

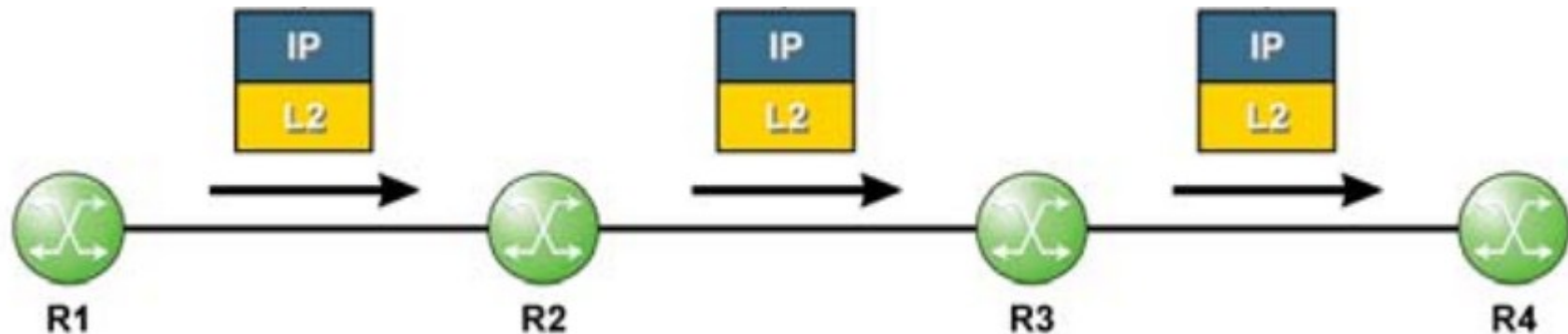
# **MPLS**

# **Multiprotocol Label Switching**

# MPLS - Introducción (1)

- En redes IP, la información se envía en paquetes de datos, llamados datagramas.
- Dichos datagramas se envían de extremo a extremo y la decisión acerca del camino a elegir se define en cada salto.
- Los saltos, en el mundo IP, son los routers.
- Las decisiones de caminos se realizan en base a la información contenida en la cabecera del datagrama, concretamente, analizando la IP de destino.

# MPLS - Introducción (2)



- En cada salto, se analiza y se remueve la encapsulación L2.
- Cada paquete, en cada router que atraviesa, es sometido a un análisis donde se compara su dirección IP de destino con cada entrada de la tabla de ruteo (longest prefix match);
- Ubicada la red que más se ajusta a dicha dirección, se elige luego la interfaz de salida.
  - **La tabla de ruteo se construye en base a protocolos de enrutamiento**, ya sean estáticos o dinámicos.
- Se construye una nueva capa L2 y se conmuta el datagrama a la interfaz de salida correspondiente.

# MPLS - Introducción (3)

- **La red IP es una red de conmutación de paquetes no orientada a la conexión.**
  - No se pre-establece un camino extremo a extremo [1].
  - No es posible una reserva de recursos *a priori*.

# MPLS - Introducción (4)

- En redes de conmutación de paquetes, pero orientados a la conexión, se utilizan circuitos virtuales.
  - En ATM, por ejemplo, se pueden negociar recursos antes de enviar tráfico y se establecen VPI/VCI como identificadores, los cuales son utilizados para conmutar el mismo dentro de la red.
- Cuando hay circuitos virtuales, no se consumen recursos para elegir qué ruta se ha de utilizar en cada salto, **ya que el camino ha sido previamente establecido**, desde el origen al destino.
- Cada paquete lleva un identificador que indica a qué circuito virtual pertenece [2].

# MPLS - Introducción (5)

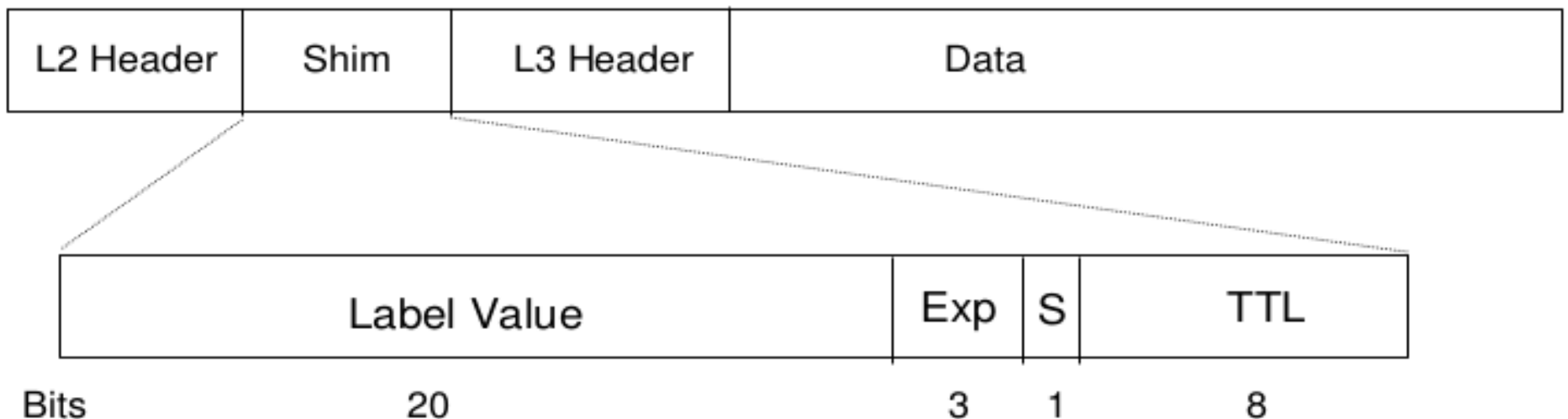
- MPLS introduce el paradigma de *orientado-a-la-conexión* a las redes de datagramas.
- Para ello agrega una etiqueta en frente de cada datagrama IP.
- El reenvío se hace ahora conmutando en base a esta etiqueta en lugar de analizar la dirección IP de destino.
- La etiqueta es ubicada en un índice -que cada router posee- lo cual permite muy velozmente encontrar la interfaz de salida [3].

# MPLS - Introducción (6)

- **MPLS se describe en la RFC 3031.**
- El término 'Multiprotocolo' refiere al hecho de que se puede transportar cualquier tipo de carga, ya sea IP, ATM, FR, Ethernet, etc.
- 'Label Switching' indica que, en lugar de conmutar paquetes en base a la información contenida en la cabecera IP, se conmutan las tramas en base a etiquetas predefinidas.

# MPLS – Stack (1)

- El encabezado de MPLS tiene una longitud de 4 bytes (32 bits).
- Se destacan la etiqueta, que identifica al circuito virtual, el campo QoS (Exp), para proporcionar calidad de servicio, el campo S, que permite saber si tenemos etiquetas apiladas y el campo TTL, tiempo de vida.



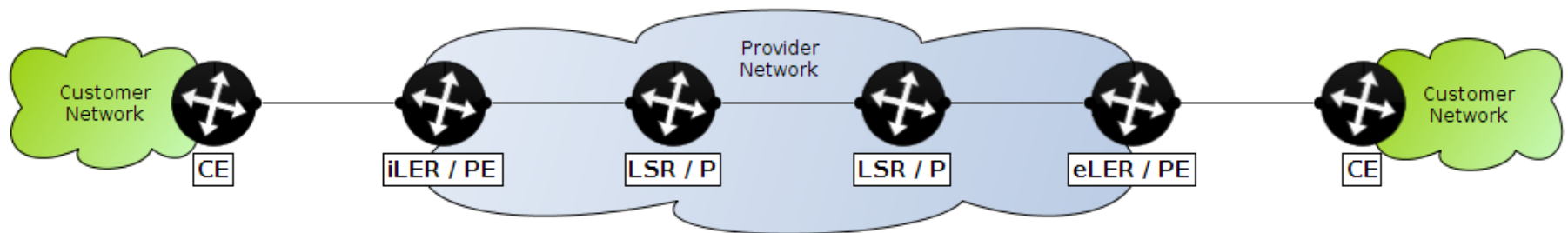


# MPLS – Terminología (1)

- Un dominio MPLS está compuesto por diversas entidades, cada una de ellas con diferentes funciones:
  - iLER: Ingress Label Edge Router
  - eLER: Egress Label Edge Router
  - LSR: Label Switching Router
  - PE: Provider Edge Router
  - CE: Customer Edge Router
  - P: Provider Core Router

## MPLS – Terminología (2)

- No obstante, existen similitudes entre las definiciones anteriores:



- Así:
  - Un router PE es equivalente a un LER
  - Un router P es equivalente a un LSR

## MPLS – Terminología (3)

- **Customer Edge Router:** El CE reside en las premisas del cliente. Provee acceso a la red del proveedor por medio de enlaces conectados directamente con el PE. El cliente desconoce de qué manera el proveedor de servicios transportará sus datos.
- **Provider Edge Router:** El PE posee al menos una interfaz directamente conectada contra el CE (pueden ser varios CEs). Además es el equipo que conecta contra uno o más routers P. El router PE es el encargado de encapsular el tráfico recibido y de permitir el ingreso al dominio MPLS.

# MPLS – Terminología (4)

- **Provider Core Router:** El router P se ubica en el backbone de la red del proveedor. Es el encargado de hacer la conmutación en base a etiquetas. Debe estar dimensionado para soportar los requerimientos necesarios.
  - **No sabe qué hay en capas superiores!**

# MPLS – Terminología (5)

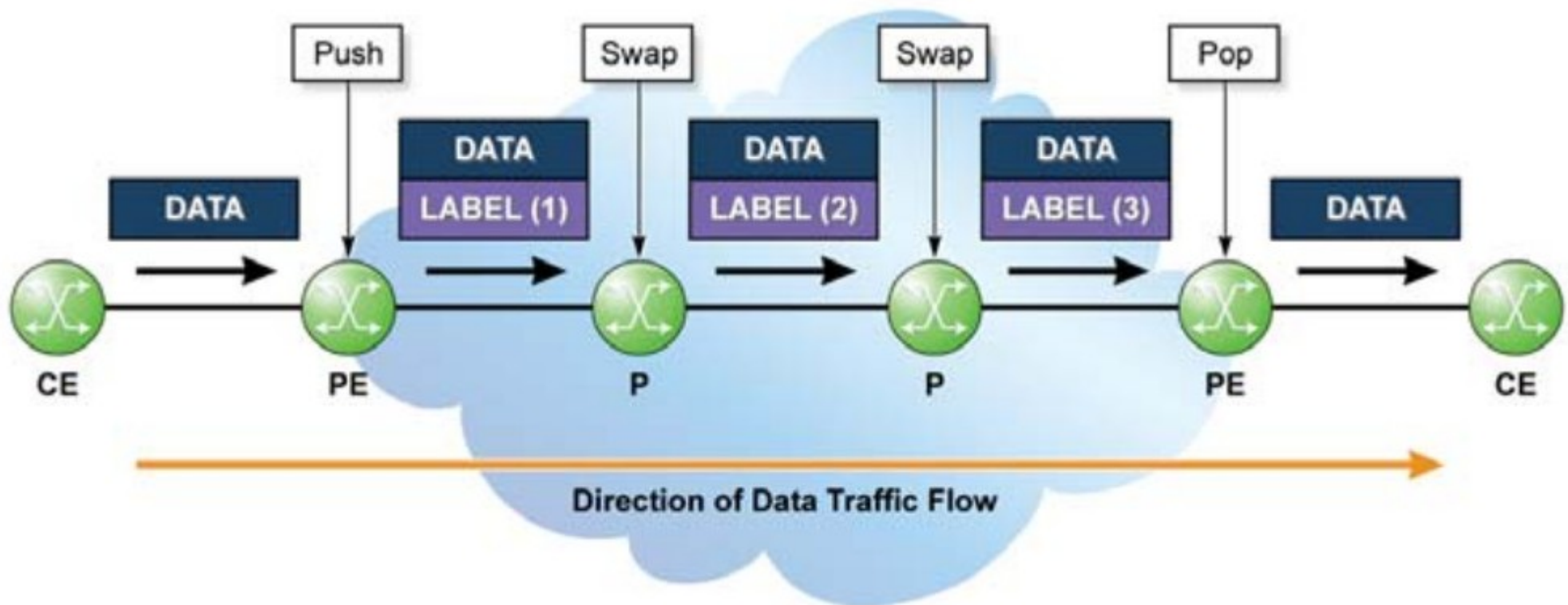
- **Label Edge Router [4]:** El router LER se ubica en la frontera entre el dominio MPLS y el cliente. Es similar al PE. Un LER puede ser:
  - Ingress LER: en este caso, todo el tráfico que viene del cliente es etiquetado apropiadamente y enviado al dominio MPLS.
  - Egress LER: recibe tráfico MPLS y elimina las etiquetas antes de reenviar el tráfico al CE.
- **Label Switching Router [4]:** El LSR reside dentro del dominio MPLS. Se encarga de llevar el tráfico desde un iLER hasta un eLER. Cuando un LSR recibe tráfico etiquetado, en función de la etiqueta de entrada, la reemplaza por una de salida y reenvía el paquete con la nueva etiqueta al próximo router. Es similar al router P.

# MPLS – Conmutación (1)

- Una vez dentro del dominio MPLS los paquetes son conmutados en base a la etiqueta que llevan.
- El router iLER/PE, en función de la IP de destino del paquete recibido desde el CE, inserta (**push**) una etiqueta MPLS y lo envía al siguiente salto.
- El siguiente salto (router P/LSR) analiza la etiqueta del paquete entrante, busca la etiqueta de salida correspondiente, reemplaza (**swap**) la original y envía el paquete con la etiqueta nueva al siguiente salto.
  - El análisis es realizado en la LFIB (Label Forwarding Information Base)
- Cuando el paquete llega al eLER, éste remueve (**pop**) la etiqueta y envía el paquete IP al destino deseado.

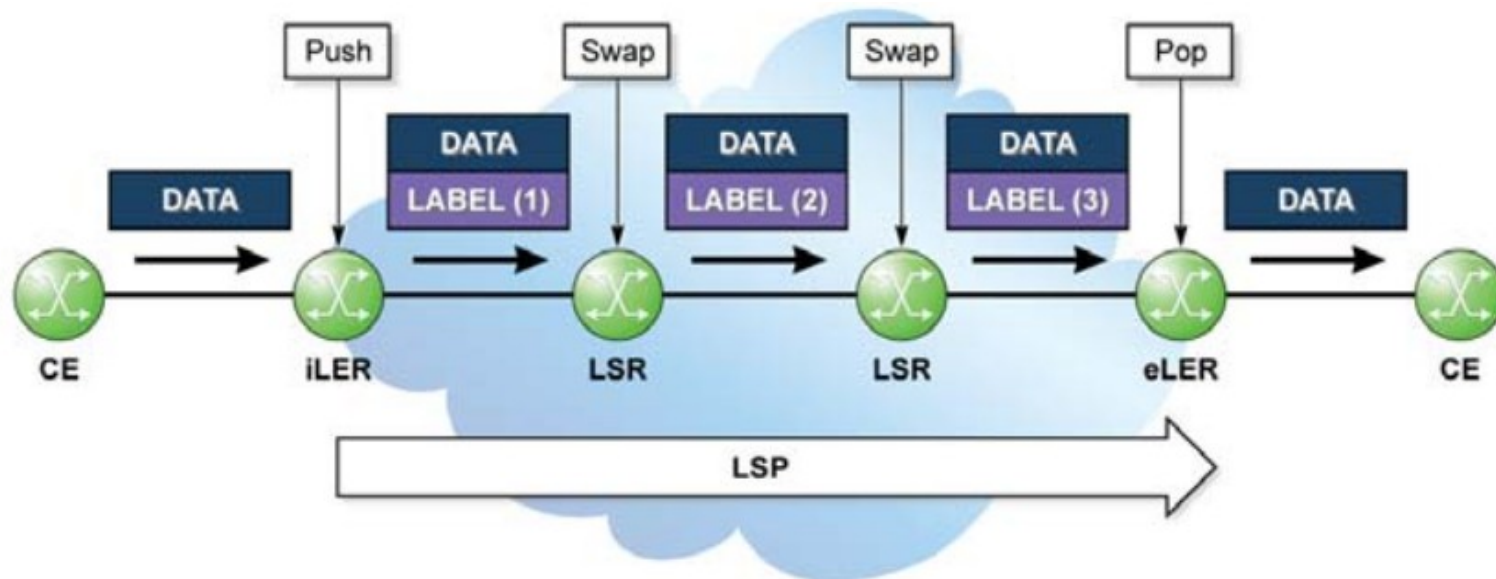
## MPLS – Conmutación (2)

- Entonces, las etiquetas pueden ser insertadas (push), reemplazadas (swap) o removidas (pop) **dependiendo de la ubicación de la red donde se encuentre el paquete y en función de lo que defina cada Label Forwarding Database.**



# MPLS – Conmutación (3)

- El camino que sigue un paquete dentro del dominio MPLS, en base a la conmutación de etiquetas, se denomina Label Switched Path (LSP), el cual empieza en un iLER y termina en un eLER [6].
- Es una conexión lógica que representa el camino entre dos LERs.
  - Se lo conoce también como túnel de transporte.
  - Los túneles son unidireccionales: hay uno de ida y otro de vuelta, lo que implica que las funciones eLER e iLER son alternantes.





# MPLS – Conmutación (4)

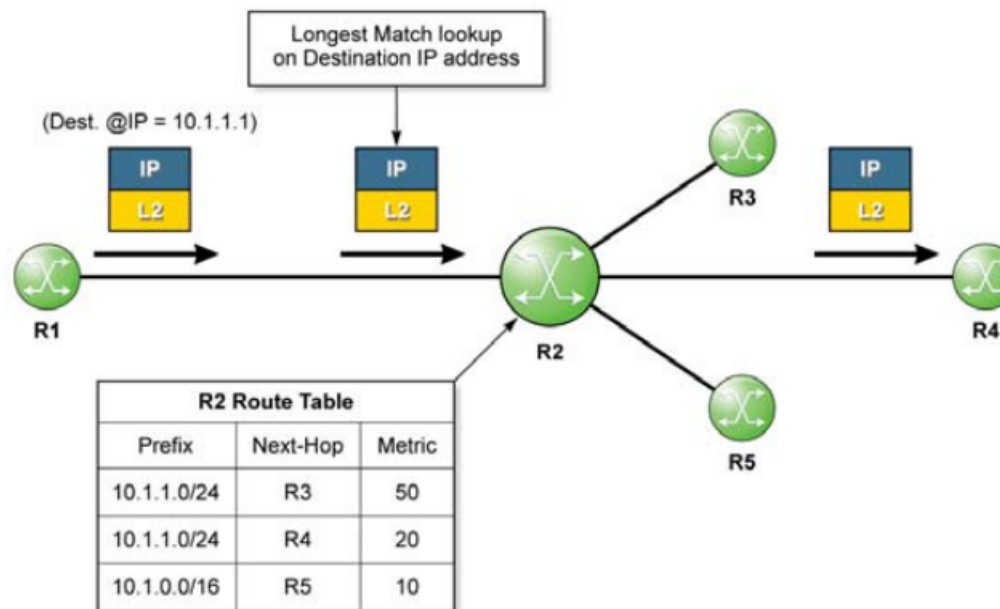
- Los túneles de transporte son establecidos **antes** de la llegada de los paquetes de usuario.
- Definen la serie de etiquetas que se utilizarán entre un iLER y un eLER.
- Por eso hablamos de cierta comparación entre MPLS y Circuitos Virtuales: el camino debe existir *a priori*.
  - Hay protocolos que se encargan de señalizarlos. Los protocolos más usados son **LDP – Label Distribution Protocol** y **RSVP – Resource Resevation Protocol**.

# MPLS – Conmutación (5)

- Una **Forwarding Equivalence Class (FEC)** es un grupo de paquetes IP que son conmutados de la misma manera y bajo el mismo tratamiento [5][7][8][10].
  - En una red IP clásica, los paquetes son clasificados/analizados en cada salto, de acuerdo a su dirección IP de destino.
    - Se hace cada vez que el paquete entra a un router distinto.
  - En una red MPLS, los paquetes pueden ser clasificados dentro de una FEC en base –también– a direcciones IP de destino u otros parámetros.
    - Se hace una sola vez: cuando el paquete llega al iLER.

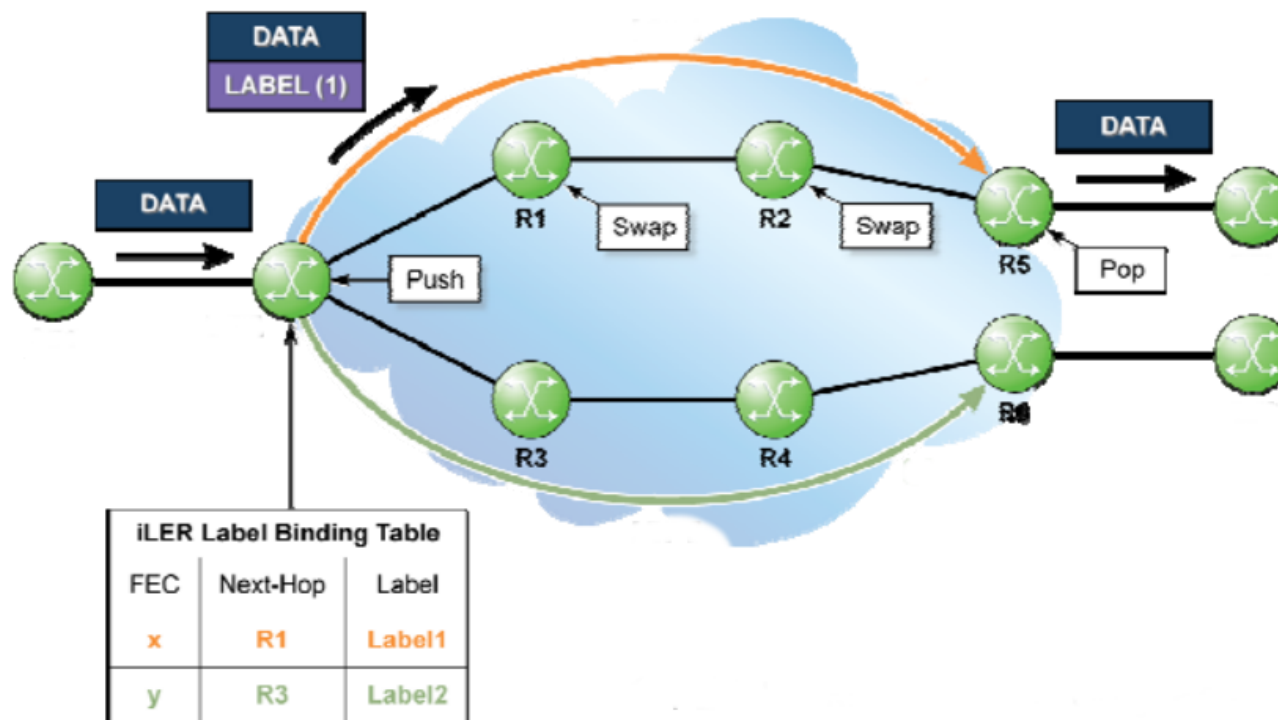
# MPLS – Conmutación (6)

- En redes IP clásicas, las FECs se relacionan con entradas correspondientes a la tabla de routing.
  - La tabla de un router tiene una ruta 10.1.1.0/24 con next-hop R4.
  - Dos paquetes que llegan a R2 con IP de destino 10.1.1.1 y 10.1.1.2: pertenecen a la misma FEC → vía el next-hop R4.
  - La búsqueda del FEC se hace en cada salto.



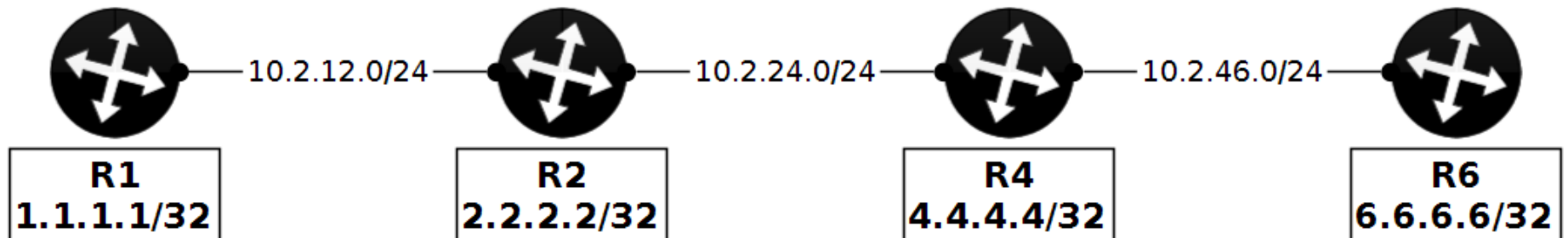
# MPLS – Conmutación (7)

- En redes MPLS, la FEC se busca sólo en los iLER.
- Encontrar una FEC implica encontrar qué etiqueta MPLS insertar (push) en el paquete.
- Insertar una etiqueta en un paquete implica encontrar un LSP: **el túnel LSP debe existir con anterioridad.**



# MPLS – Conmutación (8)

Se realizará un envío de tráfico ICMP desde 6.6.6.6/32 a 1.1.1.1/32.

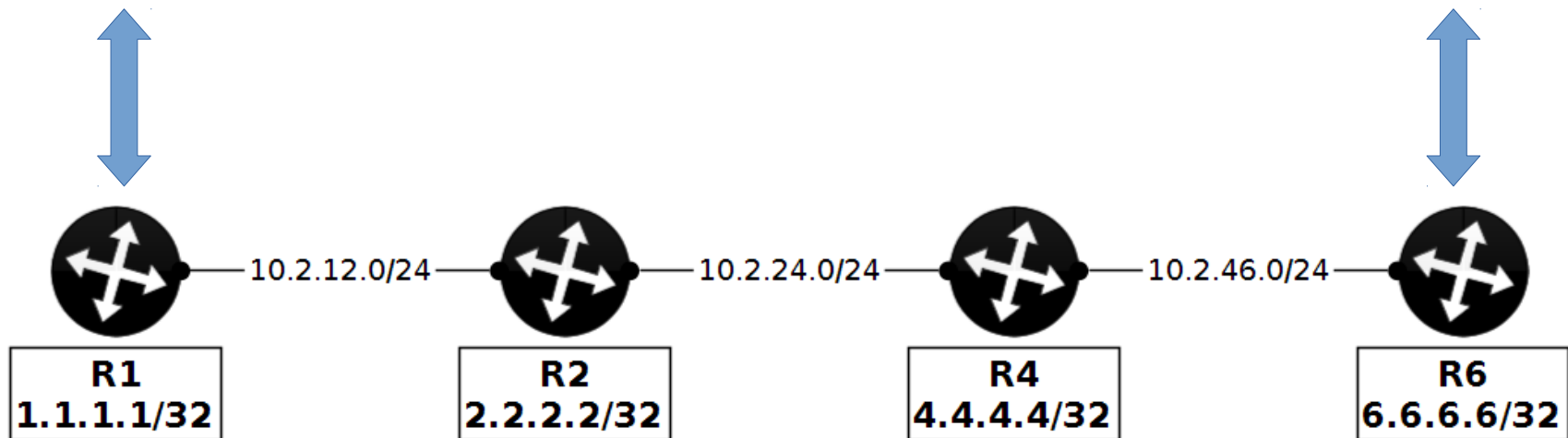


# MPLS – Conmutación (8)

- Se aprecia aquí el contenido de las LIBs de los routers R1 y R6.
- En cada caso existen entradas POP, SWAP y PUSH, de acuerdo al destino.

LDP Prefix Bindings (Active)					
Prefix	Op	IngLbl	EgrLbl	EgrIntf/LspId	EgrNextHop
1.1.1.1/32	Pop	262143	--	--	--
2.2.2.2/32	Push	--	131071	1/1/1	10.12.0.2
4.4.4.4/32	Push	--	131069	1/1/1	10.12.0.2
4.4.4.4/32	Swap	262141	131069	1/1/1	10.12.0.2
6.6.6.6/32	Push	--	131068	1/1/1	10.12.0.2
6.6.6.6/32	Swap	262140	131068	1/1/1	10.12.0.2
No. of Prefix Active Bindings: 6					

LDP Prefix Bindings (Active)					
Prefix	Op	IngLbl	EgrLbl	EgrIntf/LspId	EgrNextHop
1.1.1.1/32	Push	--	131069	1/1/1	10.46.0.4
1.1.1.1/32	Swap	262140	131069	1/1/1	10.46.0.4
2.2.2.2/32	Push	--	131070	1/1/1	10.46.0.4
2.2.2.2/32	Swap	262141	131070	1/1/1	10.46.0.4
4.4.4.4/32	Push	--	131071	1/1/1	10.46.0.4
6.6.6.6/32	Pop	262143	--	--	--
No. of Prefix Active Bindings: 6					



# MPLS – Conmutación (8)

- Análisis Wireshark en R6. El tráfico es etiquetado de acuerdo a lo que dice la LIB.

LDP Prefix Bindings (Active)					
Prefix	Op	IngLbl	EgrLbl	EgrIntf/LspId	EgrNextHop
1.1.1.1/32	Push	--	131069	1/1/1	10.46.0.4
1.1.1.1/32	Swap	262140	131069	1/1/1	10.46.0.4
2.2.2.2/32	Push	--	131070	1/1/1	10.46.0.4
2.2.2.2/32	Swap	262141	131070	1/1/1	10.46.0.4
4.4.4.4/32	Push	--	131071	1/1/1	10.46.0.4
6.6.6.6/32	Pop	262143	--	--	--
No. of Prefix Active Bindings: 6					

```

→ 50 11:55:28.125335 6.6.6.6 1.1.1.1 Echo (ping) request 10
← 51 11:55:28.127942 1.1.1.1 6.6.6.6 Echo (ping) reply 10

Frame 50: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on in
Ethernet II, Src: 00:b7:cc:28:08:01 (00:b7:cc:28:08:01), Dst: 00:b7:cc:4b:e
MultiProtocol Label Switching Header, Label: 131069, Exp: 7, S: 1, TTL: 64
0001 1111 1111 1111 1101 .... = MPLS Label: 131069
.... 111. .... = MPLS Experimental Bits: 7
.... 1 .... = MPLS Bottom Of Label Stack: 1
.... 0100 0000 = MPLS TTL: 64
Internet Protocol Version 4, Src: 6.6.6.6, Dst: 1.1.1.1
Internet Control Message Protocol

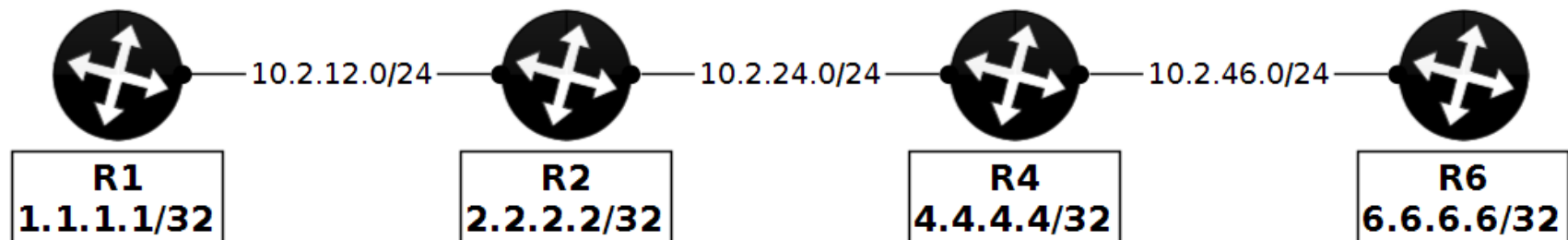
```

```

→ 50 11:55:28.125335 6.6.6.6 1.1.1.1 Echo (ping) request 10
← 51 11:55:28.127942 1.1.1.1 6.6.6.6 Echo (ping) reply 10

Frame 51: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on in
Ethernet II, Src: 00:b7:cc:4b:ea:03 (00:b7:cc:4b:ea:03), Dst: 00:b7:cc:28:0
MultiProtocol Label Switching Header, Label: 262143, Exp: 7, S: 1, TTL: 62
0011 1111 1111 1111 1111 .... = MPLS Label: 262143
.... 111. .... = MPLS Experimental Bits: 7
.... 1 .... = MPLS Bottom Of Label Stack: 1
.... 0011 1110 = MPLS TTL: 62
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 6.6.6.6
Internet Control Message Protocol

```



# MPLS – Conmutación (8)

- Análisis Wireshark en R1. El tráfico es etiquetado de acuerdo a lo que dice la LIB.

Prefix	Op	IngLbl	EgrLbl	EgrIntf/LspId	EgrNextHop
1.1.1.1/32	Pop	262143	--	--	--
2.2.2.2/32	Push	--	131071	1/1/1	10.12.0.2
4.4.4.4/32	Push	--	131069	1/1/1	10.12.0.2
4.4.4.4/32	Swap	262141	131069	1/1/1	10.12.0.2
6.6.6.6/32	Push	--	131068	1/1/1	10.12.0.2
6.6.6.6/32	Swap	262140	131068	1/1/1	10.12.0.2

No. of Prefix Active Bindings: 6

```

→ 28 12:04:43.480986 6.6.6.6 1.1.1.1 Echo (ping) request id
← 29 12:04:43.481755 1.1.1.1 6.6.6.6 Echo (ping) reply id

▶ Frame 28: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface
▶ Ethernet II, Src: 00:be:a8:31:41:02 (00:be:a8:31:41:02), Dst: 00:b7:cc:35:67:01
▼ MultiProtocol Label Switching Header, Label: 262143, Exp: 7, S: 1, TTL: 62
  0011 1111 1111 1111 1111 .... = MPLS Label: 262143
  .... 111. .... = MPLS Experimental Bits: 7
  .... 1 .... = MPLS Bottom Of Label Stack: 1
  .... 0011 1110 = MPLS TTL: 62
▶ Internet Protocol Version 4, Src: 6.6.6.6, Dst: 1.1.1.1
▶ Internet Control Message Protocol

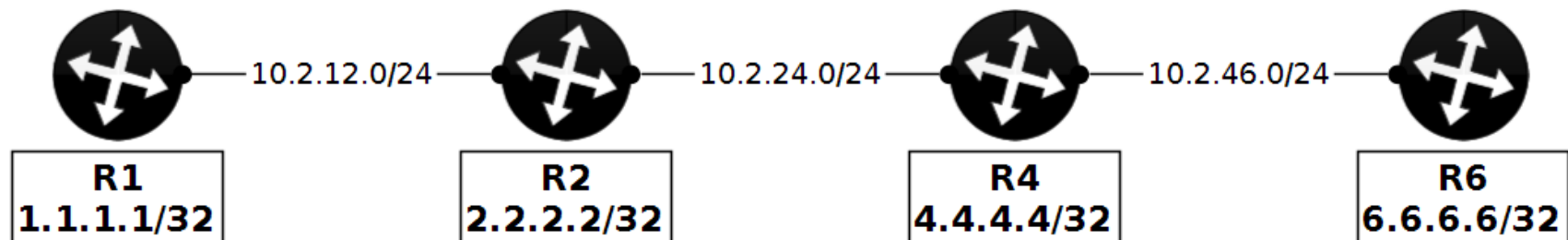
```

```

→ 28 12:04:43.480986 6.6.6.6 1.1.1.1 Echo (ping) request id
← 29 12:04:43.481755 1.1.1.1 6.6.6.6 Echo (ping) reply id

▶ Frame 29: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface
▶ Ethernet II, Src: 00:b7:cc:35:67:01 (00:b7:cc:35:67:01), Dst: 00:be:a8:31:41:02
▼ MultiProtocol Label Switching Header, Label: 131068, Exp: 7, S: 1, TTL: 64
  0001 1111 1111 1111 1100 .... = MPLS Label: 131068
  .... 111. .... = MPLS Experimental Bits: 7
  .... 1 .... = MPLS Bottom Of Label Stack: 1
  .... 0100 0000 = MPLS TTL: 64
▶ Internet Protocol Version 4, Src: 1.1.1.1, Dst: 6.6.6.6
▶ Internet Control Message Protocol

```





# MPLS – Labels (1)

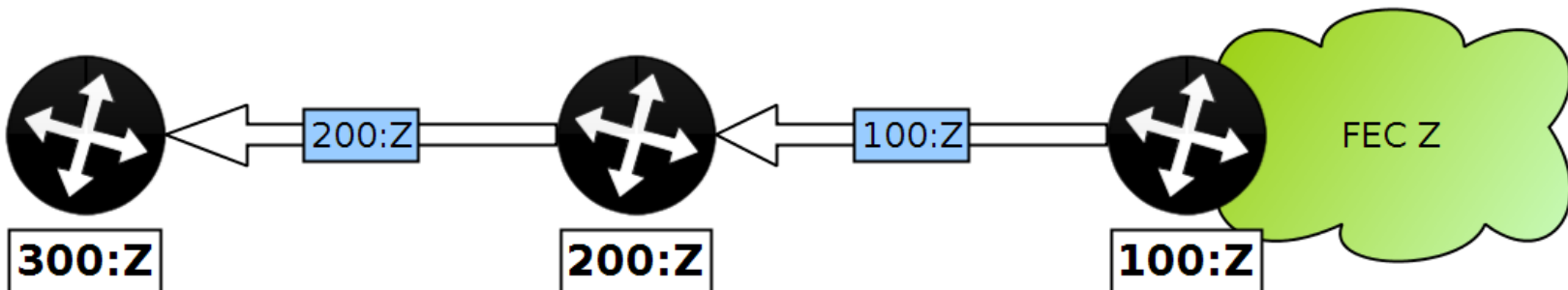
- Los caminos (LSP) deben existir con anterioridad.
- Los caminos están definidos en base a conjuntos de etiquetas:
  - Las etiquetas en cada paquete se van reemplazando conforme se atraviesan los diferentes routers LSR/P.
- Las etiquetas las elige cada LSR y asigna una por cada FEC.
- Una vez que un LSR ha decidido utilizar una etiqueta para designar un FEC, se dice que se ha creado un 'binding' entre la etiqueta y el FEC.
  - **Lo que el LSR informa a sus vecinos es en realidad el 'binding' creado [9].**

# MPLS – Labels (2)

- RFC3031 (Arquitectura de MPLS) no define una forma de comunicar 'bindings' [11].
- Hay varios protocolos que realizan este trabajo. Uno de ellos es LDP (RFC5036) y su forma más común de trabajar es enviando 'bindings' a sus vecinos conforme se asignan etiquetas localmente a los diversos FECs.
- Es necesario que los protocolos de routing hayan convergido antes de poder definir los caminos.
  - Los protocolos convergen cuando sólo existen en las tablas de routing las mejores rutas.
- Esto es así porque las FECs están en las tablas de routing.
  - Cada destino en una tabla de routing es potencialmente un FEC distinto para el que existirá un LSP.

# MPLS – Labels (3)

- Procedimiento para LDP:
  - Formar vecindad con vecinos
  - Elegir etiqueta 'L' para  $FEC_i$
  - Crear asociación (binding) entre  $FEC_i$  y 'L'  $\rightarrow L:FEC_i$
  - Informar vecino acerca del binding creado
  - El vecino, cada vez que quiera enviar tráfico con destino  $FEC_i$ , usará la etiqueta 'L'.



## MPLS – Labels (4)

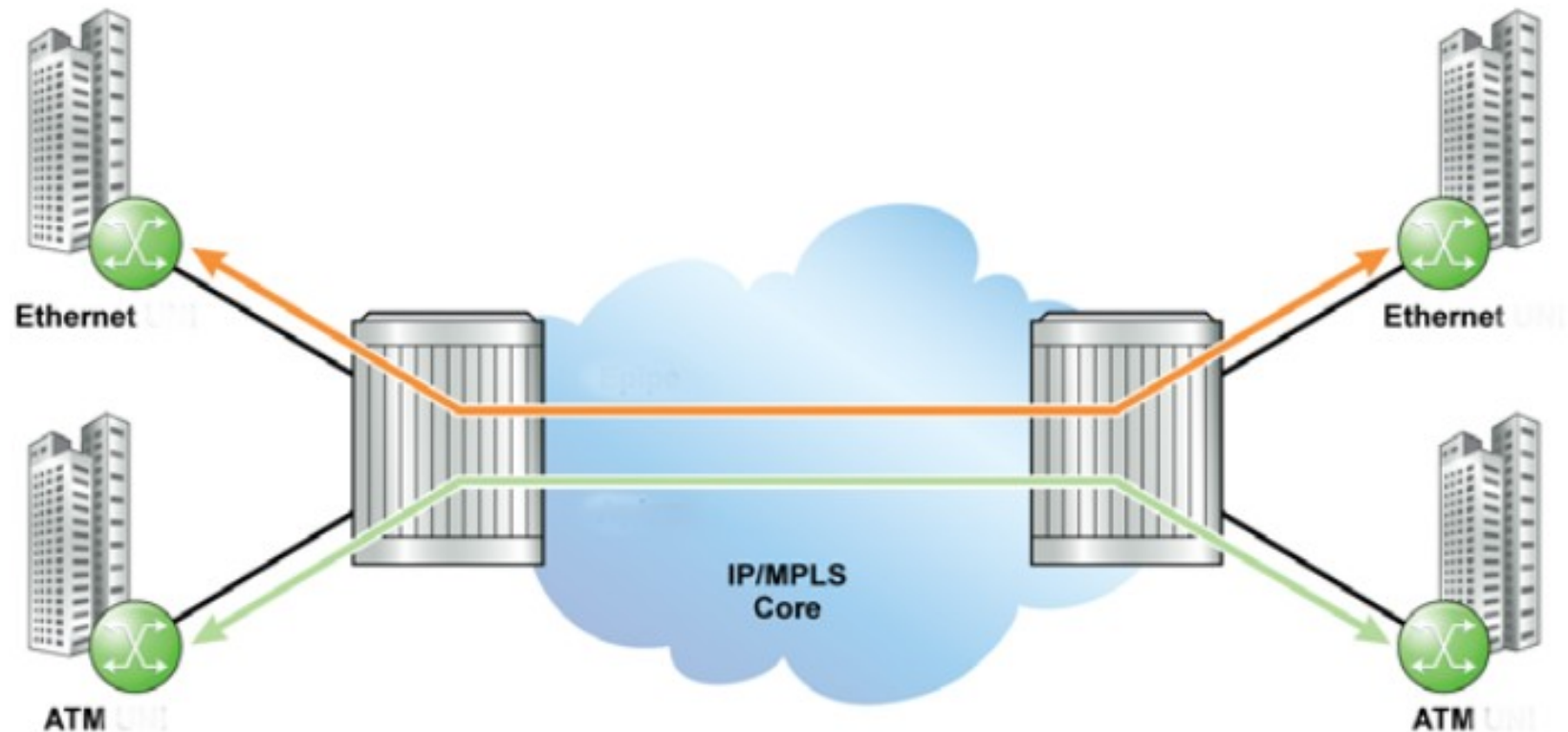
- Un LSR que recibe un 'binding' de un vecino podrá utilizarlo sólo si dicho FEC existe en su propia tabla de routing [12].
- En una red mallada, los routers pueden recibir 'bindings' de diferentes vecinos para un mismo destino.
- En este caso, el protocolo de distribución de etiquetas (LDP) **hace caso estricto a la estructura de la tabla de routing.**

# MPLS – Notas finales (1)

- MPLS mejora notablemente el envío de datos debido a varios aspectos:
  - Permite un comportamiento orientado a conexión en redes IP, facilitando el rendimiento en el envío de tráfico de tiempo real;
  - En el plano de datos, no se hace análisis de capa 3 para conmutarlo, por lo tanto no hay que subir en la pila de protocolos para saber qué hacer con el mismo (swap de etiquetas);
  - Permite la creación de múltiples túneles en función de FECs determinados (recordemos que podemos crear FECs también usando otros criterios, como el como el campo TOS);

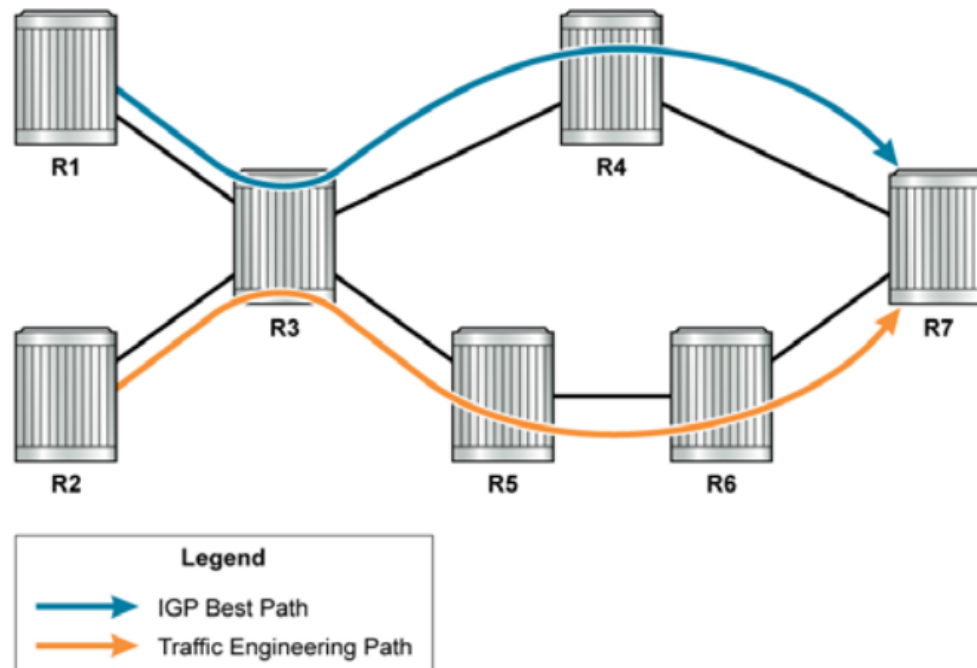
## MPLS – Notas finales (2)

- Al utilizar doble encapsulado se pueden crear VPNs dentro del domino MPLS, permitiendo que localidades remotas se comporten como si estuviesen en una LAN local, directamente conectadas.



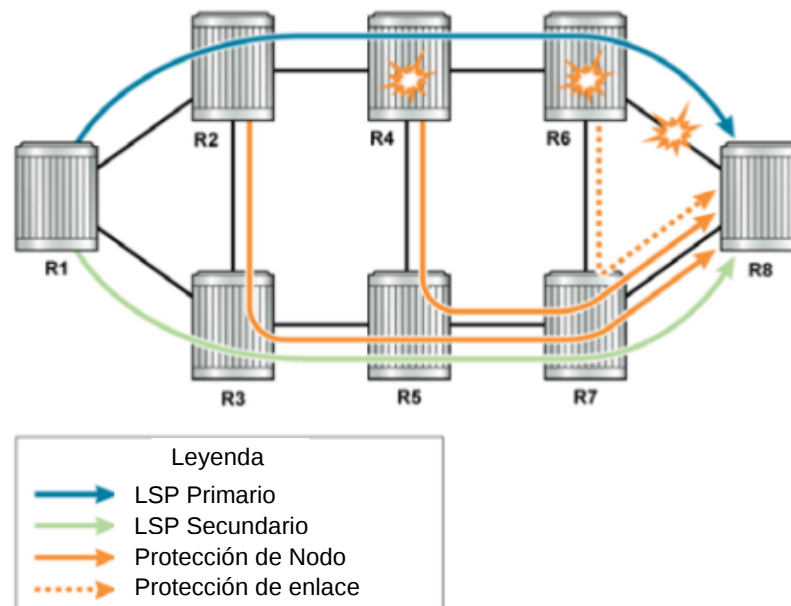
# MPLS – Notas finales (3)

- Con LDP, los LSPs siguen el comportamiento del protocolo de enrutamiento utilizado.
- Utilizando en cambio RSVP, se pueden aplicar técnicas de ingeniería de tráfico para, por ejemplo, generar caminos redundantes en caso de congestión en la red.



# MPLS – Notas finales (4)

- En la misma línea de la ingeniería de tráfico (usando RSVP), se pueden crear caminos protegidos, los cuales responden en tiempos menores a 50ms (dependiendo el caso) en caso de falla del camino activo.
- Esta característica se llama RSVP-TE FastReroute





# Referencias

- [1] – Redes de computadoras, Tanenbaum, 5ta ed, secc. 5.1.3
- [2] – Redes de computadoras, Tanenbaum, 5ta ed, secc. 5.1.4
- [3] – Redes de computadoras, Tanenbaum, 5ta ed, secc. 5.6.5
- [4] – RFC 3031. secc 2.2
- [5] – RFC 3031. secc 2.1
- [6] – RFC 3031. secc 3.15
- [7] – RFC 3031. secc 3.27.3
- [8] – RFC 3031. secc 3.1
- [9] – RFC 3031. secc 3.4
- [10] – RFC 5036. secc 2.1
- [11] – RFC 3031. secc 3.6
- [12] – RFC 5056. secc 3.5.7.1