**Test MPLS**

**Para transportar paquetes de datos a través de la red**

Los routers con MPLS utilizan un mecanismo basado en etiquetas para alcanzar la red destino

**Protocolos que puede soportar MPLS**

MPLS puede soportar diferentes protocolos layer 3 (IPv4, IPv6, etc) y diferentes protocolos layer 2 (Ethernet, ATM, Frame Relay, PPP, etc)

**Organizaciones que desarrollan los estándares y especificaciones MPLS**

Internet Engineering Task Force (IETF) es la organización que lidera el desarrollo de especificaciones y estándares relacionados con MPLS

**Empareja las preguntas y respuestas**

PE → Router frontera (Edge) del proveedor de servicios de Internet (ISP),

P → Router intermedio de la red del proveedor de servicios de Internet (ISP),

LSP → Camino (Path) MPLS basado en etiquetas entre un origen y un destino,

LSR → Router que implementa MPLS,

ELSR → Router frontera (Edge) que implementa MPLS,

LDP → Protocolo para la distribución de etiquetas

**En MPLS se pueden apilar etiquetas (label stacking) para permitir, por ejemplo, la creación de túneles VPN**

Para apilar etiquetas se utiliza el campo **S** de la cabecera MPLS. **S=0** indica que hay más etiquetas apiladas y **S=1** indica la última etiqueta MPLS del *stack*

**A continuación se muestra una captura de un paquete IP con las cabeceras Ethernet, MPLS e IPv4. Los datos están en hexadecimal.**

**Ethernet: DEST MAC ADDRESS | SOURCE MAC ADDRESS | TYPE**

**MPLS: LABEL | TC | S | TTL**

**IP: VER | ... | SOURCE IP | DEST IP**

**ETHERNET MPLS IPV4**

**0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (BYTE)**

**34 64 A9 48 3B E8 48 88 35 16 22 5C 88 47 36 2B**

**01 40 45 00 04 04 61 46 00 00 40 01 E3 A9 98 01**

**02 01 98 00 00 07 ... Contenido datagrama IPv4**

En el paquete capturado el campo **TYPE** a nivel 2 indica la cabecera MPLS a continuación de Ethernet: **TYPE=0X8847**

**A continuación se muestra una captura de un paquete IP con las cabeceras Ethernet, MPLS e IPv4. Los datos están en hexadecimal.**

**Ethernet: DEST MAC ADDRESS | SOURCE MAC ADDRESS | TYPE**

**MPLS: LABEL | TC | S | TTL**

**IP: VER | ... | SOURCE IP | DEST IP**

**ETHERNET MPLS IPV4**

**0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (BYTE)**

**34 64 A9 48 3B E8 48 88 35 16 22 5C 88 47 36 2B**

**01 40 45 00 04 04 61 46 00 00 40 01 E3 A9 98 01**

**02 01 98 00 00 07 ... Contenido datagrama IPv4**

En el paquete de datos capturado la etiqueta (**label**) MPLS toma el valor 0x362B0 = 221872

**A continuación se muestra una captura de un paquete IP con las cabeceras Ethernet, MPLS e IPv4. Los datos están en hexadecimal.**

**Ethernet: DEST MAC ADDRESS | SOURCE MAC ADDRESS | TYPE**

**MPLS: LABEL | TC | S | TTL**

**IP: VER | ... | SOURCE IP | DEST IP**

**ETHERNET MPLS IPV4**

**0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (BYTE)**

**34 64 A9 48 3B E8 48 88 35 16 22 5C 88 47 36 2B**

**01 40 45 00 04 04 61 46 00 00 40 01 E3 A9 98 01**

**02 01 98 00 00 07 ... Contenido datagrama IPv4**

En el paquete de datos capturado el campo *Botton of label Stack* toma el valor **S=1** lo que indica que es la última etiqueta y a continuación se encapsula un datagrama IPv4

**A continuación se muestra una captura de un paquete IP con las cabeceras Ethernet, MPLS e IPv4. Los datos están en hexadecimal.**

**Ethernet: DEST MAC ADDRESS | SOURCE MAC ADDRESS | TYPE**

**MPLS: LABEL | TC | S | TTL**

**IP: VER | ... | SOURCE IP | DEST IP**

**ETHERNET MPLS IPV4**

**0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (BYTE)**

**34 64 A9 48 3B E8 48 88 35 16 22 5C 88 47 36 2B**

**01 40 45 00 04 04 61 46 00 00 40 01 E3 A9 98 01**

**02 01 98 00 00 07 ... Contenido datagrama IPv4**

El campo TTL en MPLS es 0x40=64

**A continuación se muestra una captura de un paquete IP con las cabeceras Ethernet, MPLS e IPv4. Los datos están en hexadecimal.**

**Ethernet: DEST MAC ADDRESS | SOURCE MAC ADDRESS | TYPE**

**MPLS: LABEL | TC | S | TTL**

**IP: VER | ... | SOURCE IP | DEST IP**

**ETHERNET MPLS IPV4**

**0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 (BYTE)**

**34 64 A9 48 3B E8 48 88 35 16 22 5C 88 47 36 2B**

**01 40 45 00 04 04 61 46 00 00 40 01 E3 A9 98 01**

**02 01 98 00 00 07 ... Contenido datagrama IPv4**

El campo **TTL** en la cabecera del paquete**IPv4**vale 0x40=64

**En el circuito de la figura los routers asignan las etiquetas que se indican a la red 10.0.0.0/24, que previamente fue difundida por OSPF. Luego, difunden las etiquetas a los routers vecinos mediante LDP para construir el path LSP. Si un equipo de la red 10.0.0.0/24 envía un paquete de datos a 192.168.0.10, ¿qué etiquetas MPLS se añaden a lo largo del path LSP?**

Diagrama

El contenido generado por IA puede ser incorrecto.

Red 10.0.0.0/24 ->R1: Paquete IP (no label MPLS); R1->R2: Label=87; R2->R3: Label=11; R3->R4: Label=65; R4->R5: Label=23; R5->Red 192.168.0.0/24: Paquete IP (no label MPLS)

**Penultimate Hop Popping (PHP)**

Con PHP el último router ELSR en el LSP advierte al penúltimo router LSR en el LSP para eliminar (pop) la etiqueta MPLS antes de enviar el paquete al último ELSR con destino a una red final. Esto descarga al ELSR de hacer dos *lookup* para eliminar la etiqueta final y reenviar el paquete IP al destino

**Con MPLS Layer 3 VPN se pretende proporcionar conectividad *peer-to-peer* entre redes privadas de clientes (customer C-network) a través de redes públicas compartidas de proveedores (provider P-network). Utilizando MPLS Layer 3 VPN:**

Dos clientes diferentes (Customer A y Customer B) pueden estar usando el mismo espacio de direcciones IP privadas

**Para comunicar redes privadas de cliente con MPLS Layer 3 VPN se requiere apilar etiquetas (*label stack*) para enviar tráfico a través del dominio MPLS:**

Son necesarias dos etiquetas: Etiqueta para identificar la VPN y Etiquetas LDP para transportar el tráfico a través del dominio MPLS

**En la red de la figura se indican las etiquetas VPN Label Switched Path para enviar un paquete IP de Customer B: Site1 a Customer B: Site 2. El contenido de las etiquetas a lo largo del LSP es el siguiente (con formato [MAC Header | Label\_1 | Label\_2| IP packet]):**

Texto

El contenido generado por IA puede ser incorrecto.

**(2 respuestas)**

CE\_RA->PE\_R1: [MAC Header | IP packet]; PE\_R1->P\_R2: [MAC Header | 16 | 48 | IP packet]; P\_R2->P\_R3: [MAC Header | 25 | 48 | IP packet]; P\_R3->P\_R4: [MAC Header | 87 | 48 | IP packet]; P\_R4->PE\_R5: [MAC Header | 48 | IP packet]; PE\_R5 -> CE\_RB: P\_R2->P\_R3: [MAC Header | IP packet],

CE\_RA->PE\_R1: [MAC Header | IP packet]; PE\_R1->P\_R2: [MAC Header | 48 | 16 | IP packet]; P\_R2->P\_R3: [MAC Header | 48 | 25 | IP packet]; P\_R3->P\_R4: [MAC Header | 48 | 87 | IP packet]; P\_R4->PE\_R5: [MAC Header | 48 | IP packet]; PE\_R5 -> CE\_RB: P\_R2->P\_R3: [MAC Header | IP packet]

**Ingeniería de tráfico: MPLS soporta la creación de rutas diferentes entre un origen y un destino en un *backbone*de routers en Internet**

Verdadero