 UTC

| Local Label | Out gon ng Label | Pr ebb x or ID              | Oral gO ng Int erface | Ne xt H0p             | By es 5 'J t ched |
|-------------|------------------|-----------------------------|-----------------------|-----------------------|-------------------|
| "L6000      | Pop              | 2 . 2 . 2 . 2 / 32          | GI 0/0/0/ L           | LO . LO . 1 . 2       | L 5 54 PLO        |
| L&00i       | 2 0D0 D          | 5 . 5 . 5 . 5 / 32          | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &002     | 2 000 L          | T . T . 7 . T / 32          | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &003     | POp              | L0 . LO . 5 . D / 24        | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &004     | POp              | L0 . LO . 3 . D / 24        | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &005     | 20D02            | L0 . LO . 9 . 0 / 2 4       | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &00&     | 2 001L           | 2 00 . 10 . 1D . "L2/ J0    | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| L60 0T      | 2 0D0 3          | 2 00 . 10 . 1D . 8 / 30     | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L 60 08    | 2 0D04           | L0 . LO . 10 . 0 / 2 4      | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &009     | 2 0D0 5          | L0 . LO . 12 . 0 / 2 4      | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| L60L0       | unl abel led     | L92 . 1 68 . 2 0 . D / 2 -1 | GI 0/0/0/ D           | 2 00 . 1D . "L 0 . 5  | D                 |
| "L &0" L"   | 20D2D            | L92 . 1 68 . 30 . D / 24    | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &0 "L2   | 20D06            | L92 . 168 . 4 0 . D / 24    | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
| "L &0 "L3   | Un1 abet ed      | L92 . 1 68 . L0 . D / 24    | GI 0/0/0/ 3           | 2 00 . 1D . "L 0 . "L | D                 |
| "L &0 "L4   | Un1 abet ed      | 3 . 3 . 3 . 3 / 32          | GI 0/0/0/ 2           | LO . LO . 2 . 2       | D                 |
|             |                  |                             | GI 0/0/0/ L           | LO . LO . 1 . 2       | D                 |
|             |                  |                             | cJ 0/0/0/ 2           | LO . LO . 2 . 2       | L6018L0           |
|             |                  |                             | 01 0/0/0/ 2           | LO . LO . 2 . 2       | D                 |
|             |                  |                             | GI 0/0 /0/ L          | LO . LO . 1 . 2       | D                 |
|             |                  |                             | GI 0/0/0/ 2           | LO . LO . 2 . 2       | D                 |

TO 6 UTI ab e t ed LO "TO 4'D/24  
L&0L" url abel led T.O . D/2  
—nor e—g

Ready Telnet: 19a.168.22?.12a 33,



## Anexo C: Túneles

```
RP/0/0/CPU0:PE1#SH MPLS
% Incomplete command.
RP/0/0/CPU0:PE1#SH MPLS Forwarding
Thu Jul 16 18:26:34.484 UTC
```

| Local Label | Outgoing Label | Prefix or ID    | Outgoing Interface | Next Hop    | Bytes Switched |
|-------------|----------------|-----------------|--------------------|-------------|----------------|
| 16000       | Pop            | 2.2.2.2/32      | Gi0/0/0/1          | 10.10.1.2   | 9382           |
| 16001       | Pop            | 10.10.5.0/24    | Gi0/0/0/1          | 10.10.1.2   | 0              |
| 16002       | 20000          | 4.4.4.4/32      | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 4.4.4.4/32      | Gi0/0/0/2          | 10.10.2.2   | 44             |
| 16003       | 20006          | 7.7.7.7/32      | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 7.7.7.7/32      | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16004       | Pop            | 10.10.3.0/24    | Gi0/0/0/1          | 10.10.1.2   | 0              |
| 16005       | 20004          | 10.10.12.0/24   | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 10.10.12.0/24   | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16006       | 20002          | 10.10.7.0/24    | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 10.10.7.0/24    | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16007       | 20003          | 10.10.8.0/24    | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 10.10.8.0/24    | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16008       | 20010          | 10.10.9.0/24    | Gi0/0/0/1          | 10.10.1.2   | 0              |
| 16009       | 20011          | 10.10.10.0/24   | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 10.10.10.0/24   | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16010       | Unlabelled     | 192.168.10.0/24 | Gi0/0/0/3          | 200.10.10.1 | 0              |
| 16011       | 20014          | 200.10.10.8/30  | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 200.10.10.8/30  | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16012       | 20015          | 200.10.10.12/30 | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 200.10.10.12/30 | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16013       | Unlabelled     | 192.168.20.0/24 | Gi0/0/0/0          | 200.10.10.5 | 0              |
| 16014       | Unlabelled     | 3.3.3.3/32      | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16015       | Unlabelled     | 10.10.4.0/24    | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16016       | Unlabelled     | 10.10.6.0/24    | Gi0/0/0/2          | 10.10.2.2   | 0              |
| 16018       | 20019          | 5.5.5.5/32      | Gi0/0/0/1          | 10.10.1.2   | 0              |
| 16019       | 20020          | 192.168.40.0/24 | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 192.168.40.0/24 | Gi0/0/0/2          | 10.10.2.2   | 1200           |
| 16020       | 20021          | 192.168.30.0/24 | Gi0/0/0/1          | 10.10.1.2   | 0              |
|             | Unlabelled     | 192.168.30.0/24 | Gi0/0/0/2          | 10.10.2.2   | 0              |

```
RP/0/0/CPU0:PE1#
```

```
Name: tunnel-te3 Destination: 7.7.7.7 Ifhandle:0xd0
Signalled-Name: PE1_t3
Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 1, type explicit A2_A3 (Basis for Setup, path weight 3)
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 0 kbps CT0
  Creation Time: Thu Jul 16 17:54:10 2020 (04:29:01 ago)
Config Parameters:
  Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
  Metric Type: TE (global)
  Path Selection:
    Tiebreaker: Min-fill (default)
  Hop-limit: disabled
  Cost-limit: disabled
  Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
  AutoRoute: disabled LockDown: disabled Policy class: not set
  Forward class: 0 (default)
  Forwarding-Adjacency: disabled
  Autoroute Destinations: 0
  Loadshare: 0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Disabled, Protection Desired: None
  Path Protection: Not Enabled
  BFD Fast Detection: Disabled
  Reoptimization after affinity failure: Enabled
  Soft Preemption: Disabled
History:
  Tunnel has been up for: 00:03:49 (since Thu Jul 16 22:19:22 UTC 2020)
Current LSP:
  Uptime: 00:03:49 (since Thu Jul 16 22:19:22 UTC 2020)
Reopt. LSP:
  Last Failure:
    LSP not signalled, identical to the [CURRENT] LSP
    Date/Time: Thu Jul 16 18:12:24 UTC 2020 [04:10:47 ago]
Prior LSP:
  ID: 2 Path Option: 1
  Removal Trigger: path verification failed

Path info (OSPF 10 area 0):
Node hop count: 3
Hop0: 10.10.1.2
Hop1: 10.10.5.1
Hop2: 10.10.5.2
Hop3: 10.10.9.1
Hop4: 10.10.9.2
```

```

Name 7unnet e4 De set nat Ion: 7. .7.7 lfh andl e:Oxf0
na11ed-Name PE1_t4
St at us
  admJn: up oper: up Path: valid signall Jng:connected

parh optJon l, type explJcir A2_A4 (BasJs for setup, path weight 2)
G-PID :0x08 00 (der1ved Mr or egr es s 1nder lac e pr oper 71 es)
Bandwldt h Requ est ed: 0 kbps CTO
Creation Time: Thu Jul 16 17:55:18 2020 (04:27:53 ago)
conf q Parameters
Band;vJdih: 0 kbps (c+D) Priority: ? ? Affiniy: DxD/0xffff
new ric Type: vE (gl abas)
Path Set ect an
  Tl ebr eaken : Mt n-fi11 (def au1t)
HOp -1 J ml t: d1sabl ed
cost-l mst: di sabl ed
Path-l n a fi data on ti me out : 10000 asec (def au1t) , acid on: Team (def aul t)
Aux oRoute: d1sabl ed L oCkDown: d1sabl ed Pot cy ct ass: not set
For war d cl ass : 0 (defau t)
For car di ng -Adj ac ency : dJ sabl ed
out or oute DeStn narl ons : 0
Loadshar e: O equal loads har es
Aux a—but : di s ab1e d
Fast Rer oute: DI sabl ed , Pr otectl on Dest r ed: None
Path Pr ot ect 1 on: Not Enabled
BrD rast Det ectl on: D1S6b1 ed
ReoptJmizatJon after affinJty failure: Enabled
soft Preemprion: Disabled
F\J5t0T\ :
Tunnel has been up for: 0D:01:49 (since Thu Jul 16 22:21:22 UTC ZD20)
Current LSP :
  uptJme: 0D:D1:49 (sincs Thu Jul 16 2Z:21:2Z UTC 2D20)
  Re opt . L SP
    Last sa4 1ur e:
      LSP not si gna1l ed, dent ca1 t at he LCuRR ENTs LSP
      Dat e, Ti me: Thu 2 u1 16 18 :12 : 24 UTC 2020 L04 : 10 : 4? agoJ
    Pr 1 O r L SP :
      ID : 2 Pat h opt i on : 1
      Remo\ al vn1 g ger : pat h u eri IN c at i on fai l ed

Path nfo {OSPF 10 area D} :
Node hop count : 2
HoPO: 1D.1D.11.2

```

## Anexo D: Segment Routing

```

RP/0/0/CPU0:PE1#show mpls label table detail
Sun Jul 19 01:15:52.928 UTC
Table Label      Owner                               State Rewrite
-----
0 0 LSD(A)                               InUse Yes
0 1 LSD(A)                               InUse Yes
0 2 LSD(A)                               InUse Yes
0 13 LSD(A)                              InUse Yes
0 18000 OSPF(A):ospf-10                 InUse No
(Lbl-blk SRGB, vers:0, (start_label=18000, size=2000)
0 24000 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=1, intf=Gi0/0/0/0, nh=200.10.10.5)
0 24001 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/0, nh=200.10.10.5)
0 24002 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=1, intf=Gi0/0/0/1, nh=10.10.1.2)
0 24003 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/1, nh=10.10.1.2)
0 24004 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=1, intf=Gi0/0/0/3, nh=200.10.10.1)
0 24005 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/3, nh=200.10.10.1)
0 24006 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=1, intf=Gi0/0/0/5, nh=10.10.11.2)
0 24007 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/5, nh=10.10.11.2)
0 24008 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=1, intf=Gi0/0/0/2, nh=10.10.2.2)
0 24009 OSPF(A):ospf-10                 InUse Yes
(SR Adj Segment IPv4, vers:0, index=0, type=2, intf=Gi0/0/0/2, nh=10.10.2.2)
RP/0/0/CPU0:PE1#
RP/0/0/CPU0:PE1#
RP/0/0/CPU0:PE1#
RP/0/0/CPU0:PE1#

```

```

PE1#sh mpls traffic-eng tun
PE1#sh mpls traffic-eng tunnels

```

P2P TUNNELS/LSPs:

```

Name: PE1_t1                               (Tunnel) Destination: 7.7.7.7
Status:
  Admin: up      Oper: up | Path: not valid  Signalling: Down
  path option 10, (SEGMENT-ROUTING) type explicit IP_PATH1

Config Parameters:
  Bandwidth: 0 kbps (Global) Priority: 6 6 Affinity: 0x0/0xFFFF
  Metric Type: IGP (interface)
  Path Selection:
  Protection: any (default)
  Path-selection Tiebreaker:
  Global: not set Tunnel Specific: not set Effective: min-fill (default)
  Hop Limit: disabled
  Cost Limit: disabled
  Path-invalidation timeout: 10000 msec (default), Action: Tear
  AutoRoute: enabled LockDown: disabled Loadshare: 10 [0]
  auto-bw: disabled
  Fault-OAM: disabled, wrap-Protection: disabled, wrap-Capable: No

```

```

History:
  Tunnel:
    Time since created: 2 minutes, 48 seconds
    Number of LSP IDs (Tun_Instances) used: 0

```

P2MP TUNNELS:

```

P2MP SUB-LSPs:
PE1#
PE1#

```