

Xarxes de Computadors II

Tema 4. Multiprotocol Label Switching (MPLS)

Davide Careglio

Temario

- ▶ Tema 0. Repaso
- ▶ Tema 1. Arquitectura y direccionamiento en Internet
- ▶ Tema 2. Direccionamiento IPv6
- ▶ Tema 3. Encaminamiento intra-dominio
- ▶ **Tema 4. Multiprotocol Label Switching**
- ▶ Tema 5. Encaminamiento inter-dominio
- ▶ Tema 6. Conceptos avanzados

4. MultiProtocol Label Switching (MPLS)

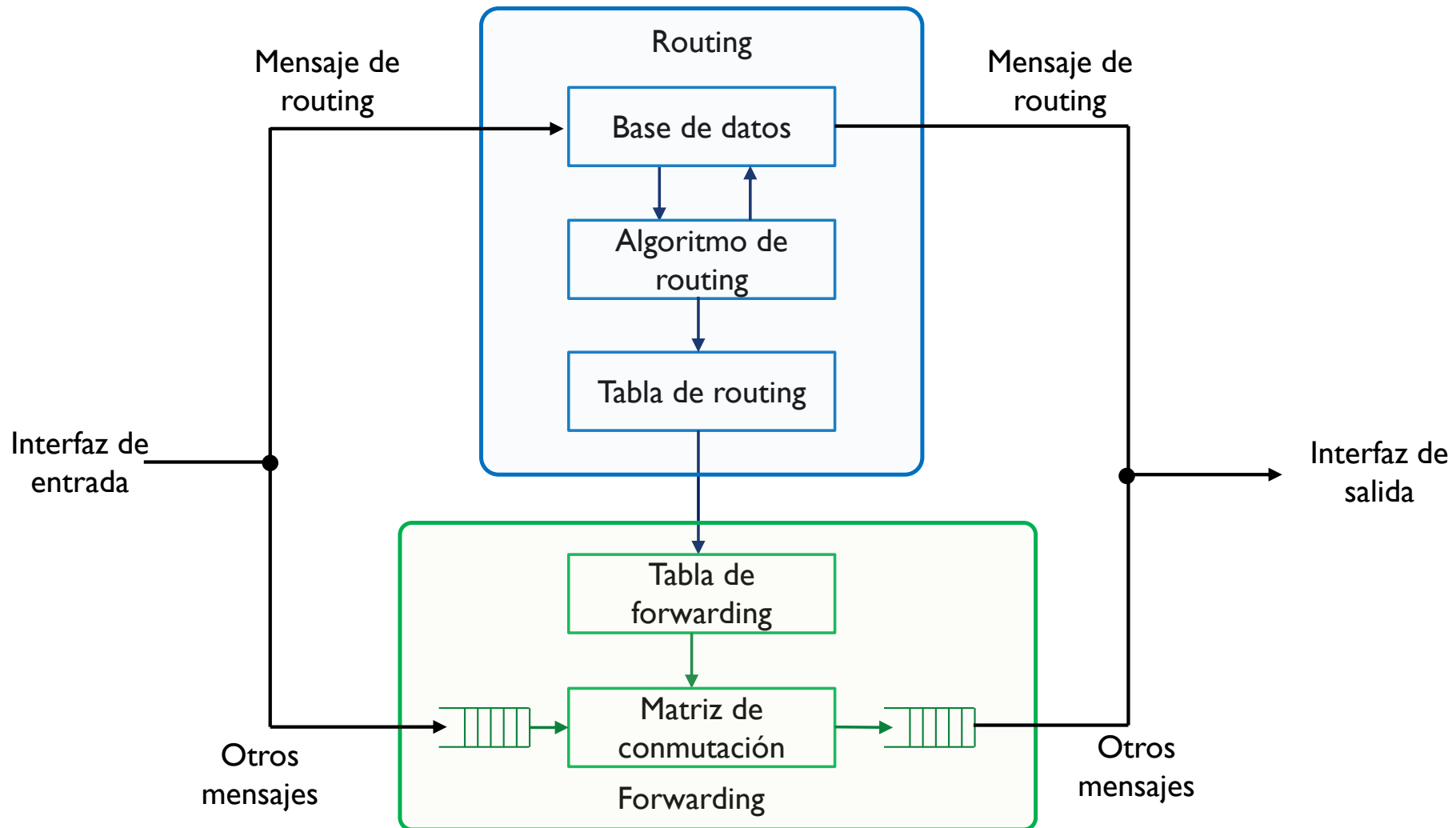
1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. Label Distribution Protocol (LDP)
6. Ejemplo de funcionamiento
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. MPLS con extensiones de TE
8. Ejemplo de funcionamiento
9. MPLS fast reroute

4.1 - Introducción a MPLS

- ▶ Las prestaciones de un router dependen en gran medida de su capacidad de re-enviar datagramas (forwarding) lo más rápido posible
- ▶ Este re-envío depende de su capacidad de toma de decisión sobre que hacer con un datagrama
- ▶ Hoy en día, los routers más potentes deben tomar decisiones en el orden de nanosegundos
- ▶ Se necesitan estructuras y mecanismos simples y eficientes

4.1 - Arquitectura de un router

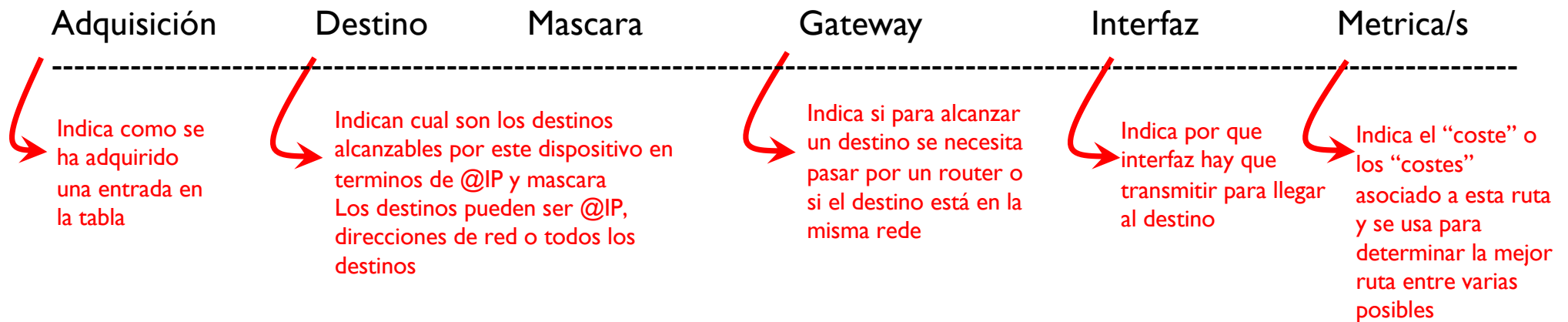
Bloques funcionales



4.1 - Arquitectura de un router

Tabla de encaminamiento

- ▶ Tabla de routing suele tener un formato estándar

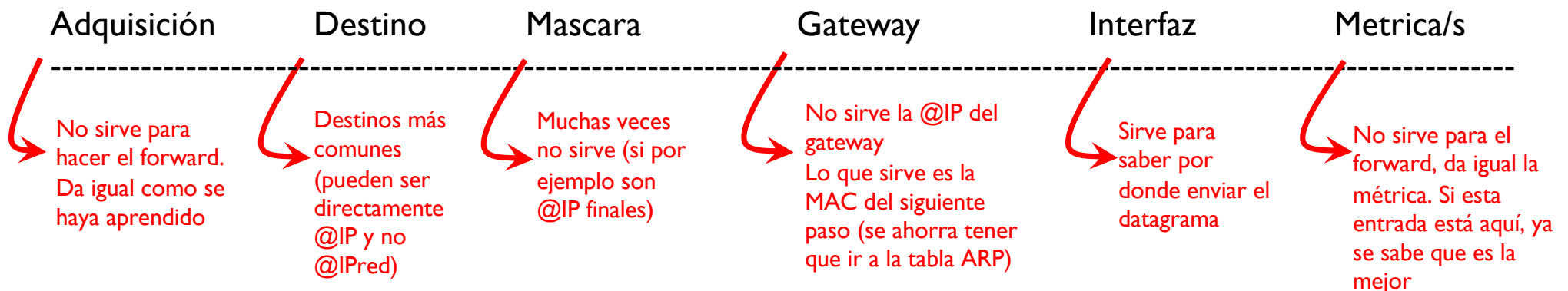


- ▶ Su estructura está optimizada para el protocolo y el algoritmo de routing
 - ▶ Actualización rápida en caso de un cambio en la tabla
 - ▶ Los mensajes de actualización pero no suelen ser muy frecuentes (orden de segundos)
 - ▶ Permite mantener muchas entradas (hoy en día puede haber tablas con 800,000 entradas)

4.1 - Arquitectura de un router

Tabla de forwarding

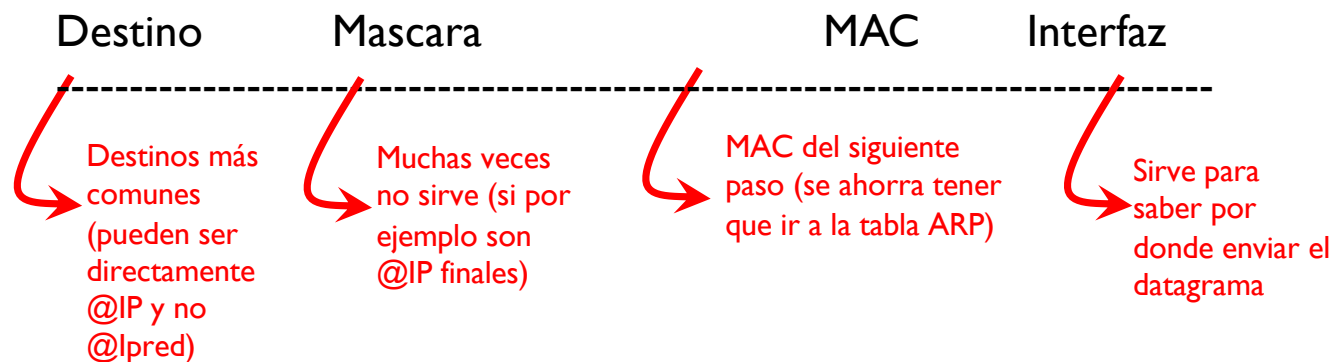
- ▶ Tabla de forwarding depende del fabricante y de la tecnología empleada
- ▶ Su estructura está optimizada para buscar que hacer con el datagrama y decidir por donde reenviarlo
 - ▶ Pueden llegar datagramas en el orden de ns
 - ▶ El rendimiento de un router depende en gran medida de cuanto rápido es en reenviar datagramas
- ▶ Las implementaciones más comunes solo cogen los destinos más usados y ponen MAC destino e interfaz de salida



4.1 - Arquitectura de un router

Tabla de forwarding

- ▶ Tabla de forwarding depende del fabricante y de la tecnología empleada
- ▶ Su estructura está optimizada para buscar que hacer con el datagrama y decidir por donde reenviarlo
 - ▶ Pueden llegar datagramas en el orden de ns
 - ▶ El rendimiento de un router depende en gran medida de cuanto rápido es en reenviar datagramas
- ▶ Las implementaciones más comunes solo cogen los destinos más usados y ponen MAC destino e interfaz de salida



4.1 - Introducción a MPLS

- ▶ MultiProtocol Label Switching
- ▶ Es un protocolo que nació para agilizar y acelerar el proceso de consulta y toma de decisión de tablas de forwarding
- ▶ Actualmente además proporciona
 - ▶ Servicio VPN
 - ▶ Mecanismos de búsqueda rápida de caminos alternativos en caso de fallo
 - ▶ Ingeniería de Trafico (TE) optimizando los recursos de red a las demandas de los clientes

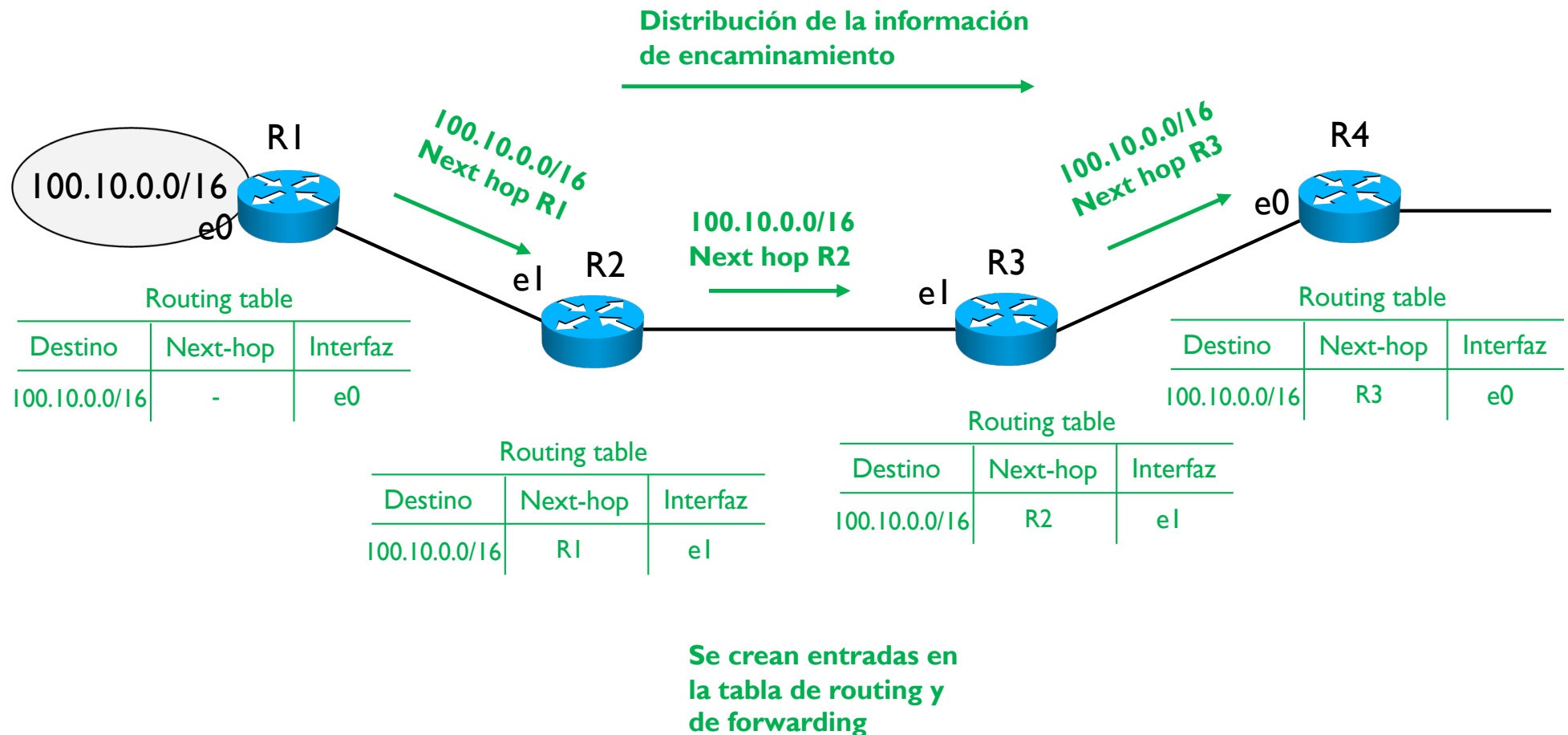
4.1 - Introducción a MPLS

- ▶ Con IP, la consulta de las tablas de forwarding se hace de esta forma
- ▶ Operación que se llaman IP route lookup



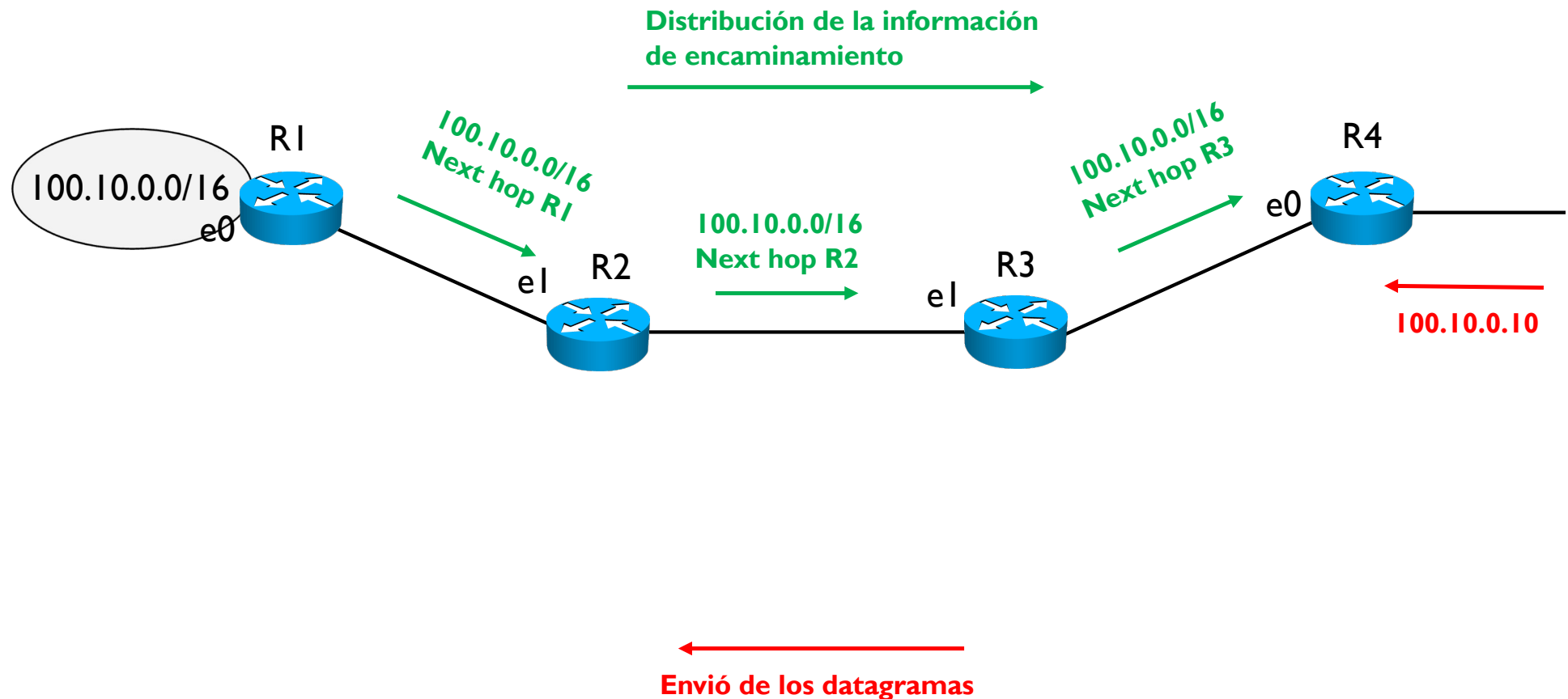
4.1 - Introducción a MPLS

- ▶ Con IP, la consulta de las tablas de forwarding se hace de esta forma
- ▶ Operación que se llaman IP route lookup



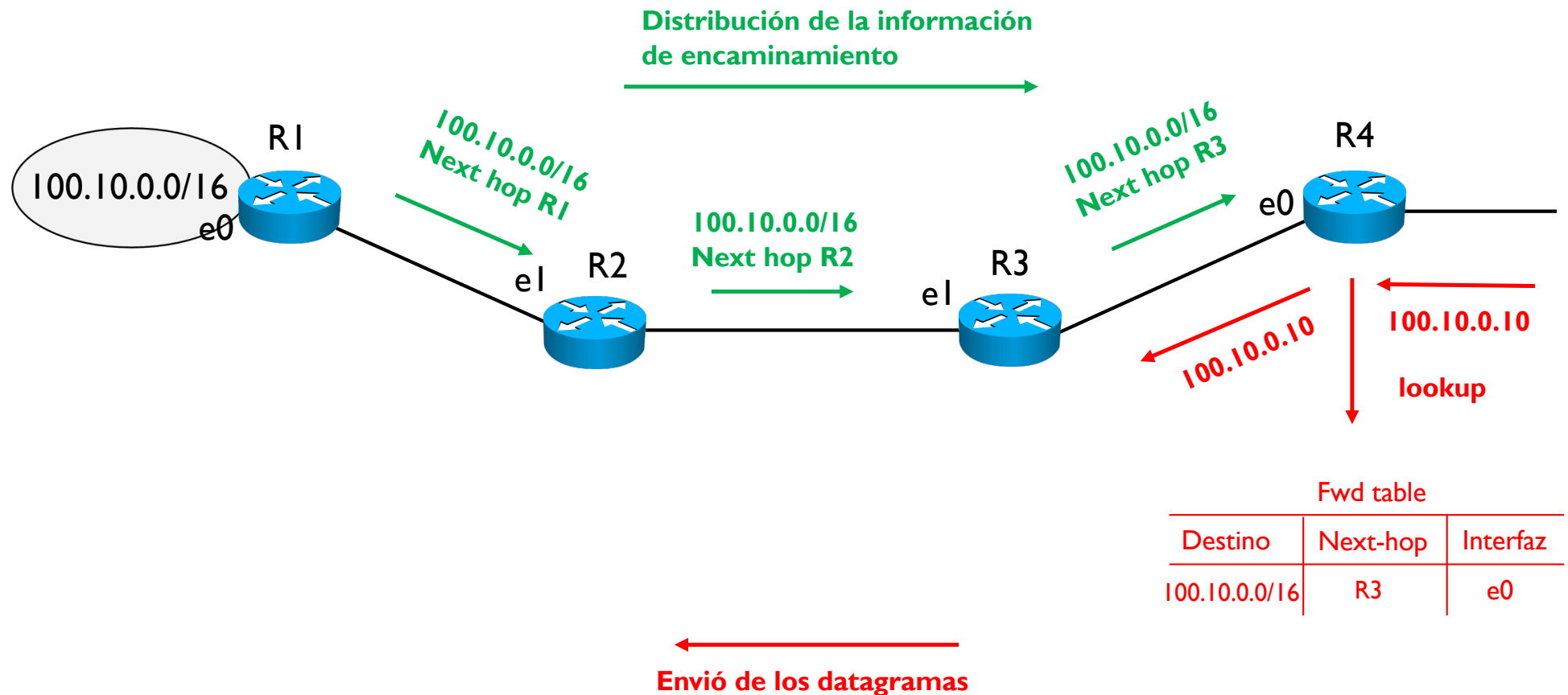
4.1 - Introducción a MPLS

- ▶ Con IP, la consulta de las tablas de forwarding se hace de esta forma
- ▶ Operación que se llaman IP route lookup



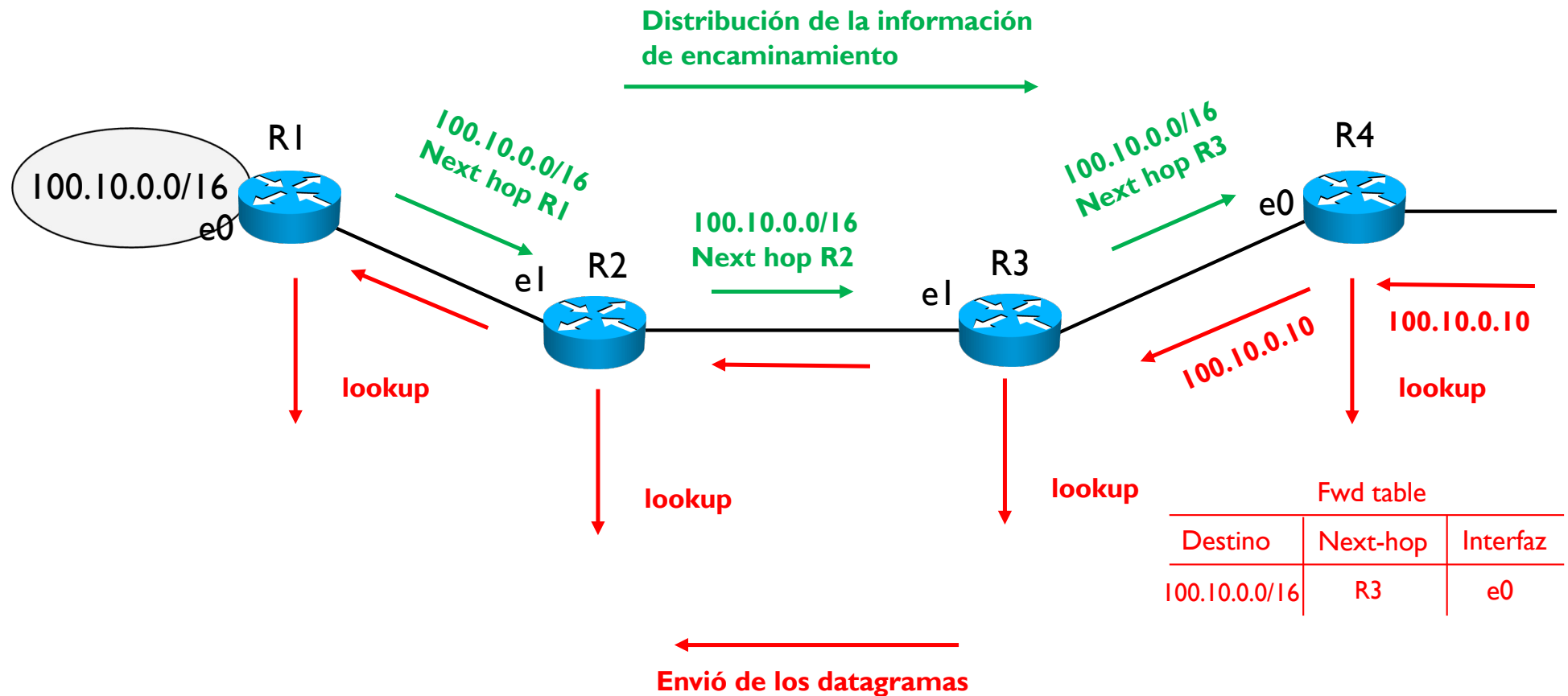
4.1 - Introducción a MPLS

- ▶ Con IP, la consulta de las tablas de forwarding se hace de esta forma
- ▶ Operación que se llaman IP route lookup



4.1 - Introducción a MPLS

- ▶ Con IP, la consulta de las tablas de forwarding se hace de esta forma
- ▶ Operación que se llaman IP route lookup

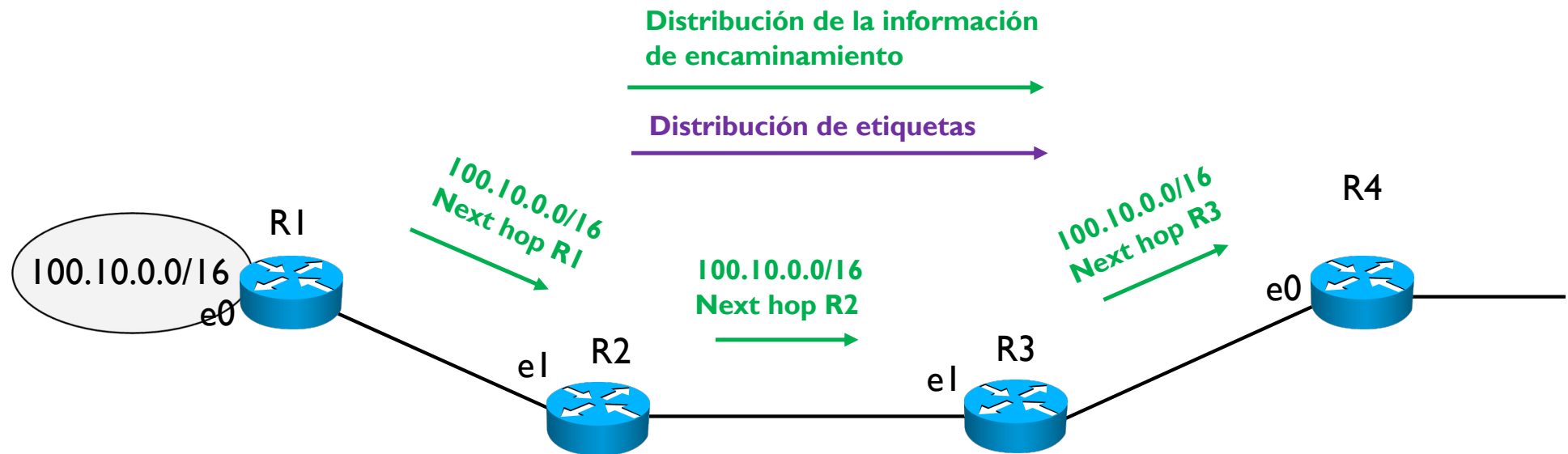


4.1 - Introducción a MPLS

- ▶ En IP hay entonces que consultar tablas de forwarding (o tablas de encaminamiento) que suelen tener muchas entradas
- ▶ En MPLS, en cambio, todo funciona por etiquetas (o labels)
 - ▶ Una etiqueta puede corresponder a un prefijo, a un conjunto de prefijos, a un conjunto de IPs, etc.
 - ▶ Las etiquetas tienen un significado local para cada pareja de routers MPLS
 - ▶ Un router MPLS consulta una tabla de etiqueta para reenviar los paquetes MPLS
 - ▶ Solo los routers de frontera de esta red MPLS necesitan hacer un IP route lookup
 - ▶ A través de estas etiquetas, se construyen caminos MPLS llamados Label Switched Path (LSP)
- ▶ MPLS no es un protocolo de encaminamiento
 - ▶ Se sigue necesitando un protocolo de encaminamiento para intercambiar información de encaminamiento entre routers
 - ▶ MPLS añade una funcionalidad al protocolo de encaminamiento

4.1 - Introducción a MPLS

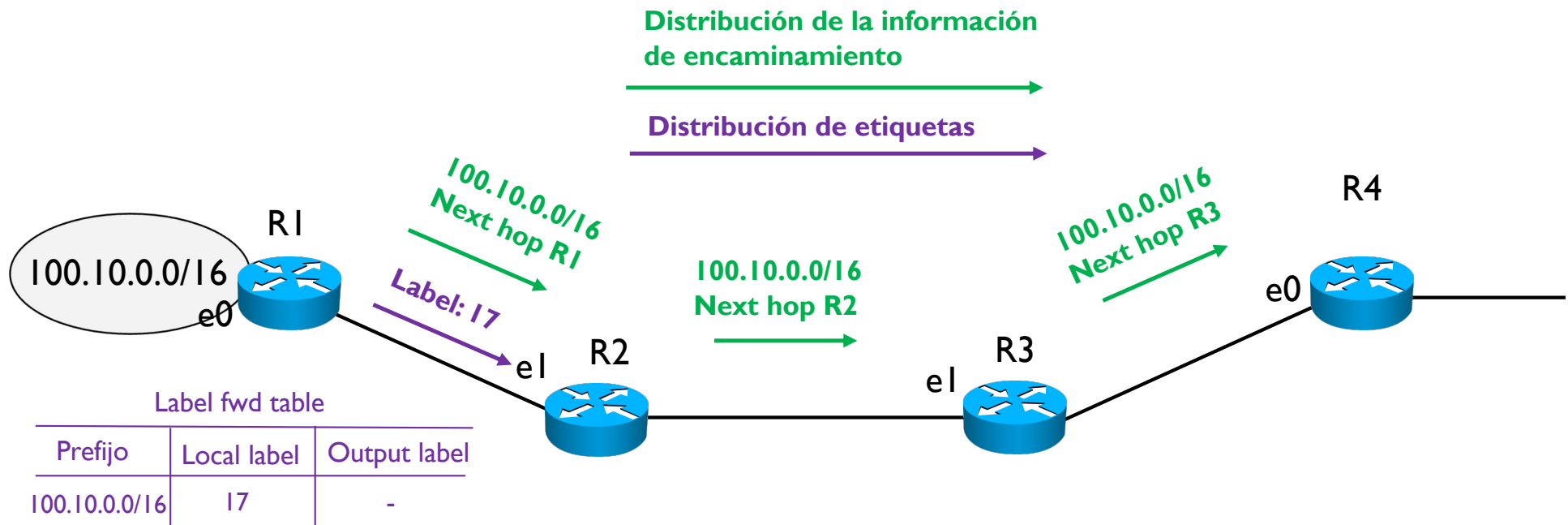
- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



Además de crear entradas en las tablas de routing y forwarding, se crean otras tablas

4.1 - Introducción a MPLS

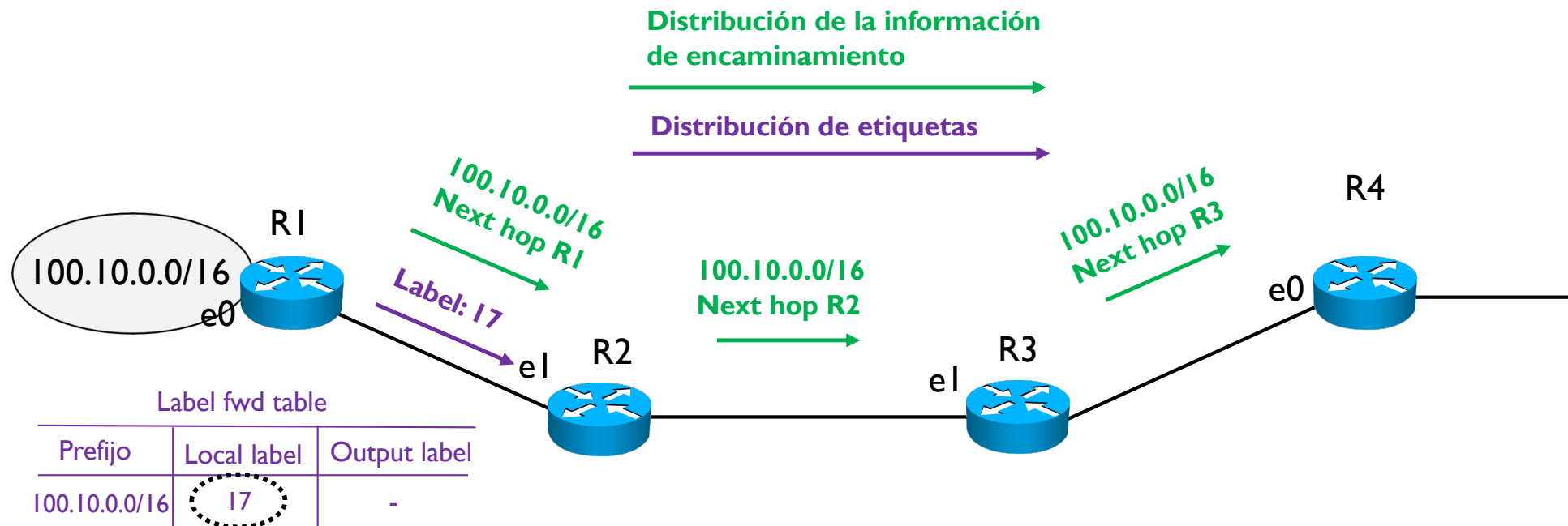
- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



Además de crear entradas en las tablas de routing y forwarding, se crean otras tablas

4.1 - Introducción a MPLS

- ▶ Con MPLS, la operación de forwarding funciona por etiquetas

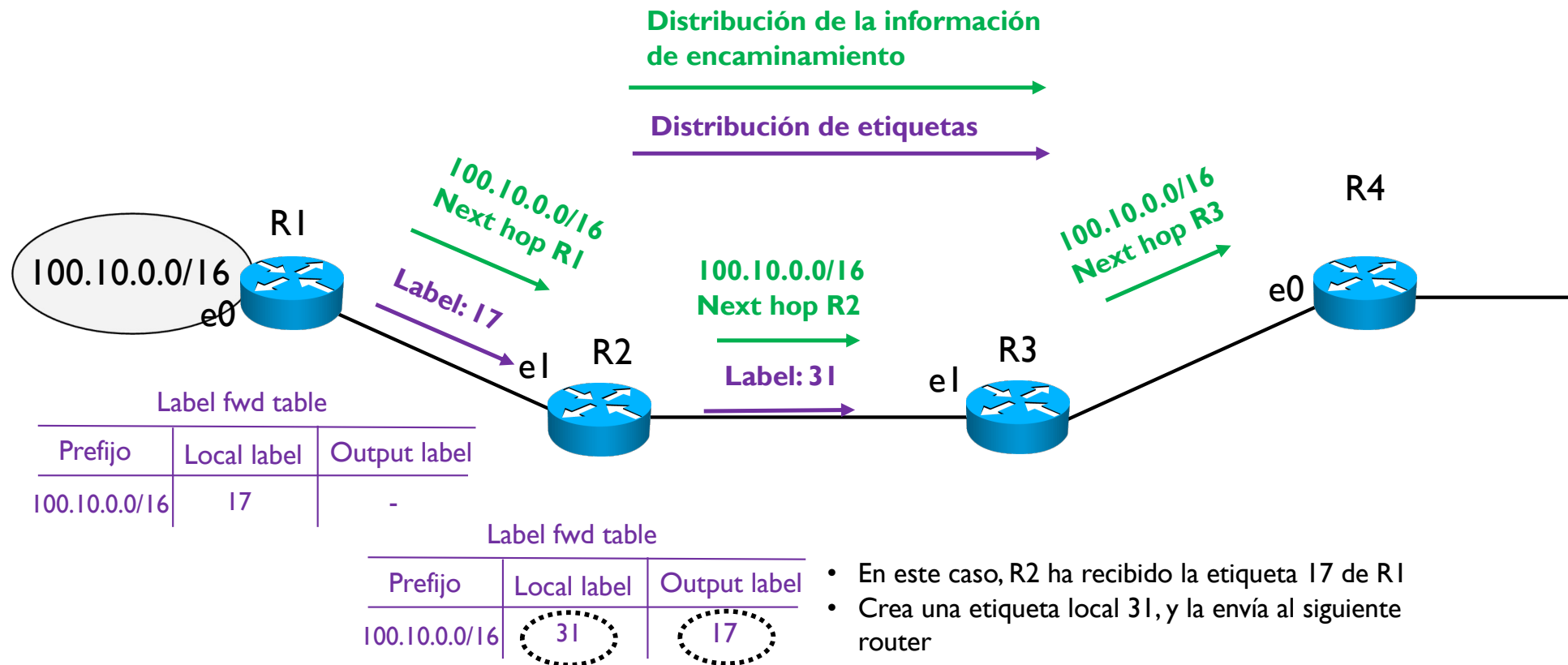


- R1 asocia al prefijo una etiqueta local y la envía al siguiente router
- Pone esta asociación en su tabla de etiquetas
- Como es el origen, no pone ninguna de salida

Además de crear entradas en las tablas de routing y forwarding, se crean otras tablas

4.1 - Introducción a MPLS

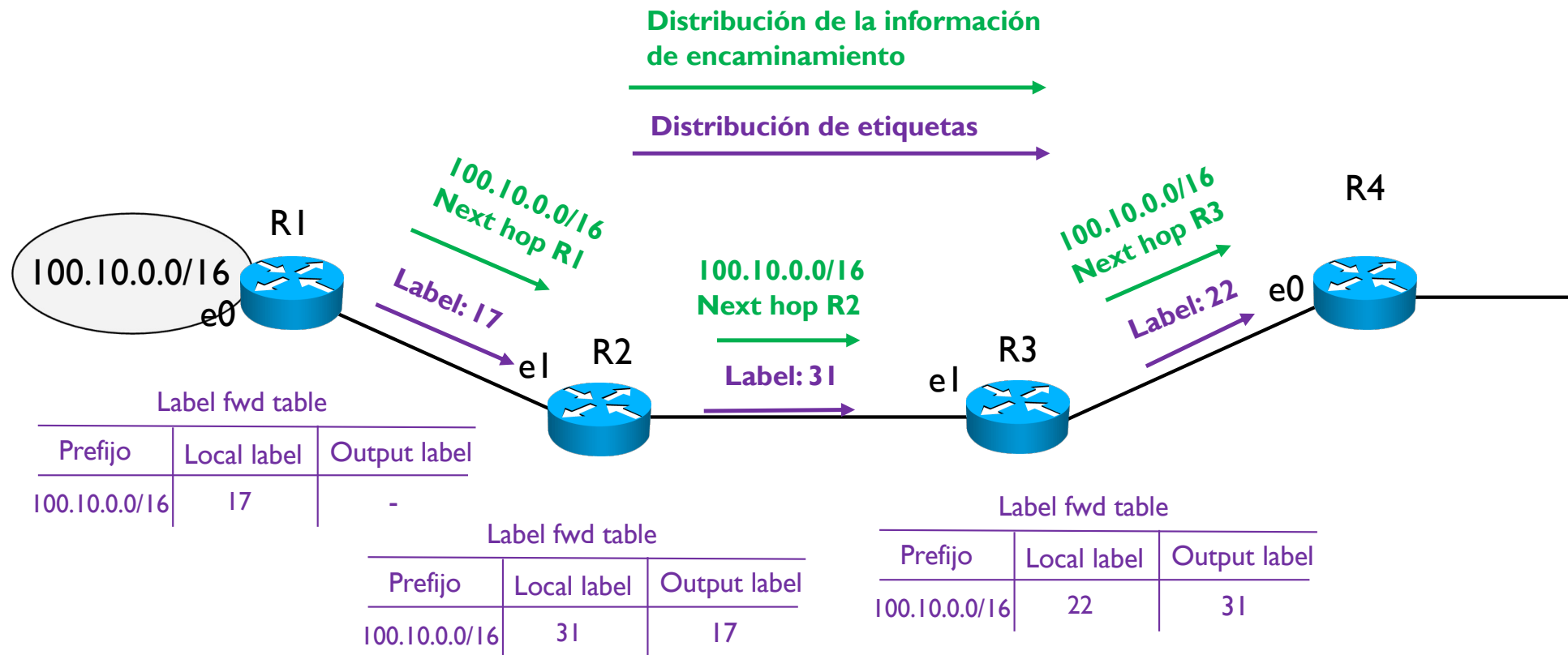
- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



Además de crear entradas en las tablas de routing y forwarding, se crean otras tablas

4.1 - Introducción a MPLS

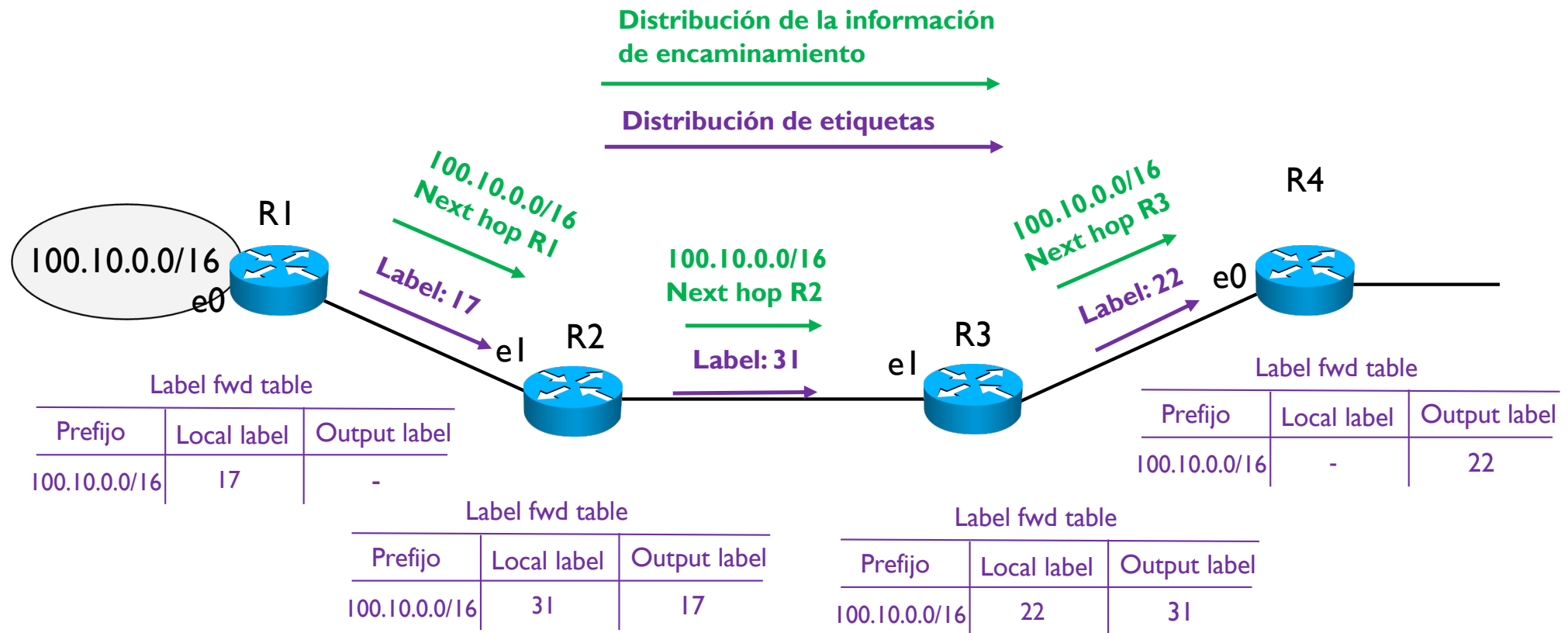
- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



Además de crear entradas en las tablas de routing y forwarding, se crean otras tablas

4.1 - Introducción a MPLS

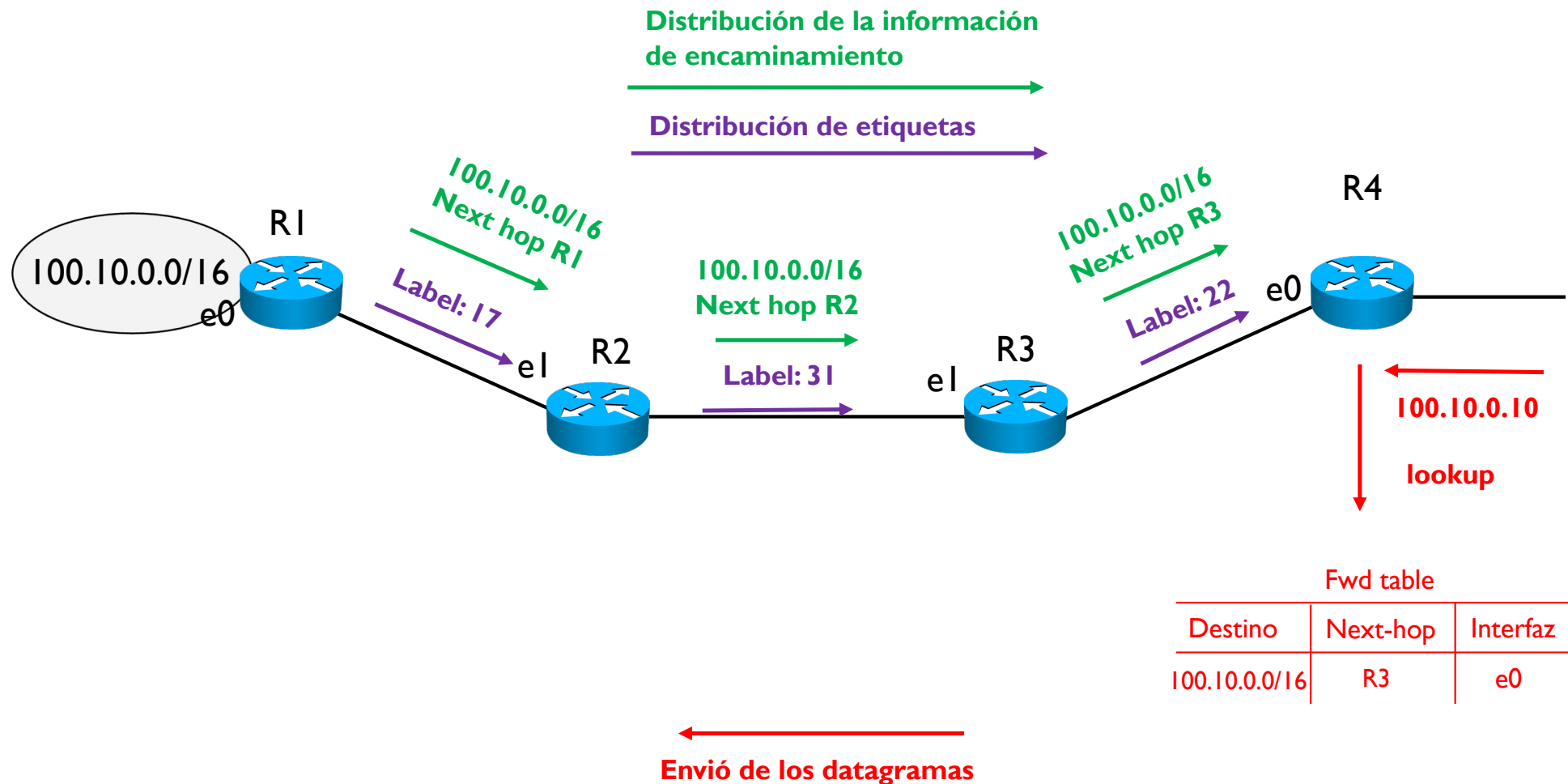
- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



- En este caso, R4 es el último router donde se usa MPLS
- No hay etiqueta local

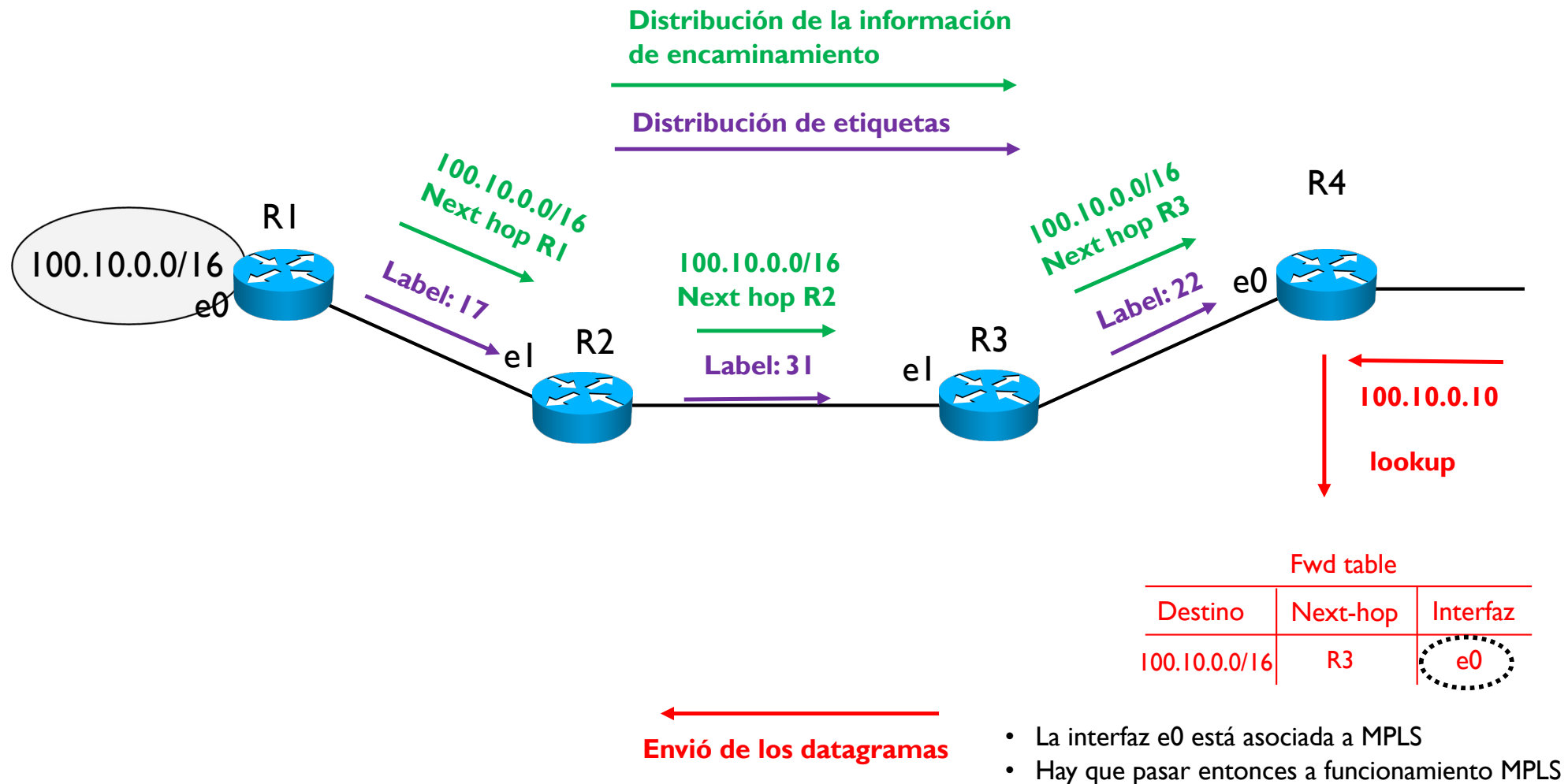
4.1 - Introducción a MPLS

- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



4.1 - Introducción a MPLS

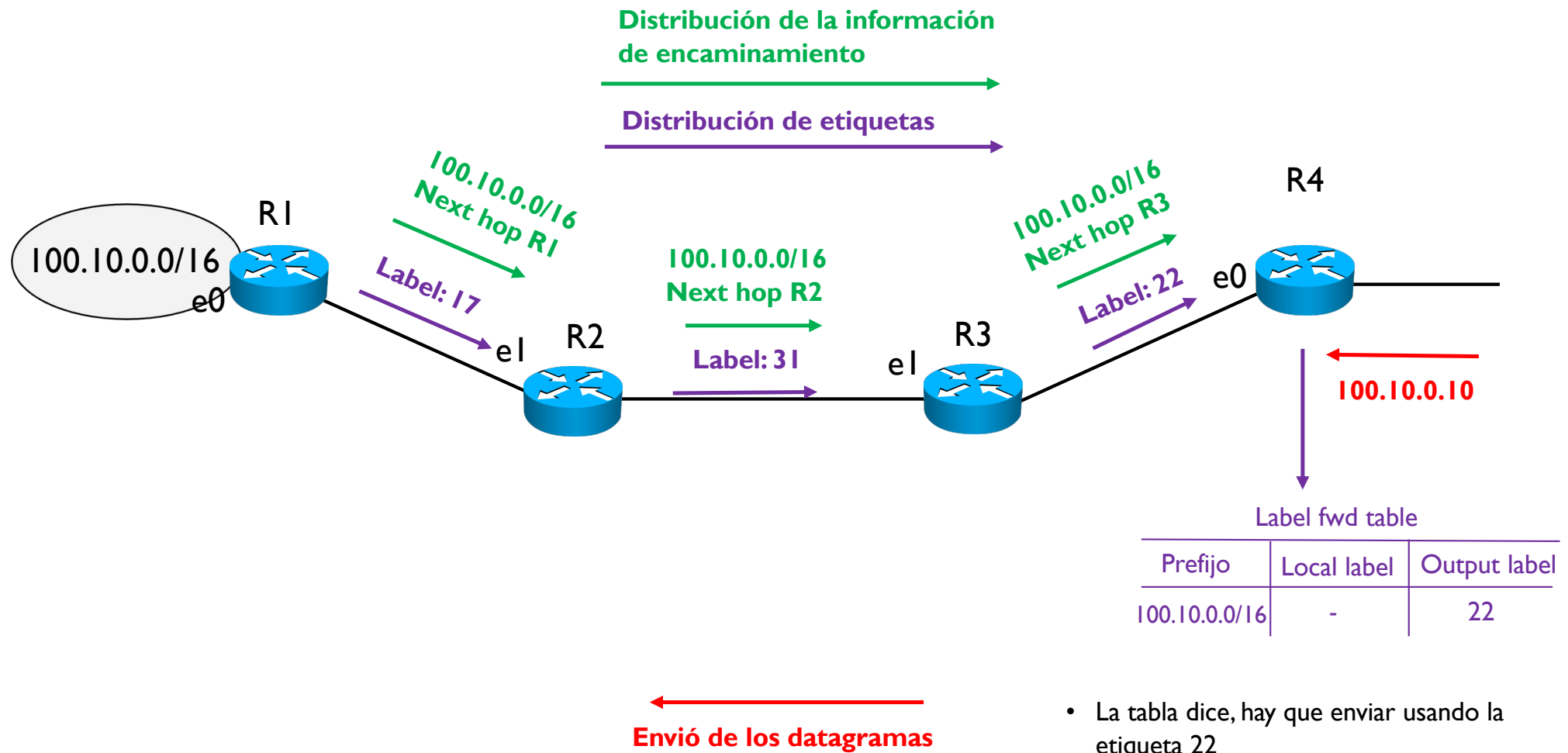
- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



- La interfaz e0 está asociada a MPLS
- Hay que pasar entonces a funcionamiento MPLS

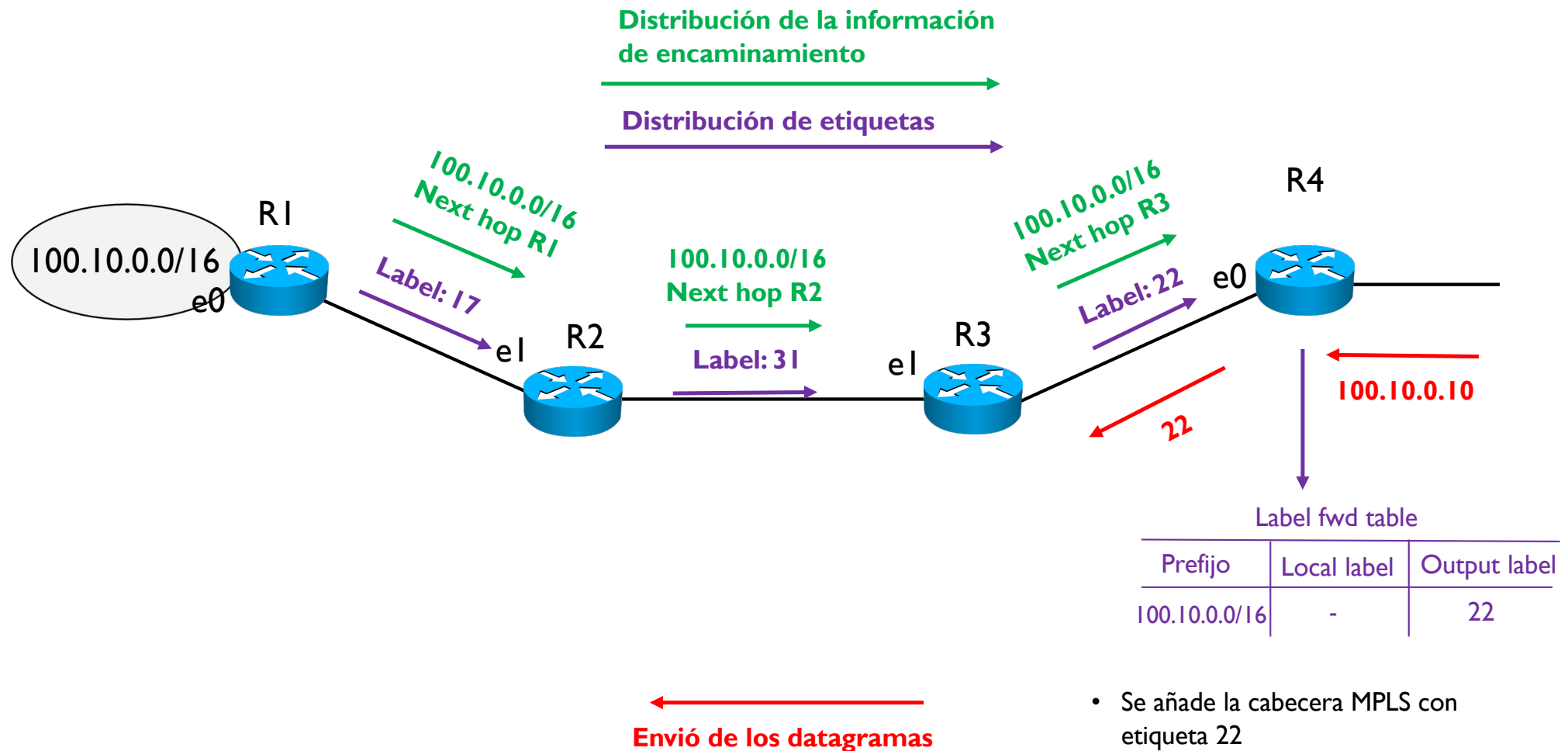
4.1 - Introducción a MPLS

- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



4.1 - Introducción a MPLS

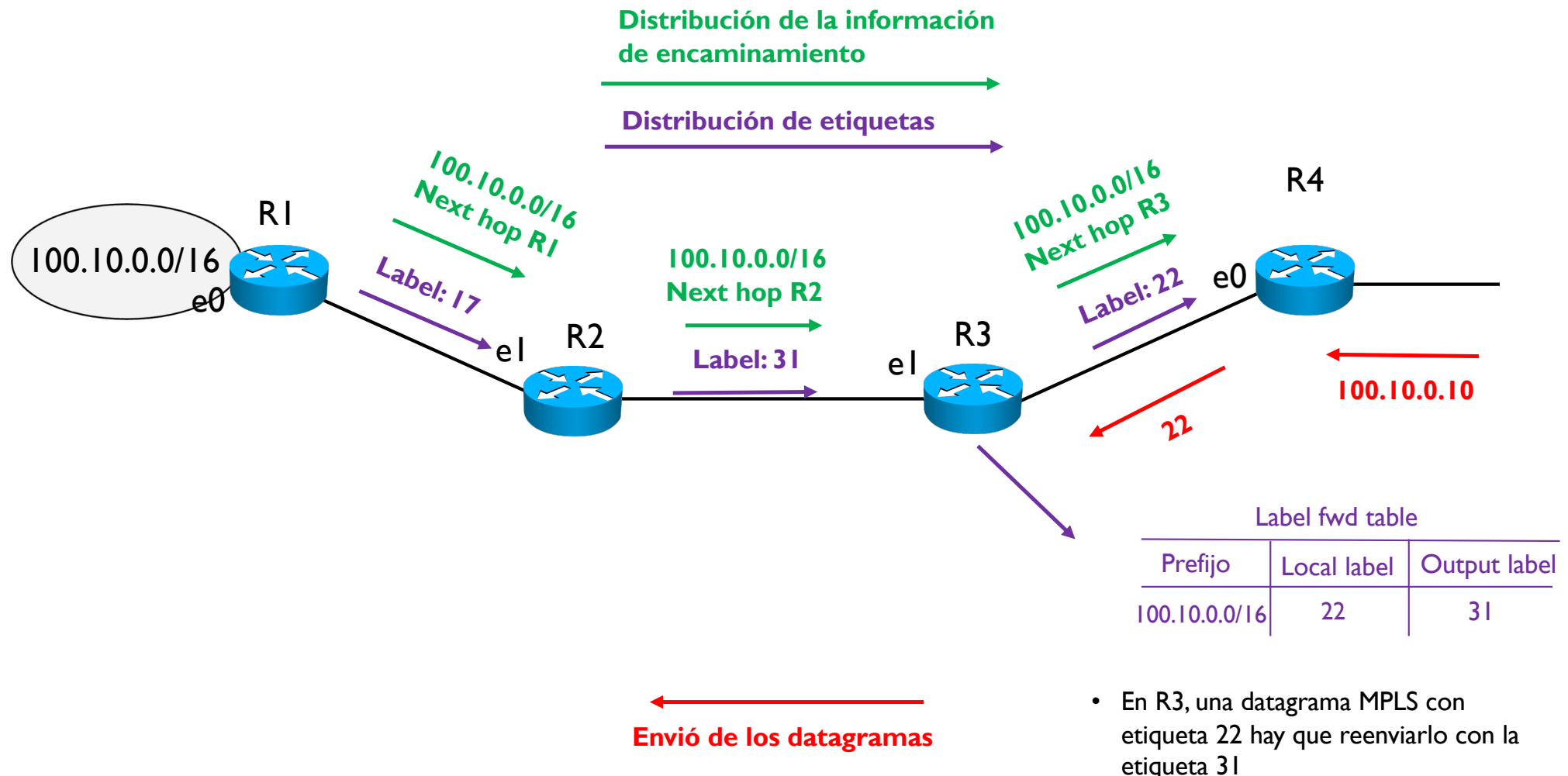
- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



- Se añade la cabecera MPLS con etiqueta 22
- Esta operación se llama label push

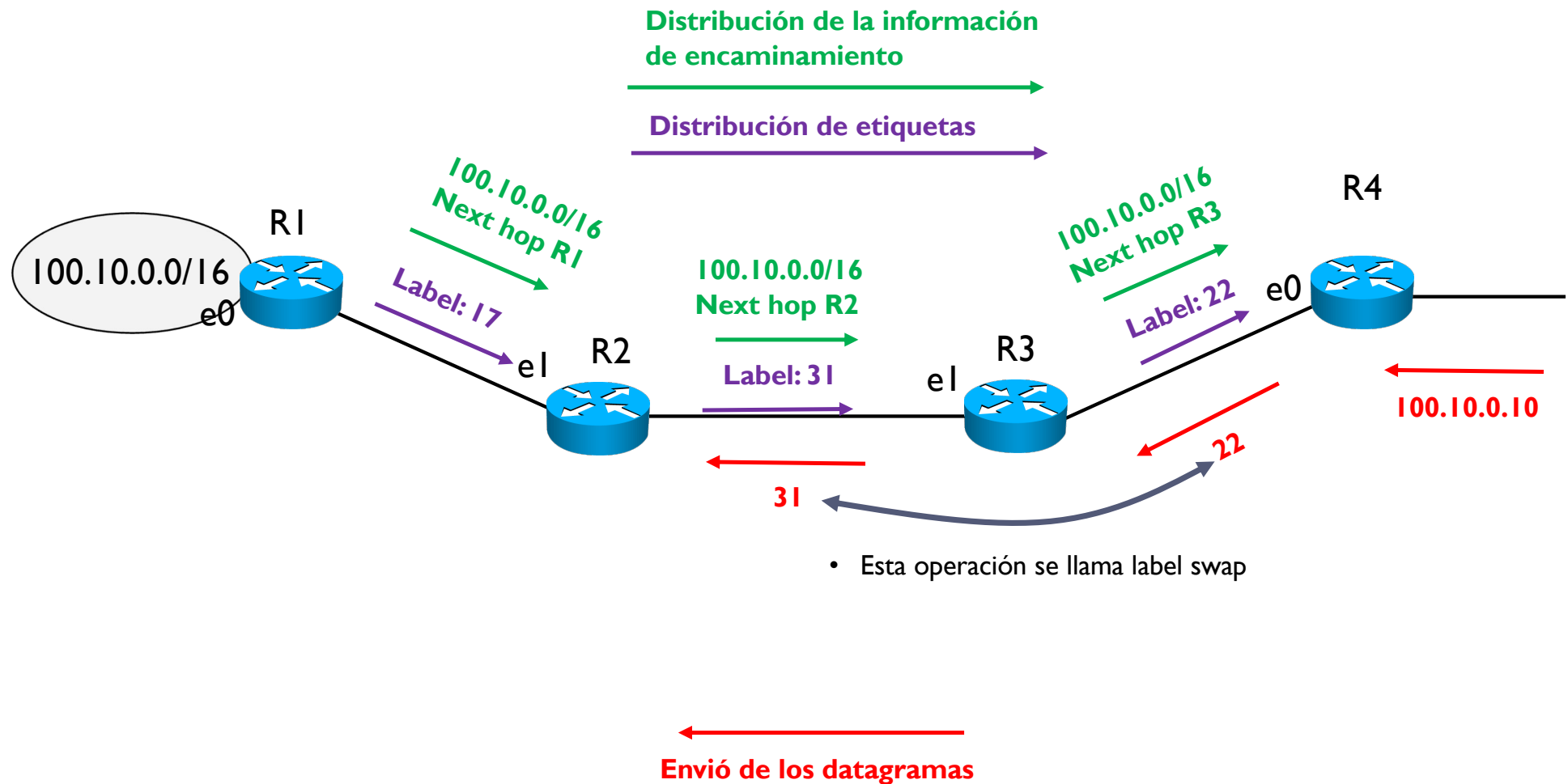
4.1 - Introducción a MPLS

- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



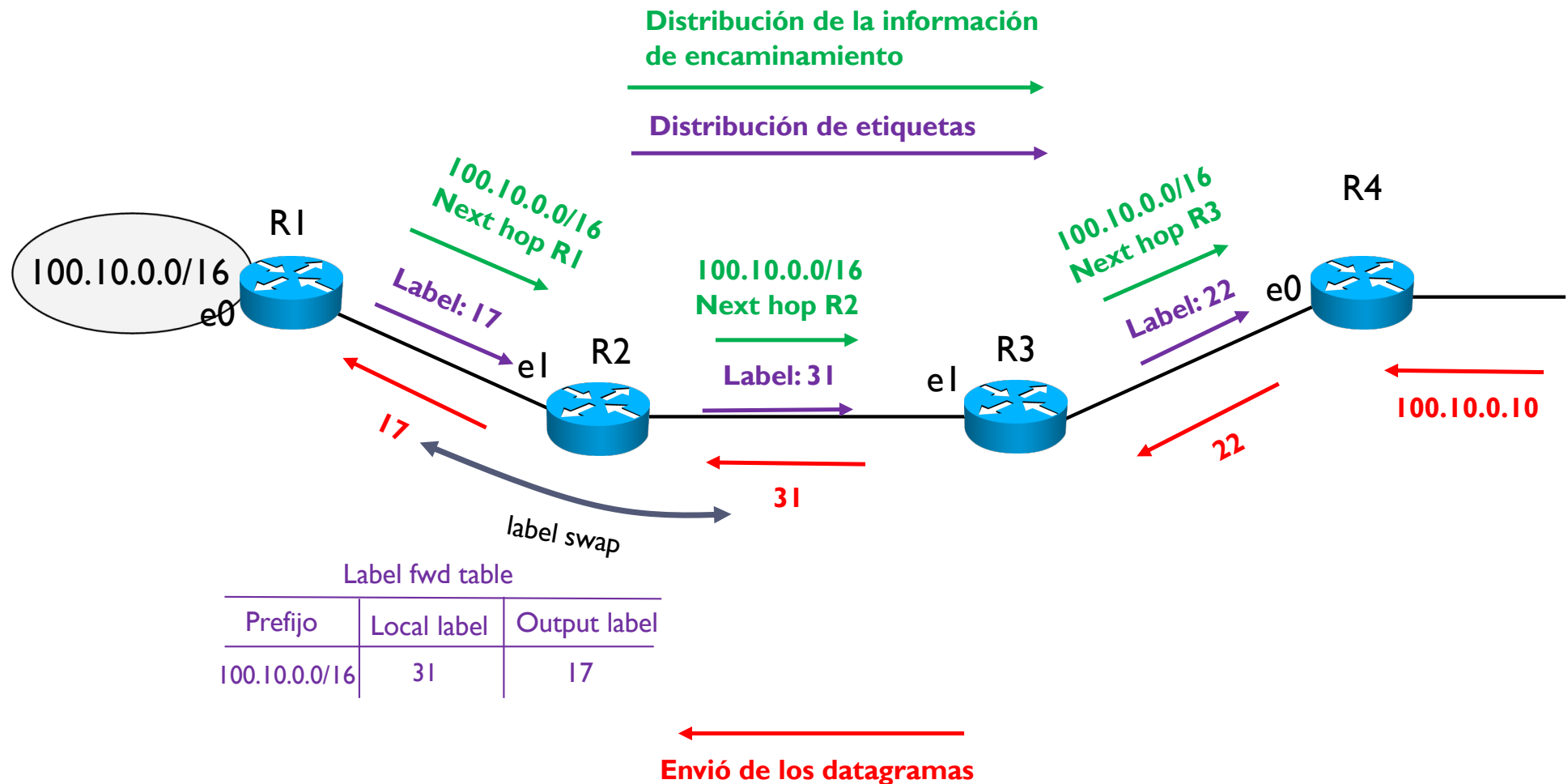
4.1 - Introducción a MPLS

- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



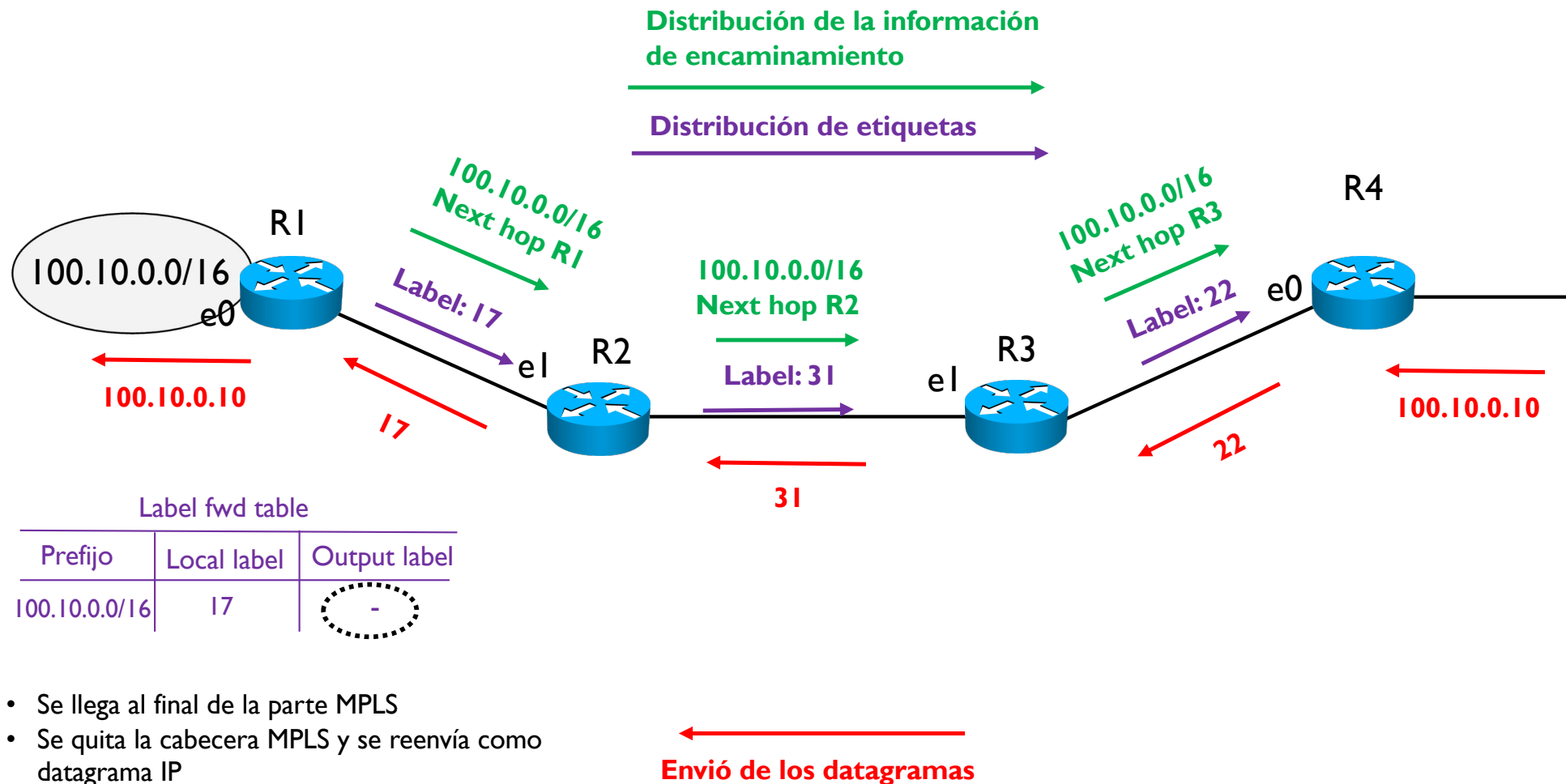
4.1 - Introducción a MPLS

- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



4.1 - Introducción a MPLS

- ▶ Con MPLS, la operación de forwarding funciona por etiquetas



- Se llega al final de la parte MPLS
- Se quita la cabecera MPLS y se reenvía como datagrama IP
- Esta operación se llama label pop

4.2 - Terminología MPLS

- ▶ **Control Plane**
 - ▶ Proporciona la funcionalidad de identificar los prefijos alcanzables y como
 - ▶ Protocolo de encaminamiento como OSPF o RIP
 - ▶ Protocolo de distribución de etiquetas como LDP o RSVP
- ▶ **Data Plane**
 - ▶ Proporciona la funcionalidad de reenviar datagramas (forwarding)
- ▶ **Label Switch Router (LSR)**
 - ▶ Un router MPLS que trata datagramas MPLS, sabe hacer label pop, push y swap
- ▶ **Edge LSR (E-LSR) o Label Edge Router (LER)**
 - ▶ Un LSR en la frontera de una infraestructura MPLS
 - ▶ Un ingress E-LSR hace label push
 - ▶ Un egress E-LSR hace label pop
- ▶ **Label Switched Path**
 - ▶ Un camino MPLS entre un ingress E-LSR y un egress E-LSR

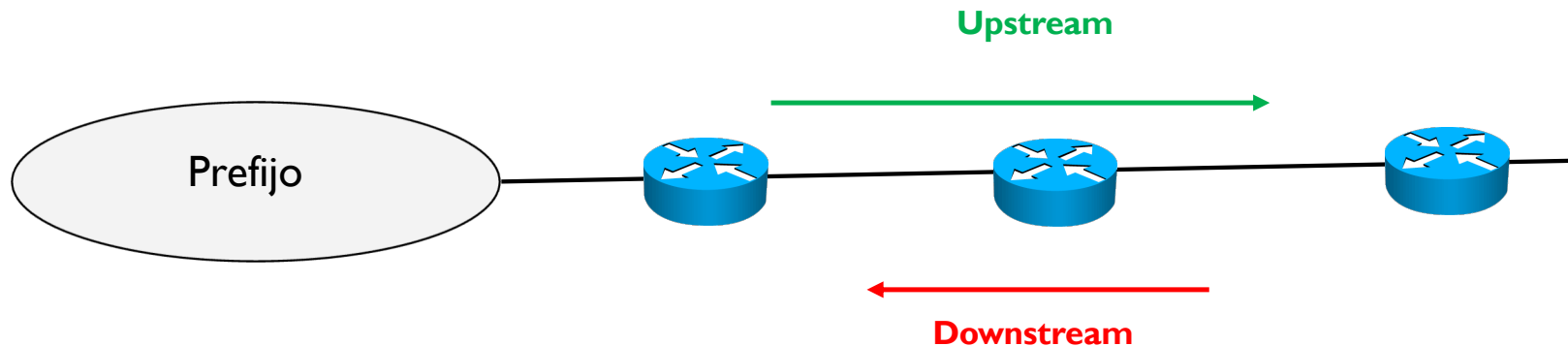
4.2 - Terminología MPLS

- ▶ **Upstream**

- ▶ Sentido por donde va la información de encaminamiento y de distribución de etiquetas a partir de un prefijo

- ▶ **Downstream**

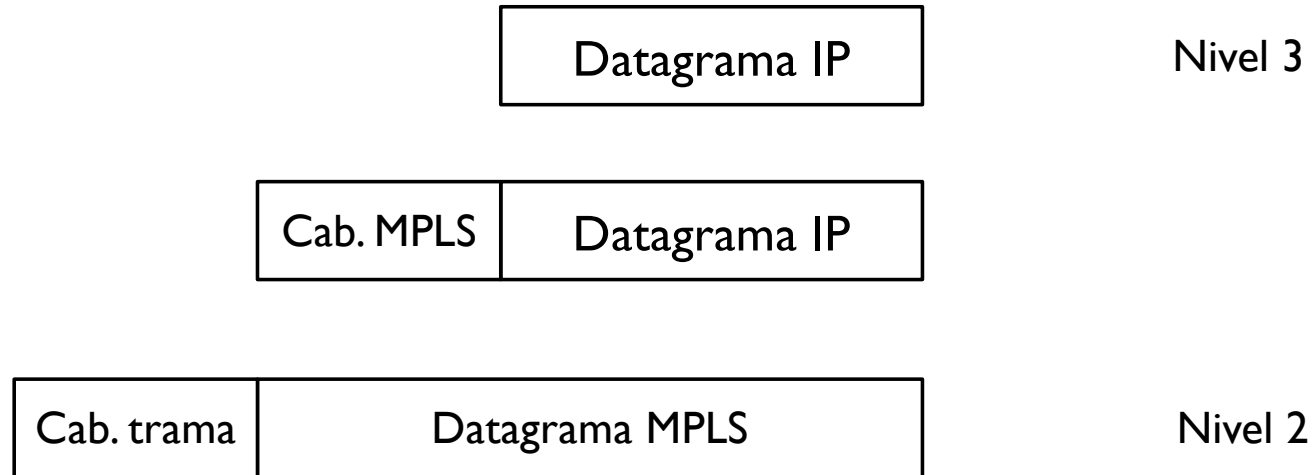
- ▶ Sentido por donde circulan los datagramas de datos hacia un prefijo



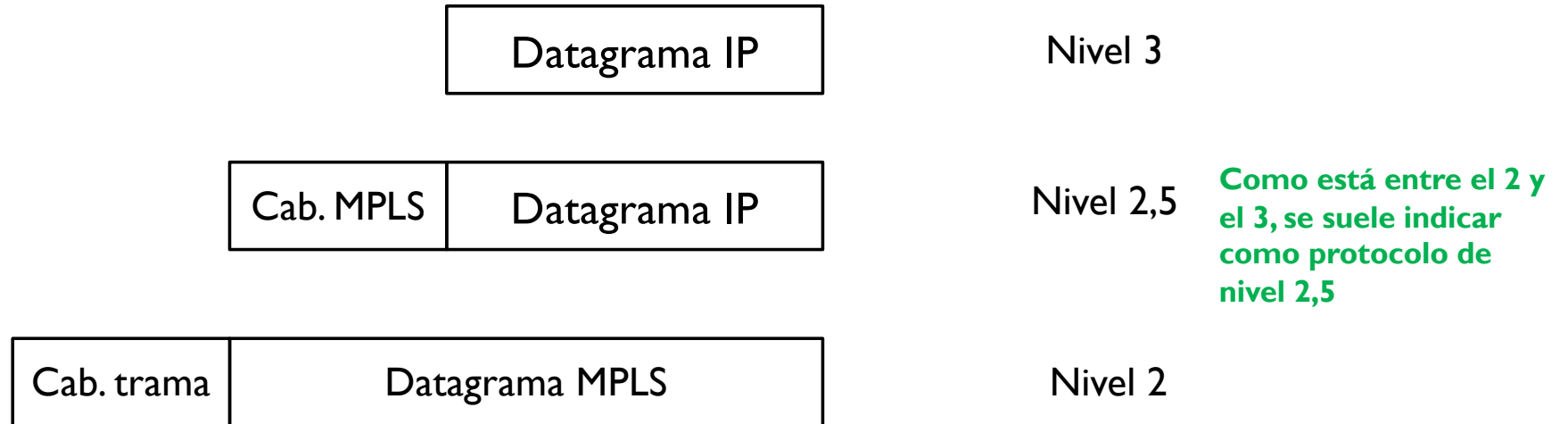
4. MultiProtocol Label Switching (MPLS)

1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. Label Distribution Protocol (LDP)
6. Ejemplo de funcionamiento
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. MPLS con extensiones de TE
8. Ejemplo de funcionamiento
9. MPLS fast reroute

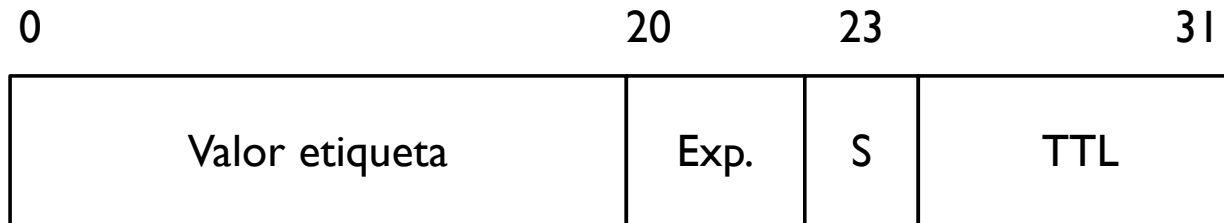
4.3 - Formato de una etiqueta



4.3 - Formato de una etiqueta

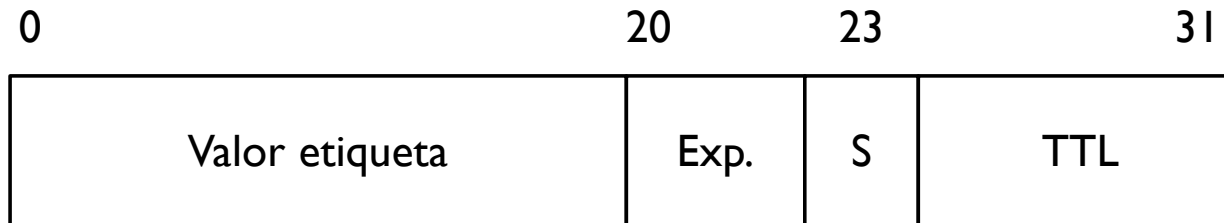


4.3 - Formato de una etiqueta



- ▶ **Valor etiquetas (20 bits)**
 - ▶ Los valores de 0 a 15 están reservados para funciones particulares
- ▶ **Exp. (3 bits)**
 - ▶ Campo experimental usado con la idea de definir prioridades diferentes
 - ▶ No se suele usar
- ▶ **S (1 bit)**
 - ▶ Se usa para poder encapsular cabeceras MPLS dentro de otras cabeceras MPLS
 - ▶ Función que se llama label stack
 - ▶ Cuando el valor es 0, quiere decir que hay otra etiqueta interna
 - ▶ Cuando es 1, es la última cabecera

4.3 - Formato de una etiqueta

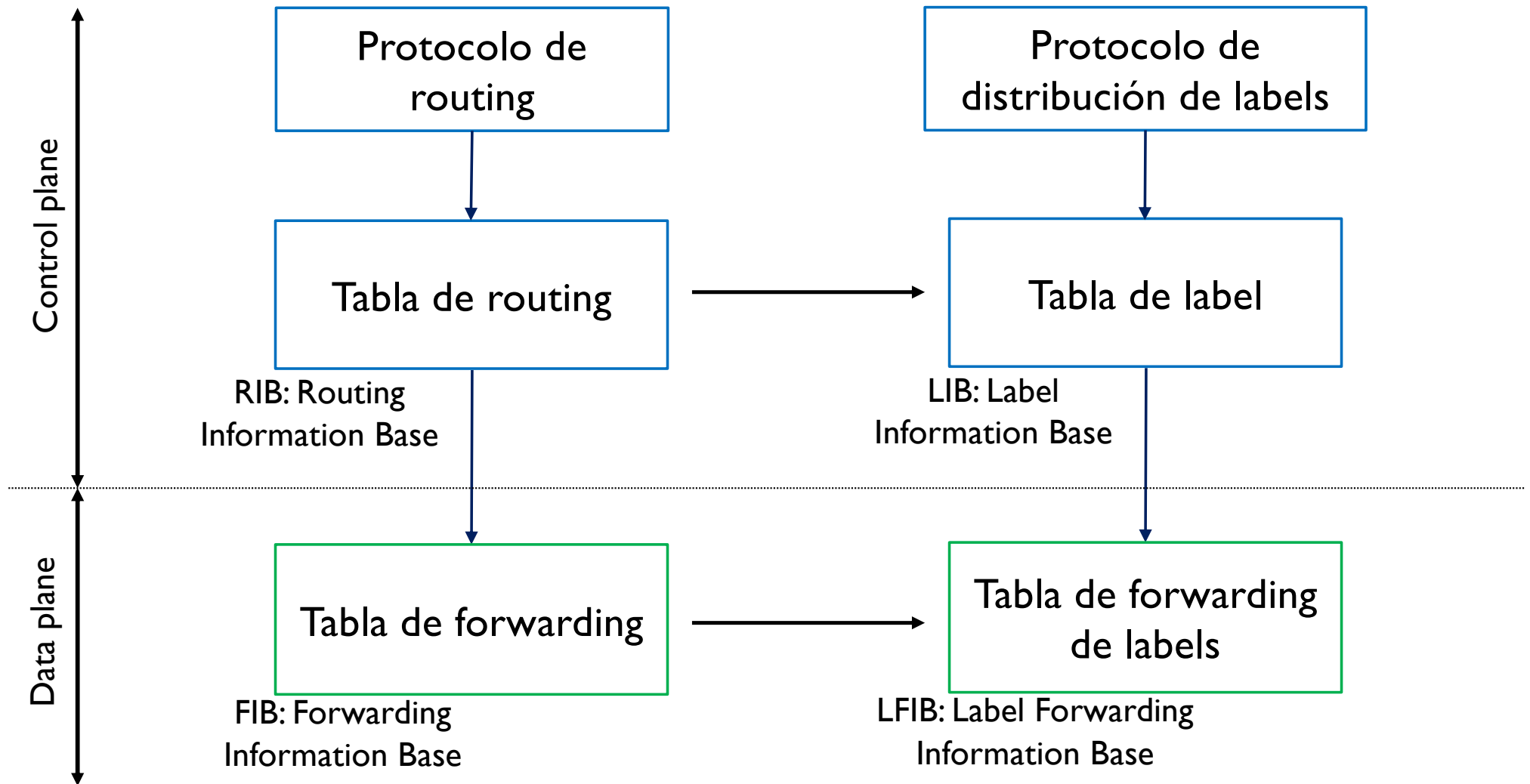


- ▶ **TTL (8 bits)**
 - ▶ Tiempo de vida del datagrama MPLS
 - ▶ Funciona igual que en IP
 - ▶ Sirve para que un datagrama perdido no se quede en la red eternamente
 - ▶ El origen pone un valor; cada router reduce el valor de 1; si el valor llega a 0, el datagrama se tira

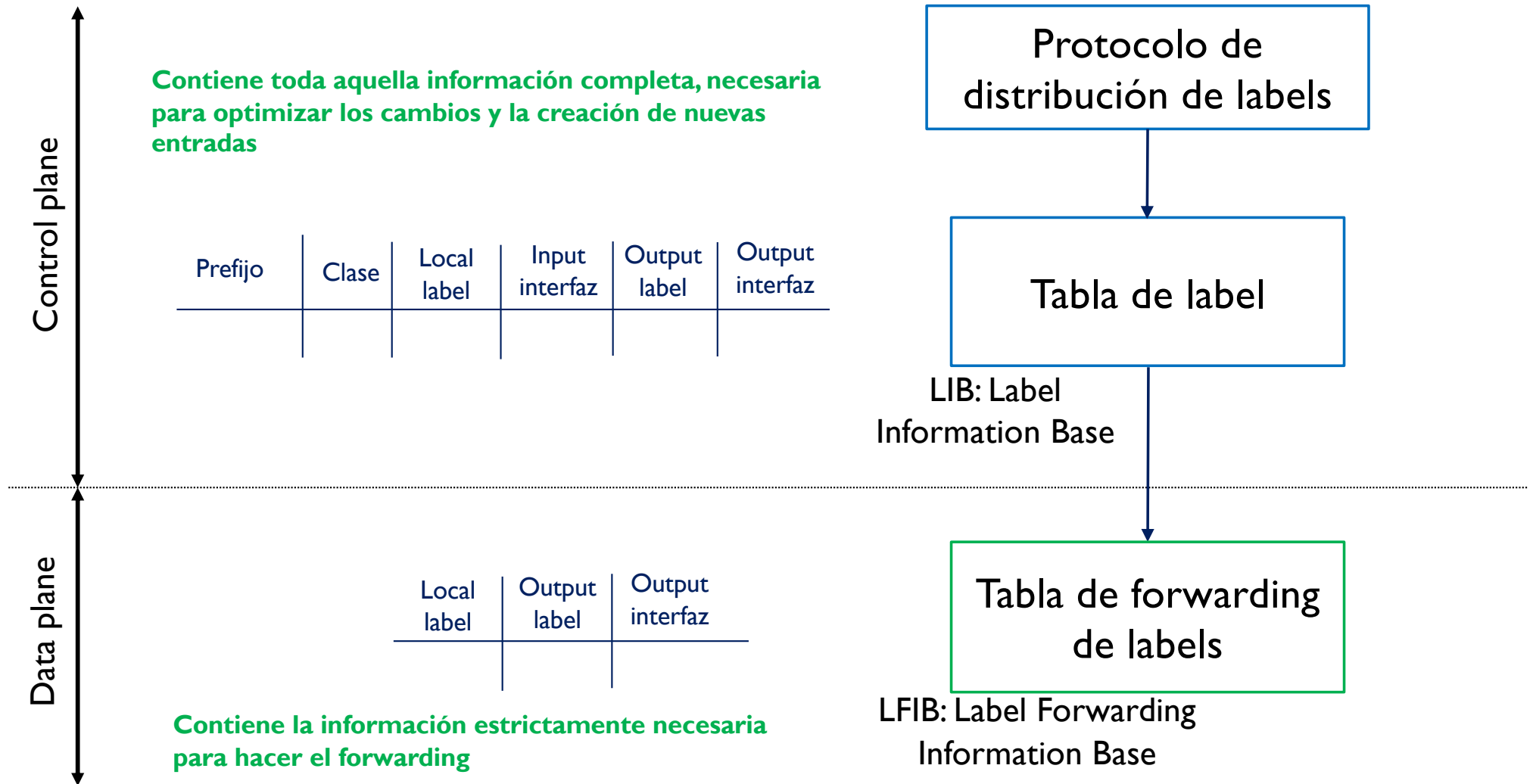
4. MultiProtocol Label Switching (MPLS)

1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. Label Distribution Protocol (LDP)
6. Ejemplo de funcionamiento
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. MPLS con extensiones de TE
8. Ejemplo de funcionamiento
9. MPLS fast reroute

4.4 - Estructuras de las tablas MPLS



4.4 - Estructuras de las tablas MPLS

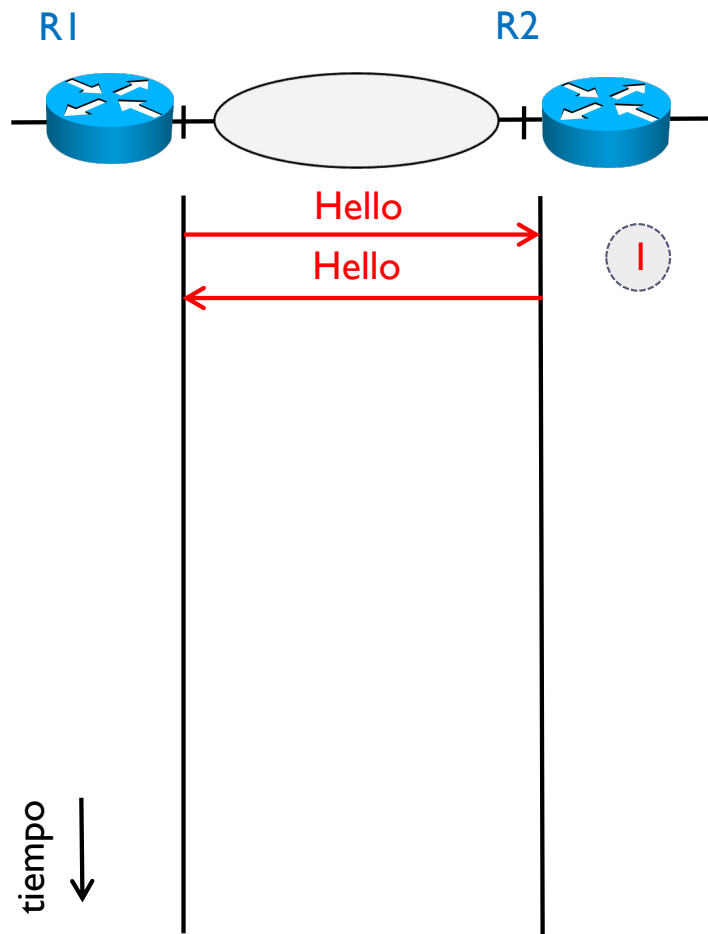


4. MultiProtocol Label Switching (MPLS)

1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. **Label Distribution Protocol (LDP)**
6. Ejemplo de funcionamiento
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. MPLS con extensiones de TE
8. Ejemplo de funcionamiento
9. MPLS fast reroute

4.5 - Label Distribution Protocol

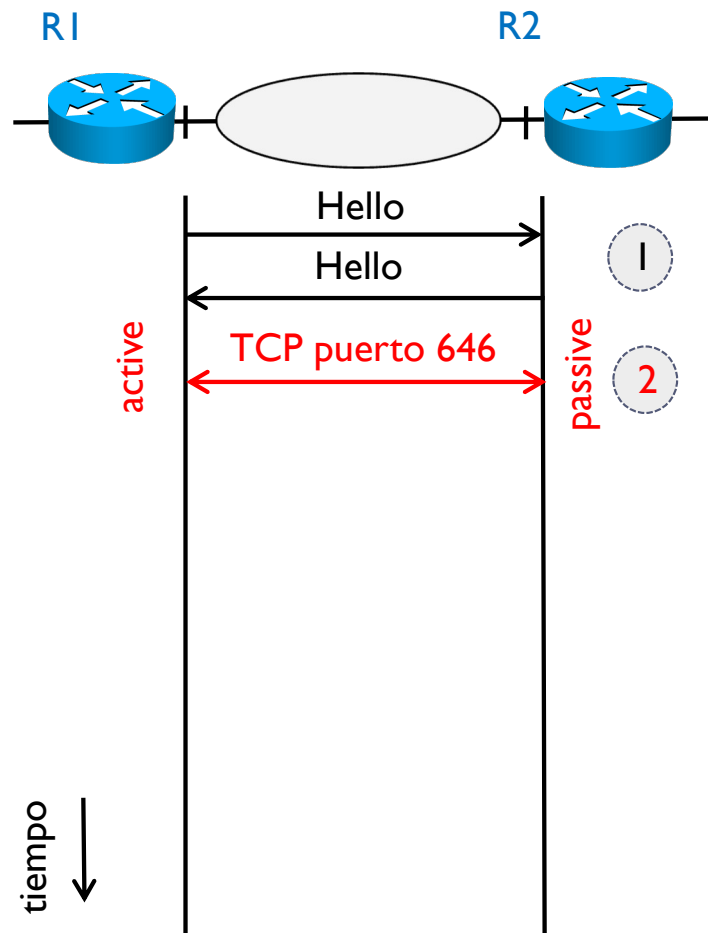
- ▶ LDP asigna e intercambia etiquetas entre LSR adyacentes
- ▶ Una vez la operación de intercambio se completa, se crean las tablas LIB y LFIB



- I. Al activar LDP, R1 y R2 se intercambian un Hello para descubrirse (igual que OSPF)

4.5 - Label Distribution Protocol

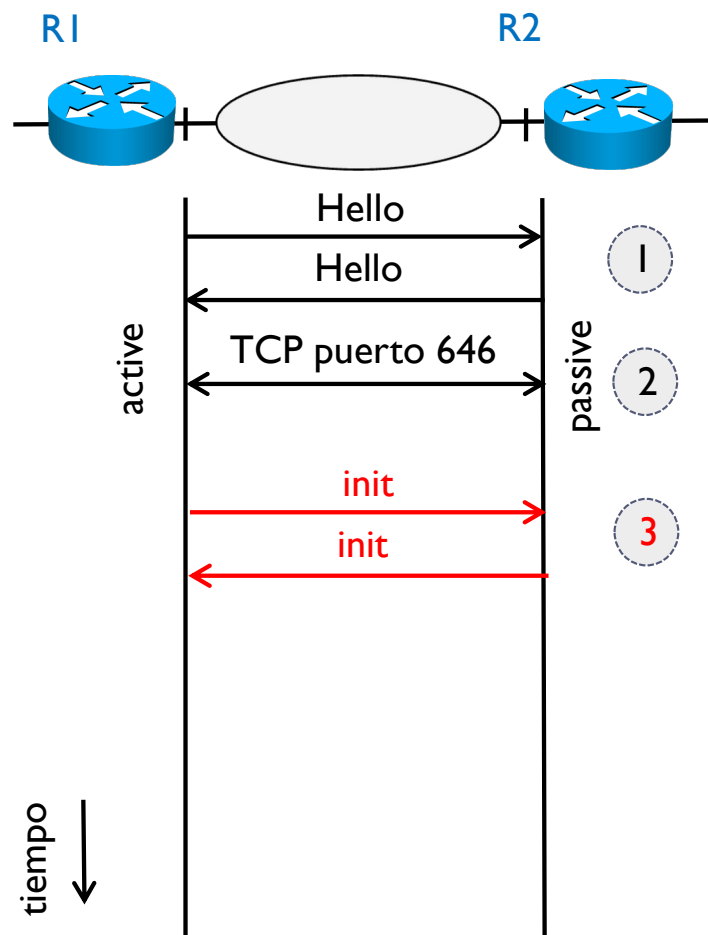
- ▶ LDP asigna e intercambia etiquetas entre LSR adyacentes
- ▶ Una vez la operación de intercambio se completa, se crean las tablas LIB y LFIB



1. Al activar LDP, R1 y R2 se intercambian un Hello para descubrirse (igual que OSPF)
2. El LSR con el RID más alto (active), abre una conexión TCP con puerto 646 (LDP)

4.5 - Label Distribution Protocol

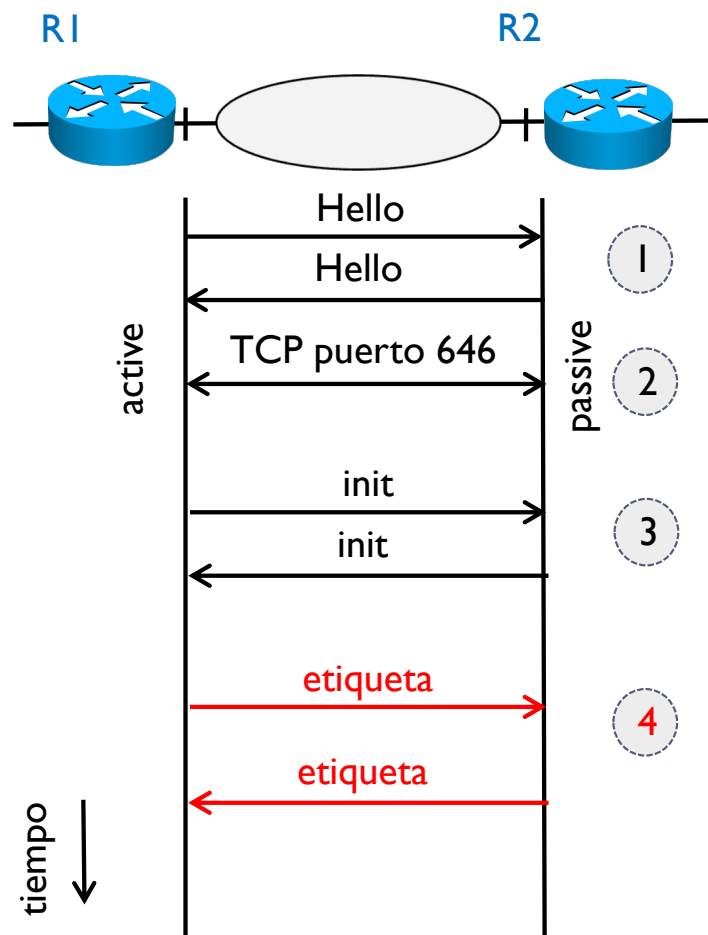
- ▶ LDP asigna e intercambia etiquetas entre LSR adyacentes
- ▶ Una vez la operación de intercambio se completa, se crean las tablas LIB y LFIB



1. Al activar LDP, R1 y R2 se intercambian un Hello para descubrirse (igual que OSPF)
2. El LSR con el RID más alto (active), abre una conexión TCP con puerto 646 (LDP)
3. **El LSR activo envía un mensaje de inicialización con parámetros de configuración de MPLS**
 - ▶ Tiempo de keepalive
 - ▶ MTU
 - ▶ Método de distribución de etiquetas

4.5 - Label Distribution Protocol

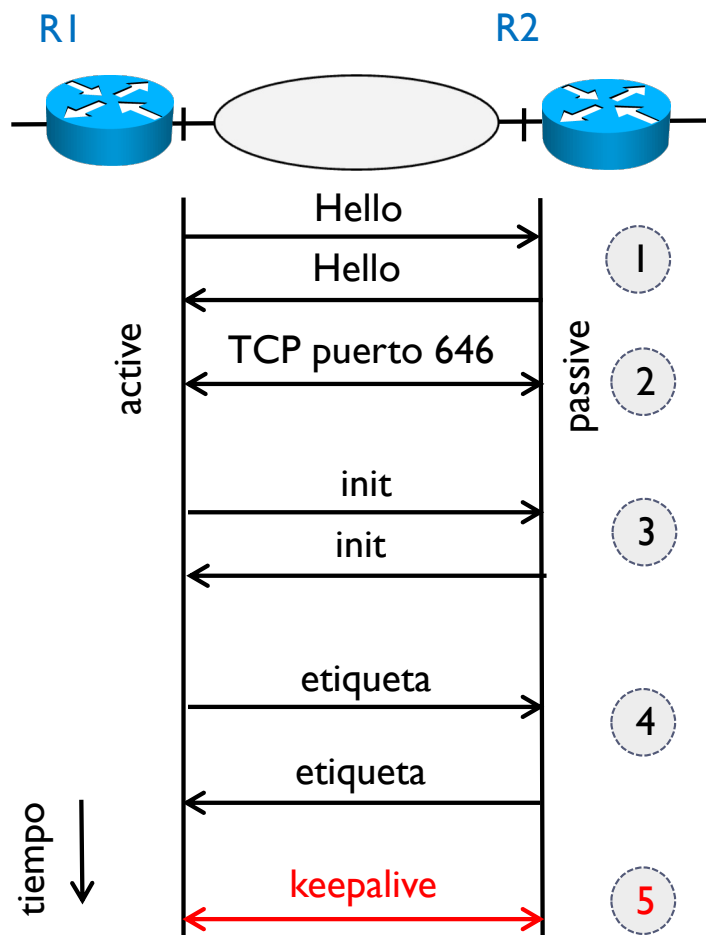
- ▶ LDP asigna e intercambia etiquetas entre LSR adyacentes
- ▶ Una vez la operación de intercambio se completa, se crean las tablas LIB y LFIB



1. Al activar LDP, R1 y R2 se intercambian un Hello para descubrirse (igual que OSPF)
2. El LSR con el RID más alto (active), abre una conexión TCP con puerto 646 (LDP)
3. El LSR activo envía un mensaje de inicialización con parámetros de configuración de MPLS
 - ▶ Tiempo de keepalive
 - ▶ MTU
 - ▶ Método de distribución de etiquetas
4. Cuando se necesita se mapea un prefijo con una etiqueta, en cualquiera de los dos sentidos

4.5 - Label Distribution Protocol

- ▶ LDP asigna e intercambia etiquetas entre LSR adyacentes
- ▶ Una vez la operación de intercambio se completa, se crean las tablas LIB y LFIB



1. Al activar LDP, R1 y R2 se intercambian un Hello para descubrirse (igual que OSPF)
2. El LSR con el RID más alto (active), abre una conexión TCP con puerto 646 (LDP)
3. El LSR activo envía un mensaje de inicialización con parámetros de configuración de MPLS
 - ▶ Tiempo de keepalive
 - ▶ MTU
 - ▶ Método de distribución de etiquetas
4. Cuando se necesita se mapea un prefijo con una etiqueta, en cualquiera de los dos sentidos
5. Cada cierto tiempo se envía un keepalive, para comprobar que la conexión sigue viva

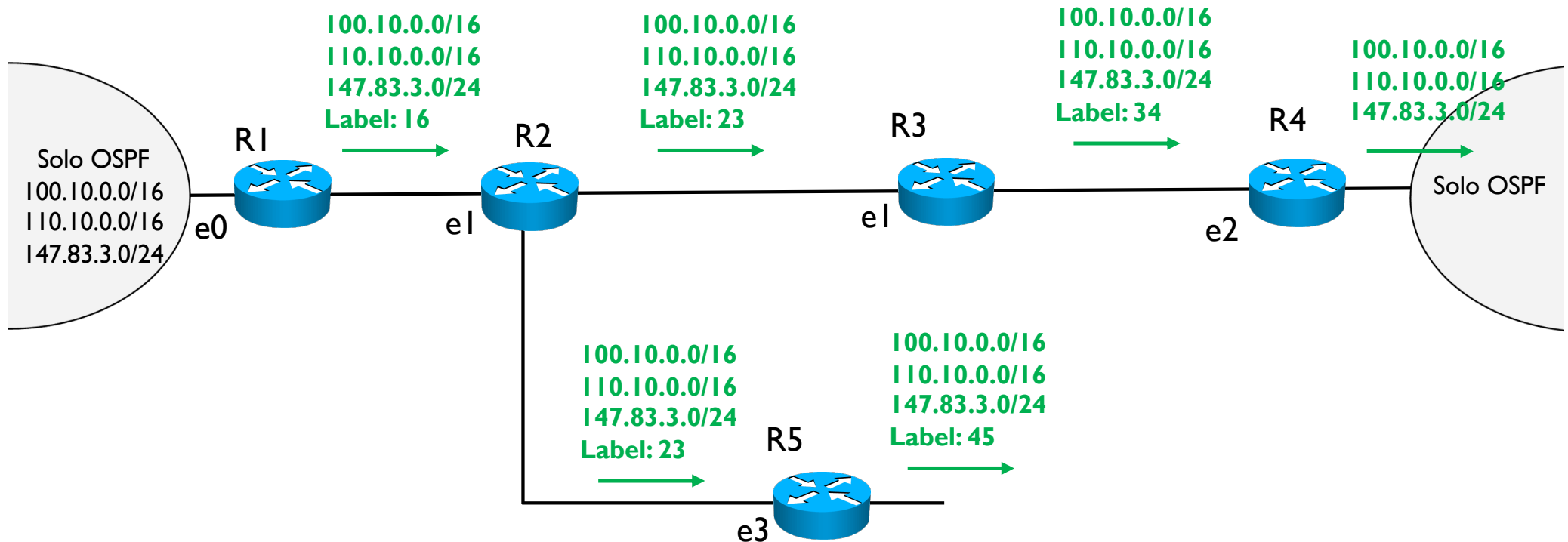
4.5 - Label Distribution Protocol

- ▶ Existen dos métodos de distribución de etiquetas
- ▶ No solicitado
 - ▶ Un LSR distribuye la asociación etiqueta/prefijo en dirección upstream
 - ▶ Dirección contraria al prefijo: es decir del prefijo hacia fuera
 - ▶ Método por defecto
- ▶ Bajo petición
 - ▶ Un LSR pide al vecino en dirección downstream que elija una etiqueta para un determinado prefijo
 - ▶ El otro LSR selecciona una etiqueta para este prefijo y le envía la asociación en dirección upstream
 - ▶ La petición van en dirección hacia el prefijo, la asociación sigue yendo del prefijo hacia fuera

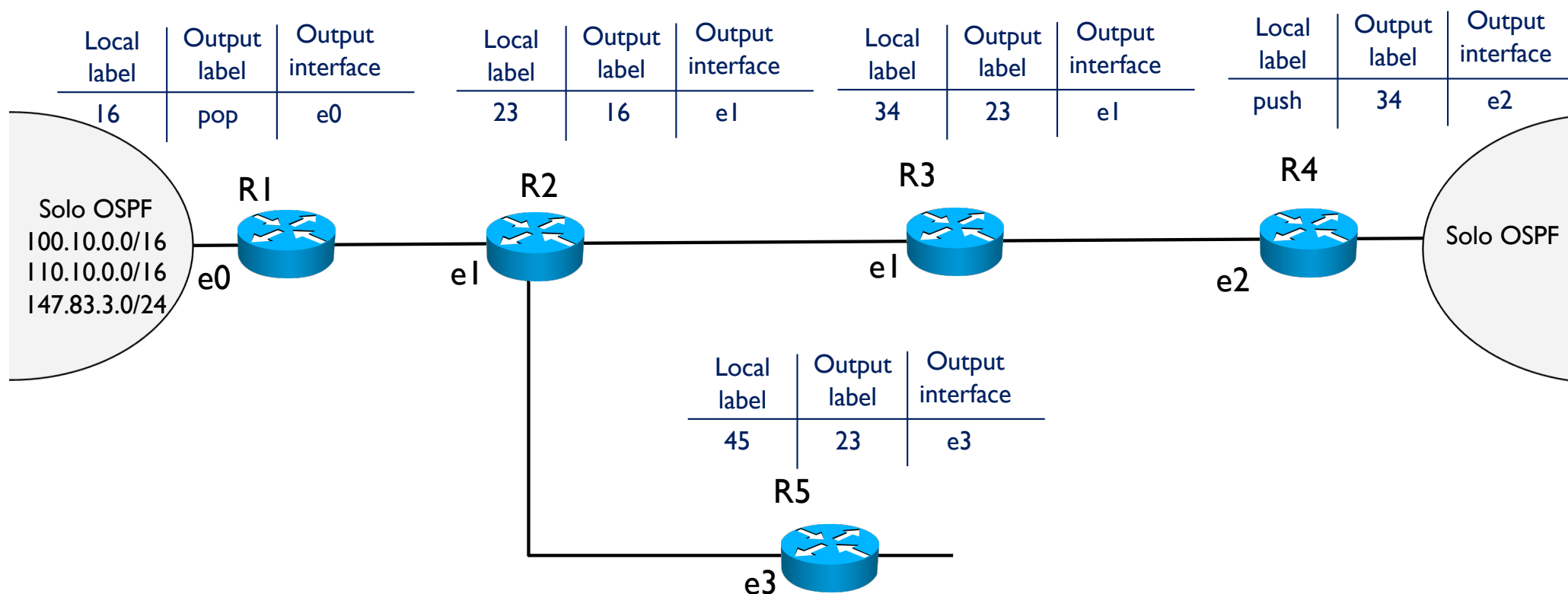
4. MultiProtocol Label Switching (MPLS)

1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. Label Distribution Protocol (LDP)
6. **Ejemplo de funcionamiento**
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. MPLS con extensiones de TE
8. Ejemplo de funcionamiento
9. MPLS fast reroute

4.6 - Ejemplo de funcionamiento

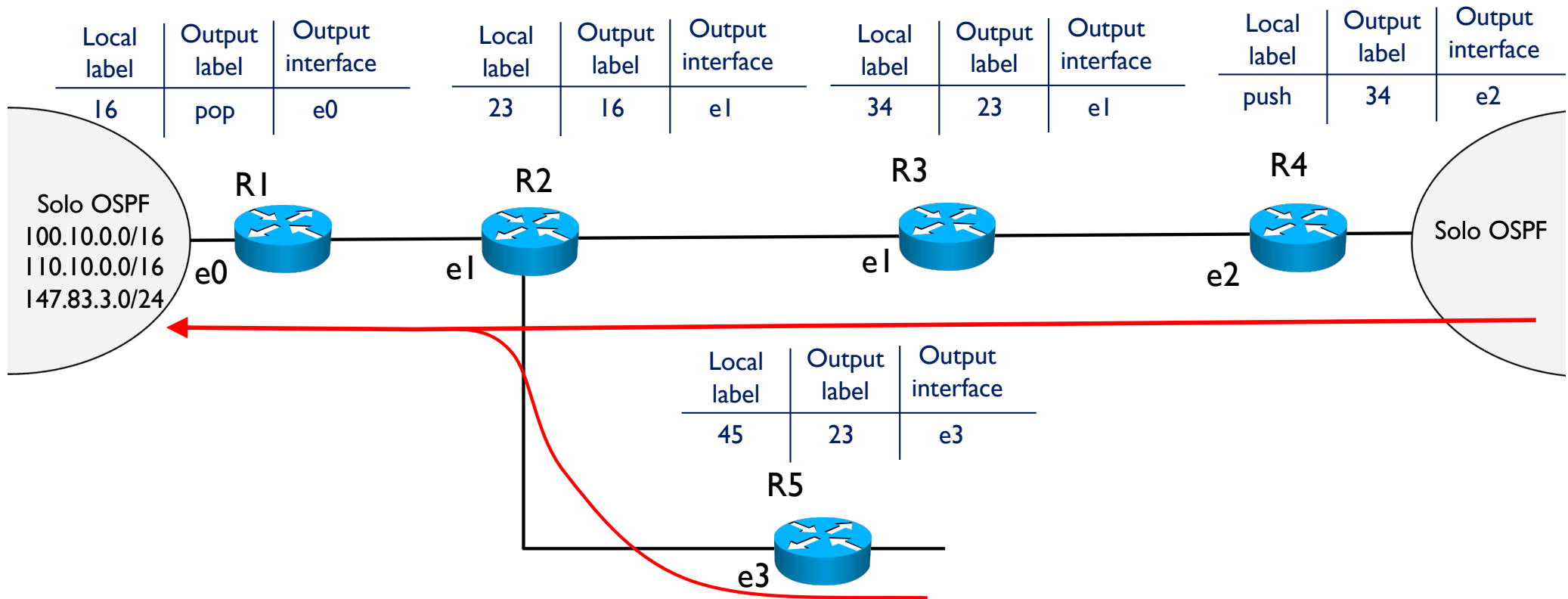


4.6 - Ejemplo de funcionamiento



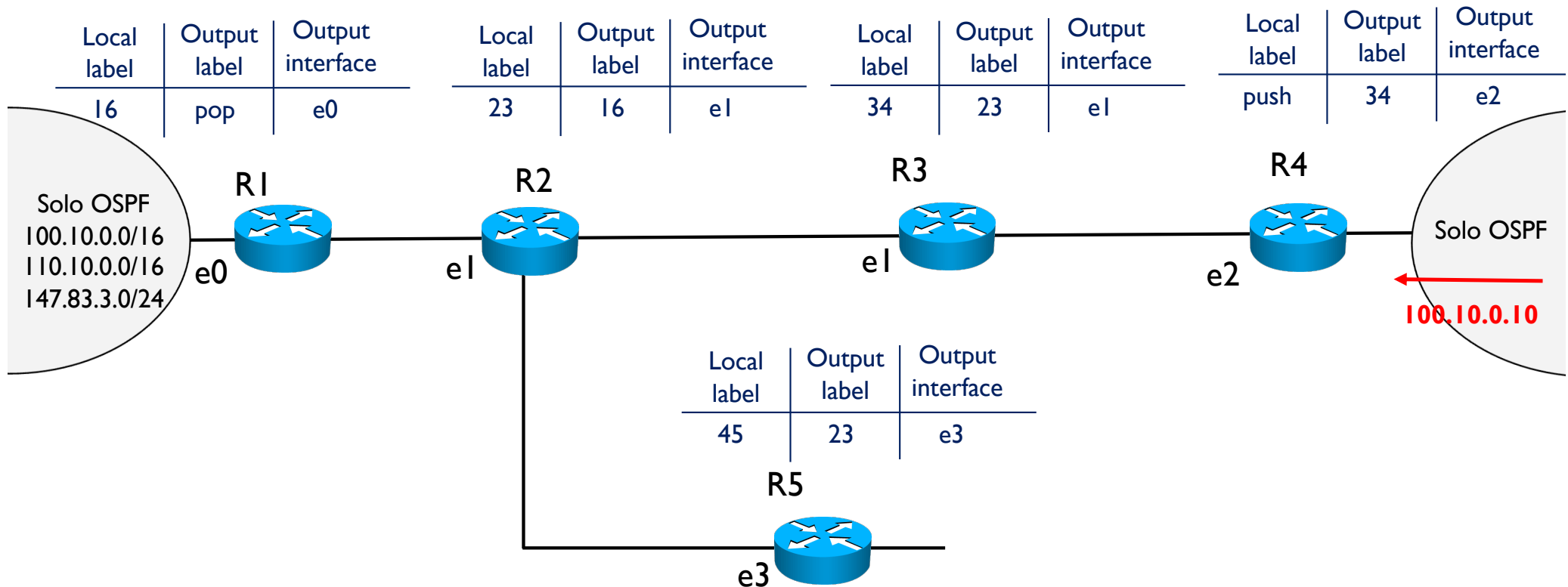
¿Sabrías indicar cual es el LSP?

4.6 - Ejemplo de funcionamiento

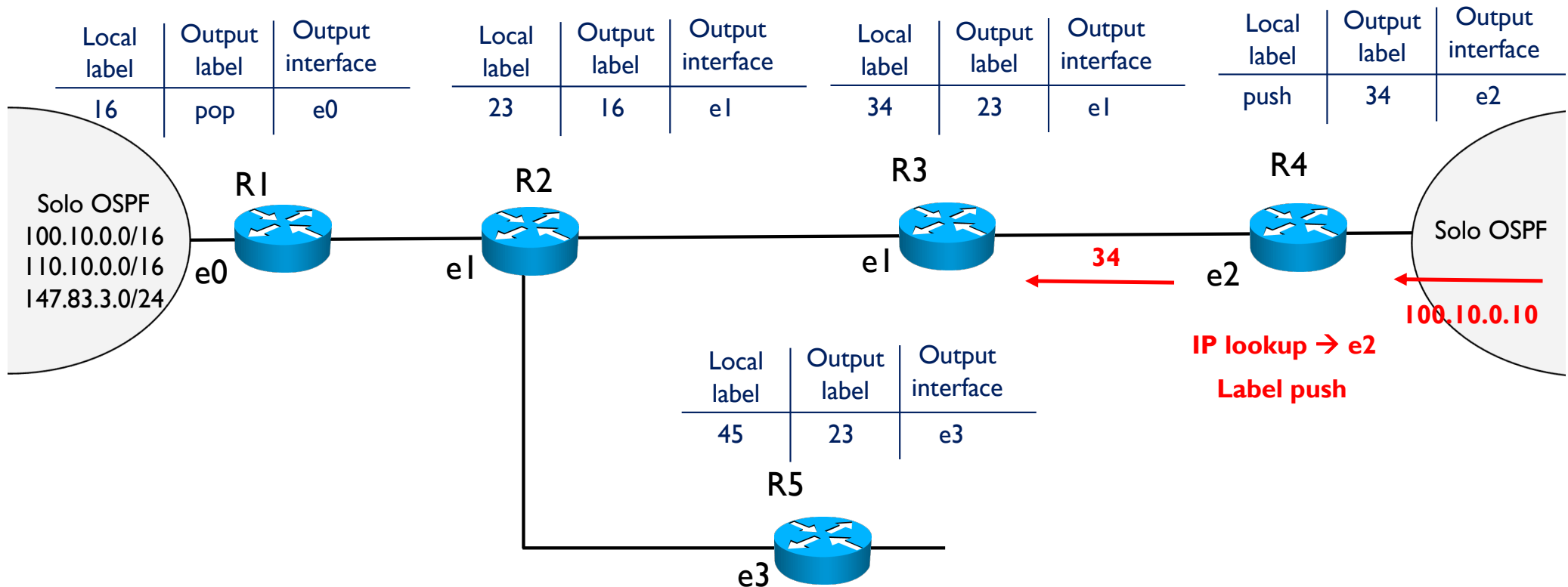


¿Sabrías indicar cual es el LSP?

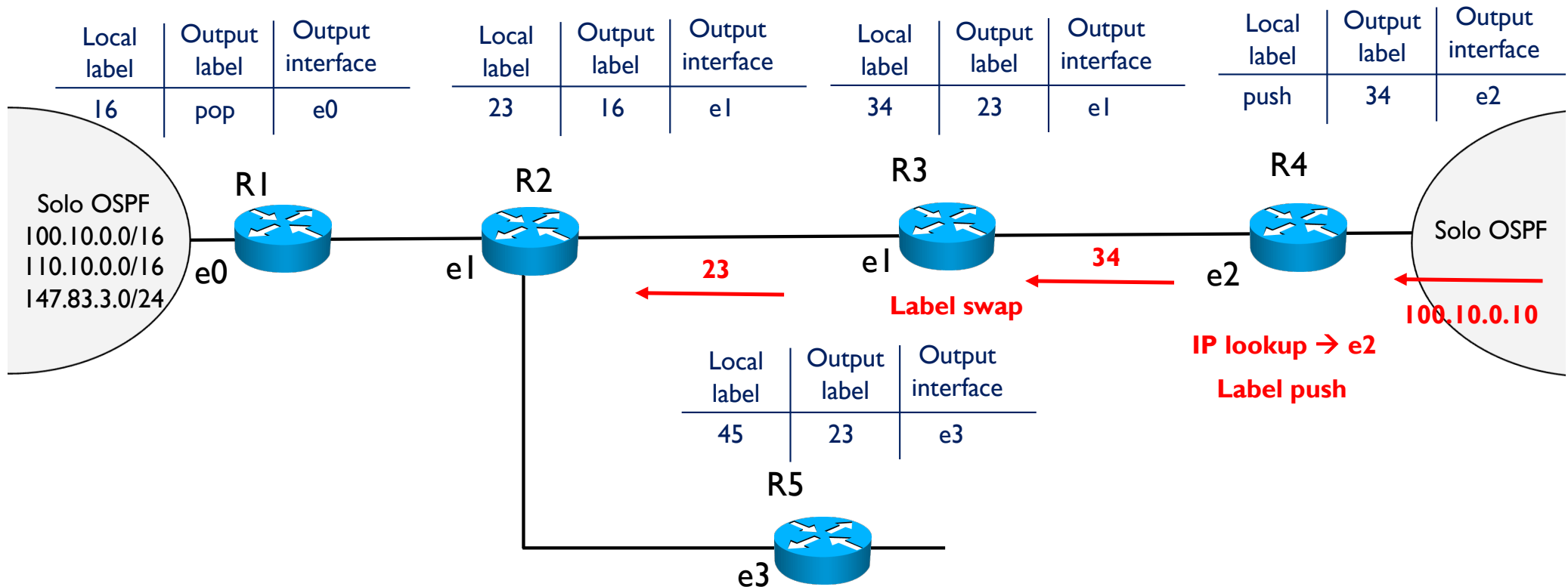
4.6 - Ejemplo de funcionamiento



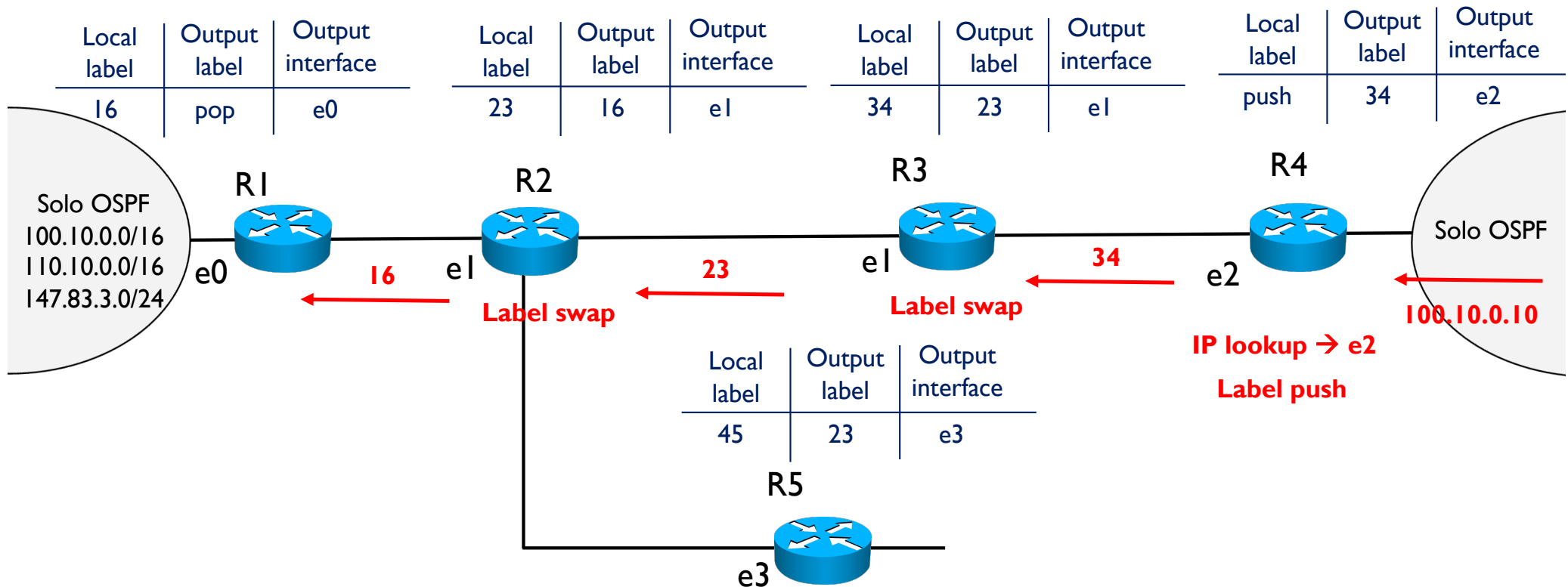
4.6 - Ejemplo de funcionamiento



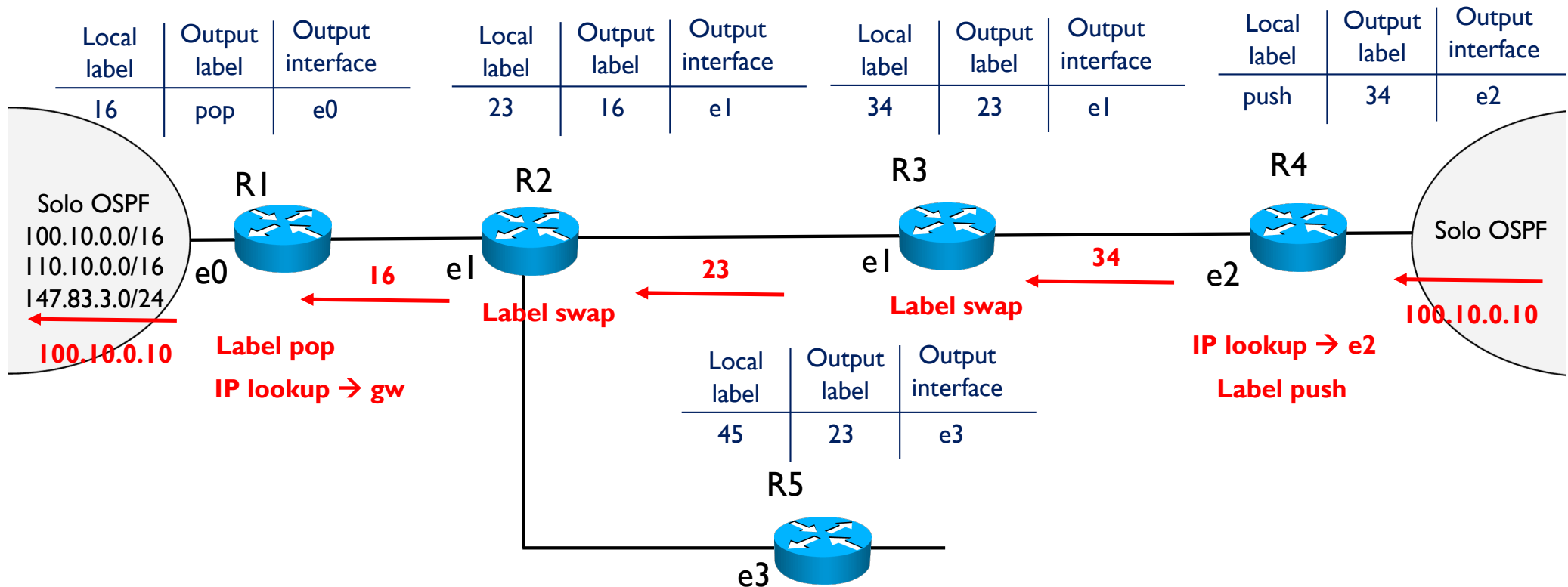
4.6 - Ejemplo de funcionamiento



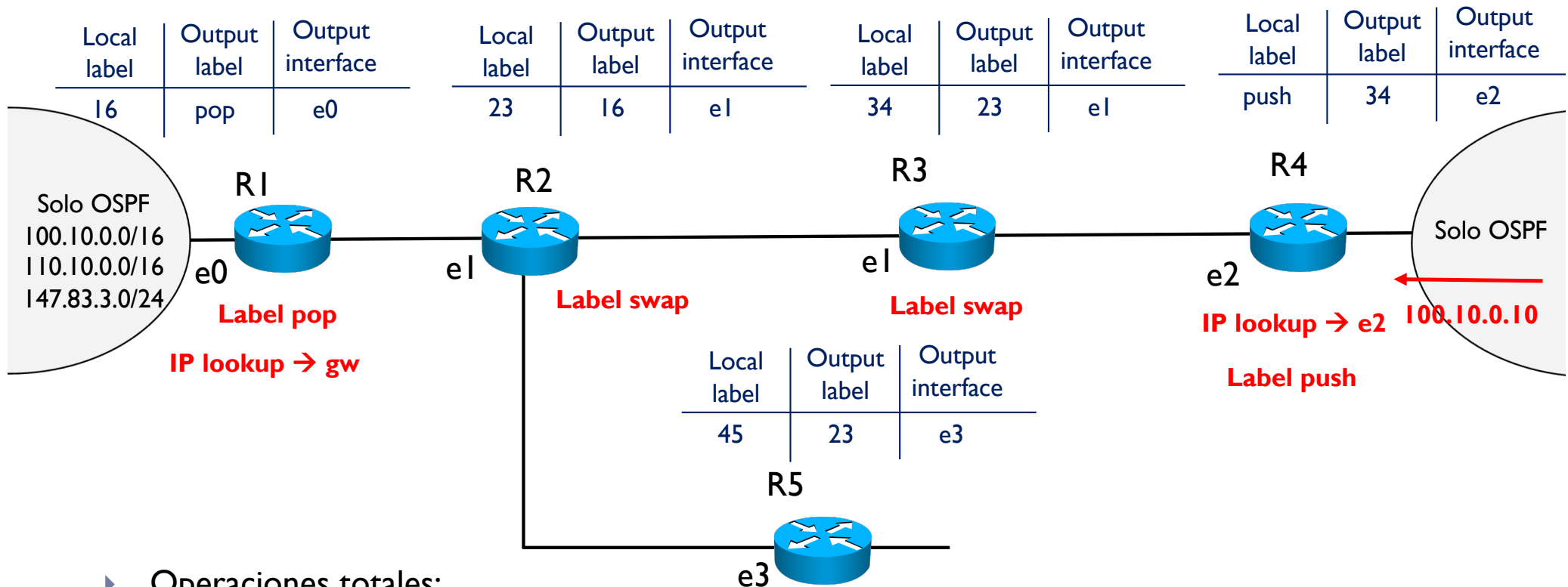
4.6 - Ejemplo de funcionamiento



4.6 - Ejemplo de funcionamiento



4.6 - Ejemplo de funcionamiento

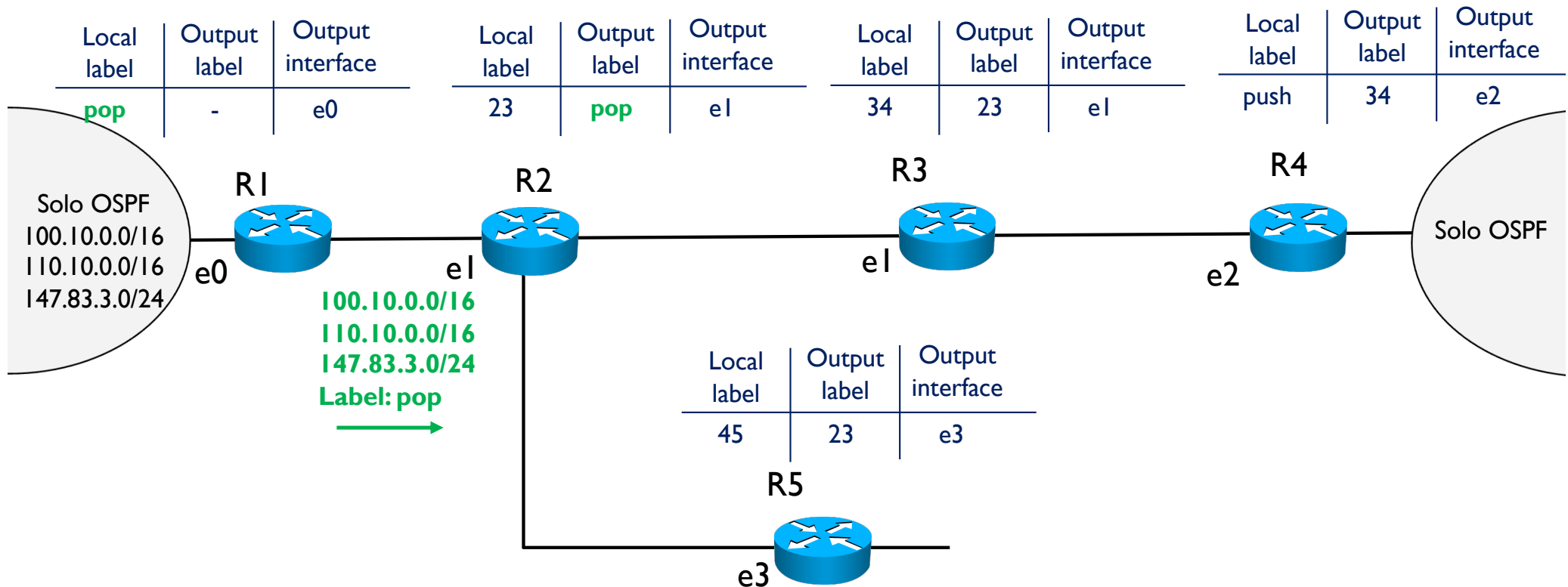


► Operaciones totales:

- 2 IP lookup
- 1 label pop
- 1 label push
- 2 label swap

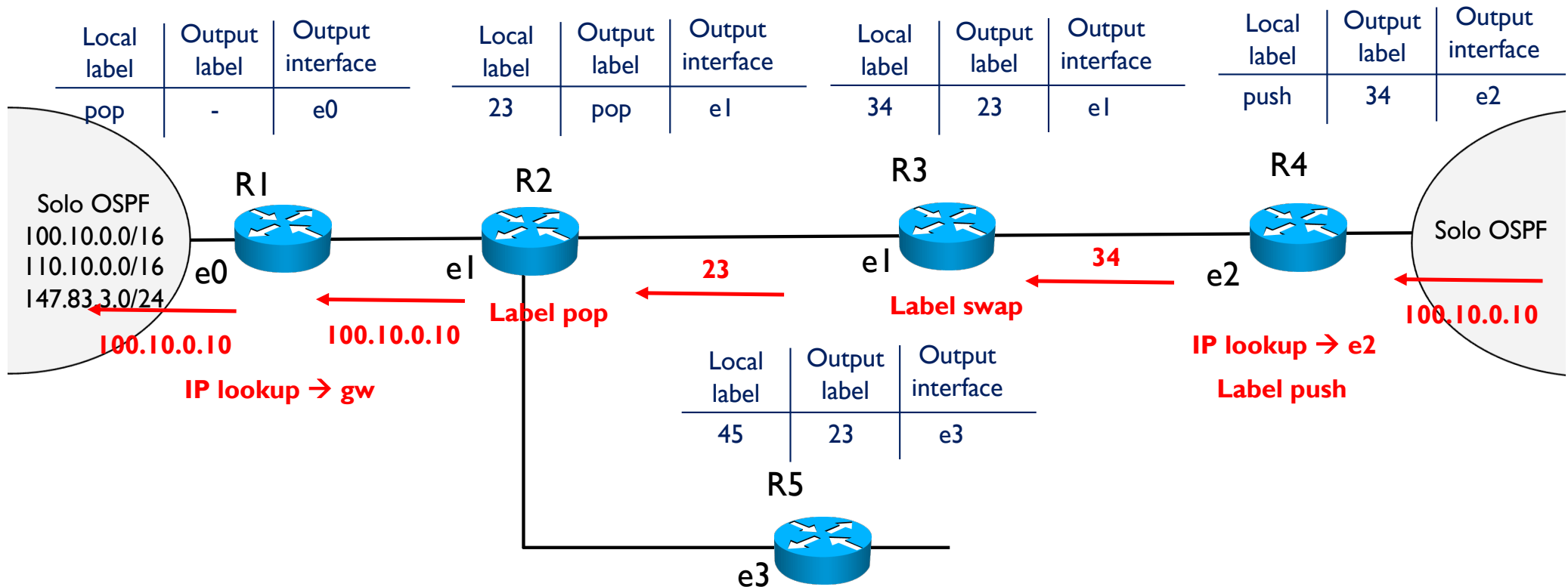
4.6.1 - Ejemplo de funcionamiento

Mejora: Penultimate Hop Popping (PHP)



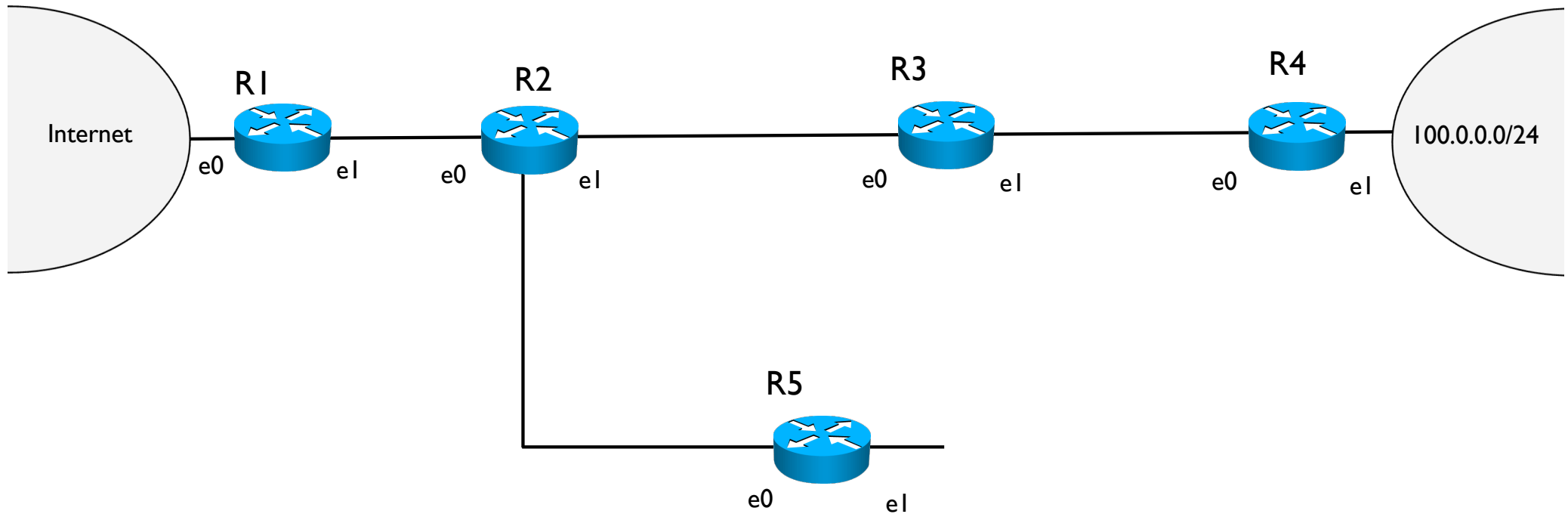
- ▶ Para ahorrarse un label swap, se puede adelantar el pop al penúltimo LSR en lugar de esperar llegar al último
- ▶ MPLS funciona por defecto con PHP

4.6.1 - Penultimate Hop Popping



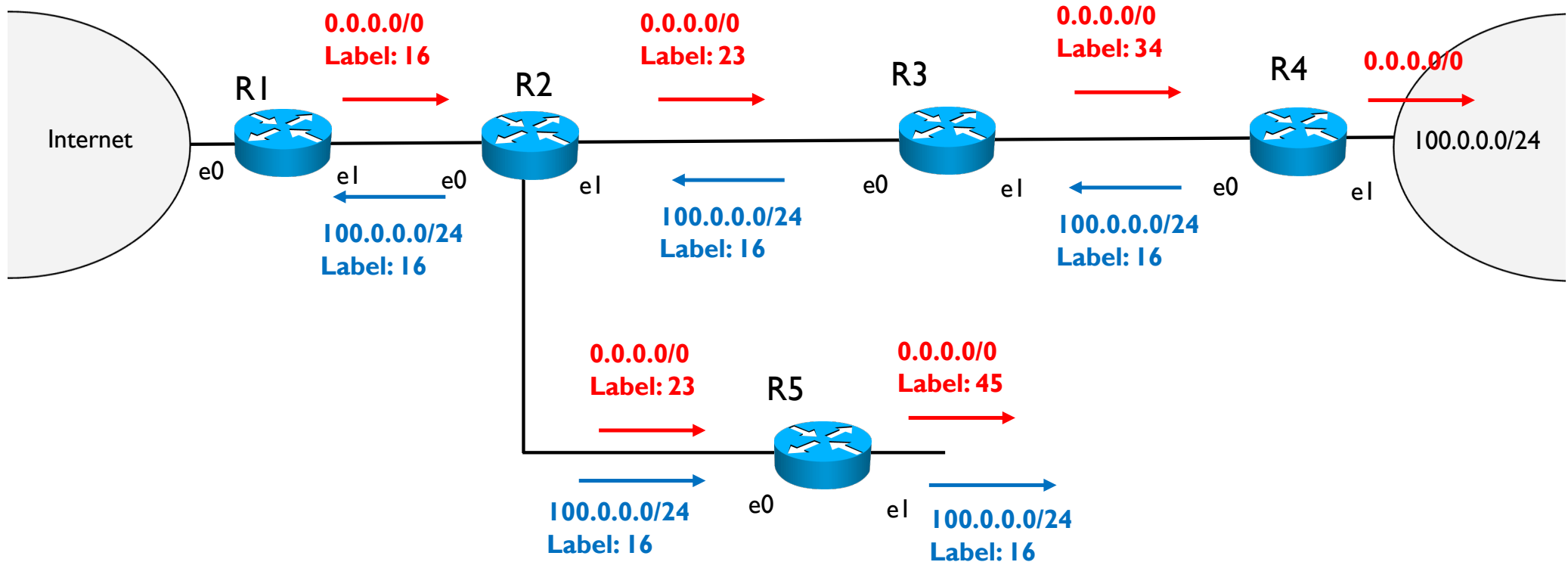
- ▶ Para ahorrarse un label swap, se puede adelantar el pop al penúltimo LSR en lugar de esperar llegar al último
- ▶ MPLS funciona por defecto con PHP

4.6.2 – Otro ejemplo



¿Cómo se envía las etiquetas?

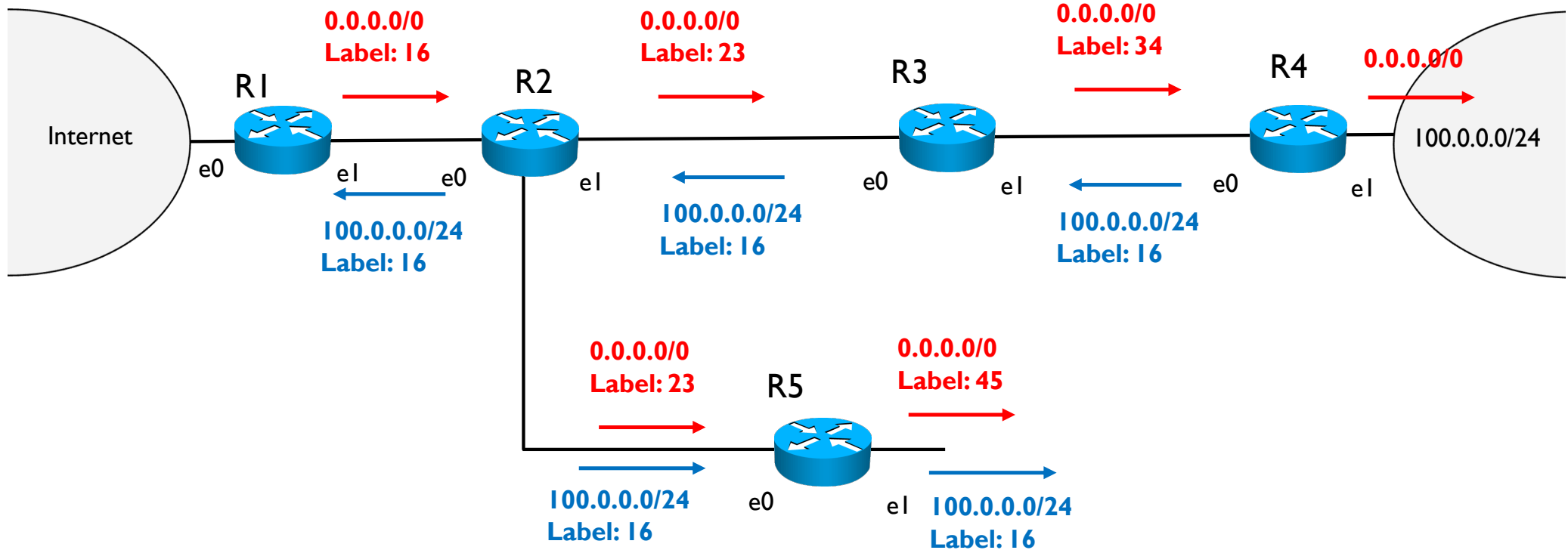
4.6.2 – Otro ejemplo



Siempre en upstream

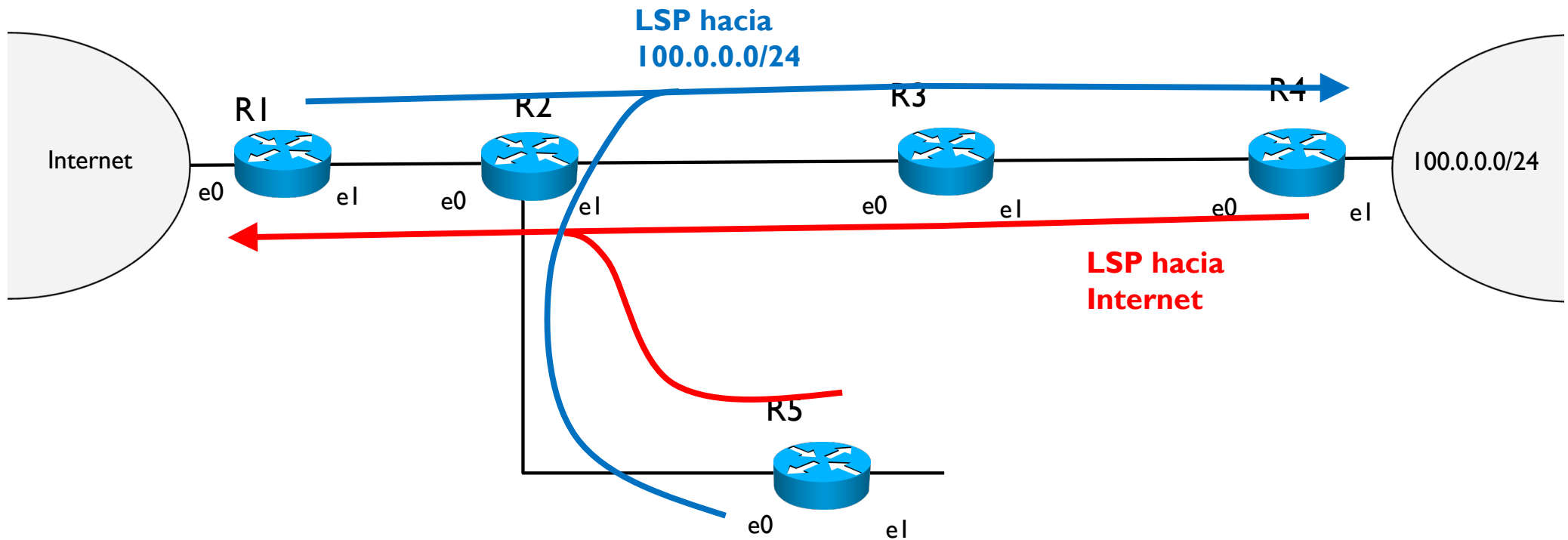
4.6.2 – Otro ejemplo

| Local label | Output label | Output interface | Local label | Output label | Output interface | Local label | Output label | Output interface | Local label | Output label | Output interface |
|-------------|--------------|------------------|-------------|--------------|------------------|-------------|--------------|------------------|-------------|--------------|------------------|
| 16 | pop | e0 | 23 | 16 | e0 | 34 | 23 | e0 | push | 34 | e0 |
| push | 16 | e1 | 16 | 16 | e1 | 16 | 16 | e1 | 16 | pop | e1 |



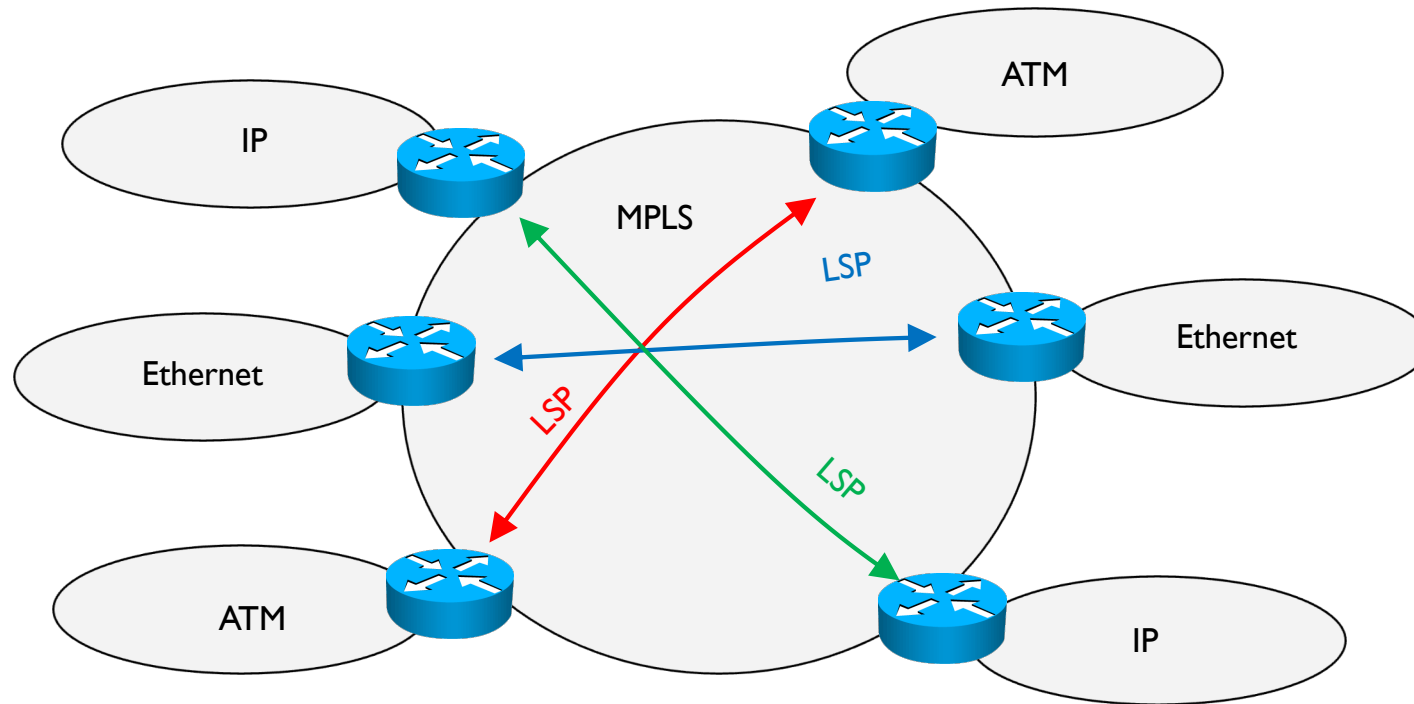
¿Cuáles serán los LSPs?

4.6.2 – Otro ejemplo



Siempre en downstream

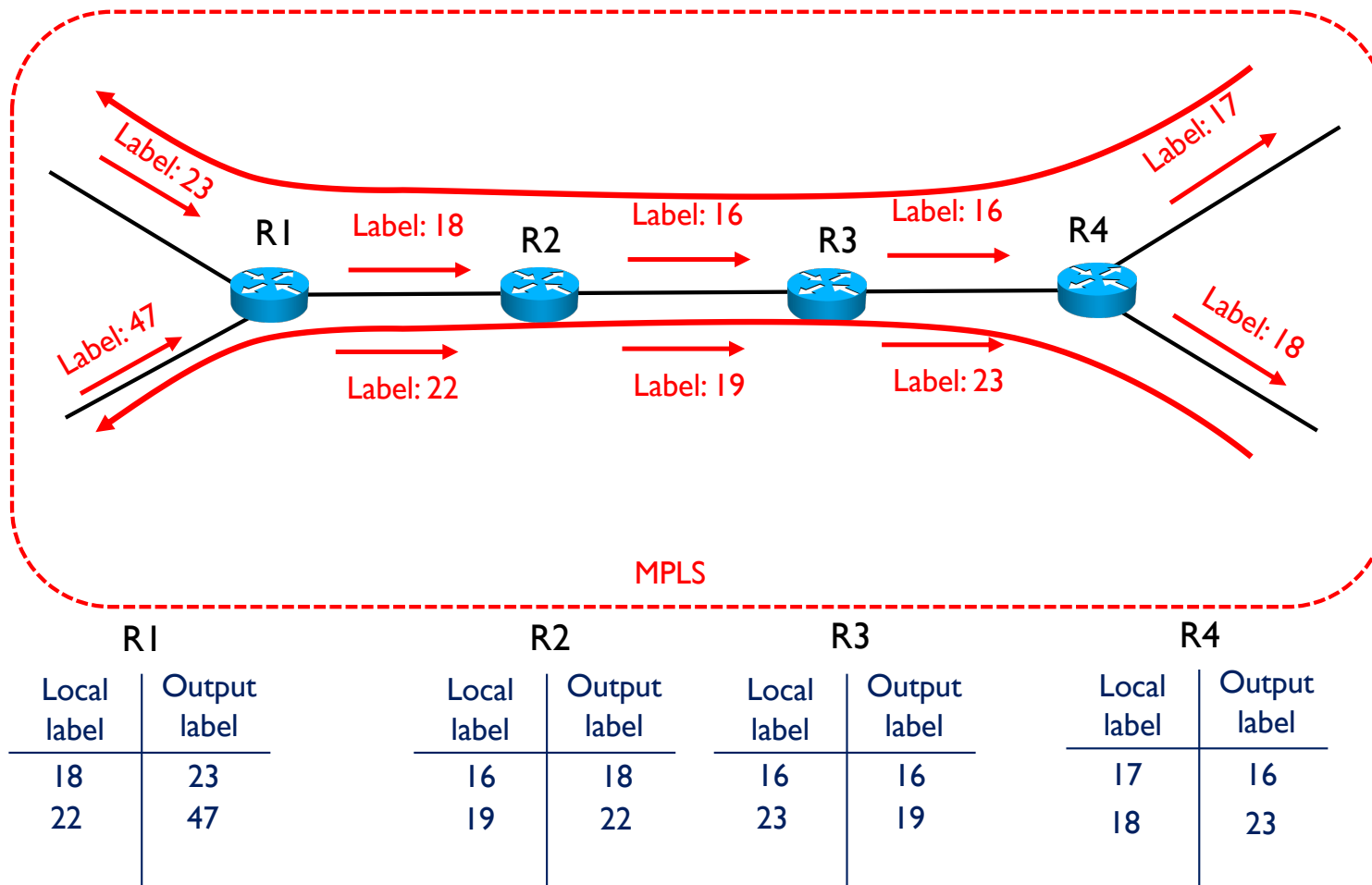
4.6.2 - Resultado final con MPLS



- ▶ Los LSP se pueden ver como túneles, donde los paquetes están “escondidos” dentro de MPLS
- ▶ Da soporte a la construcción de VPN
- ▶ Es multiprotocolo: permite otras tecnologías que no sean IP

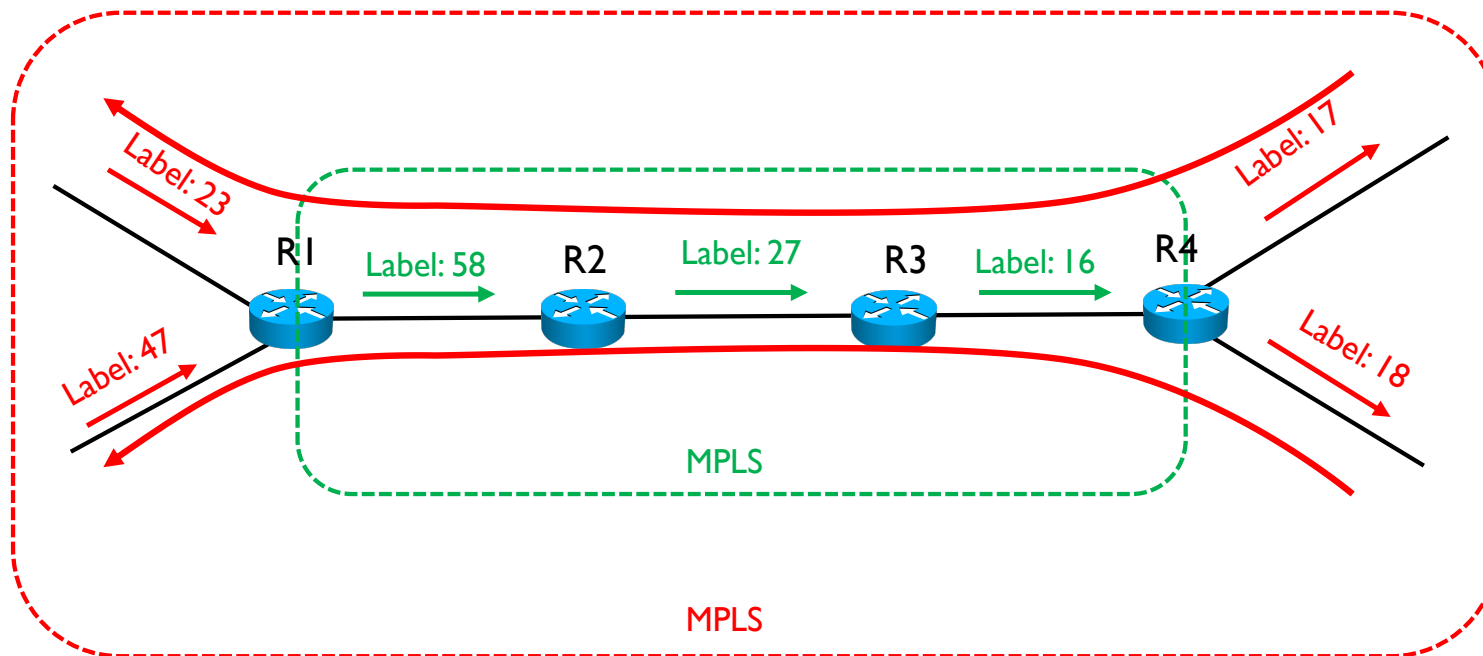
4.6.3 Label stack

- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP



4.6.3 Label stack

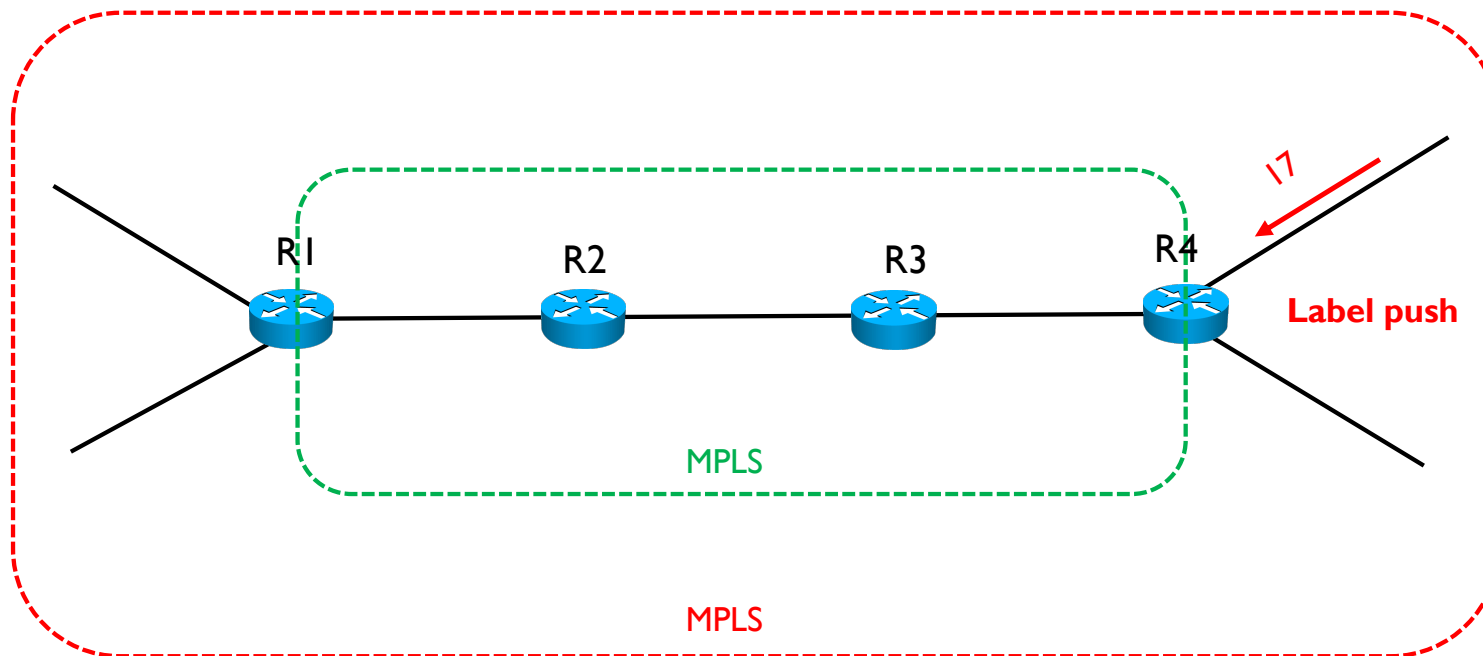
- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP



| R1 | | R2 | | R3 | | R4 | |
|-------------|--------------|-------------|--------------|-------------|--------------|-------------|--------------|
| Local label | Output label | Local label | Output label | Local label | Output label | Local label | Output label |
| 58 | pop | 27 | 58 | 16 | 27 | 17 | push |
| 17 | 23 | | | | | 18 | push |
| 18 | 47 | | | | | push | 16 |

4.6.3 Label stack

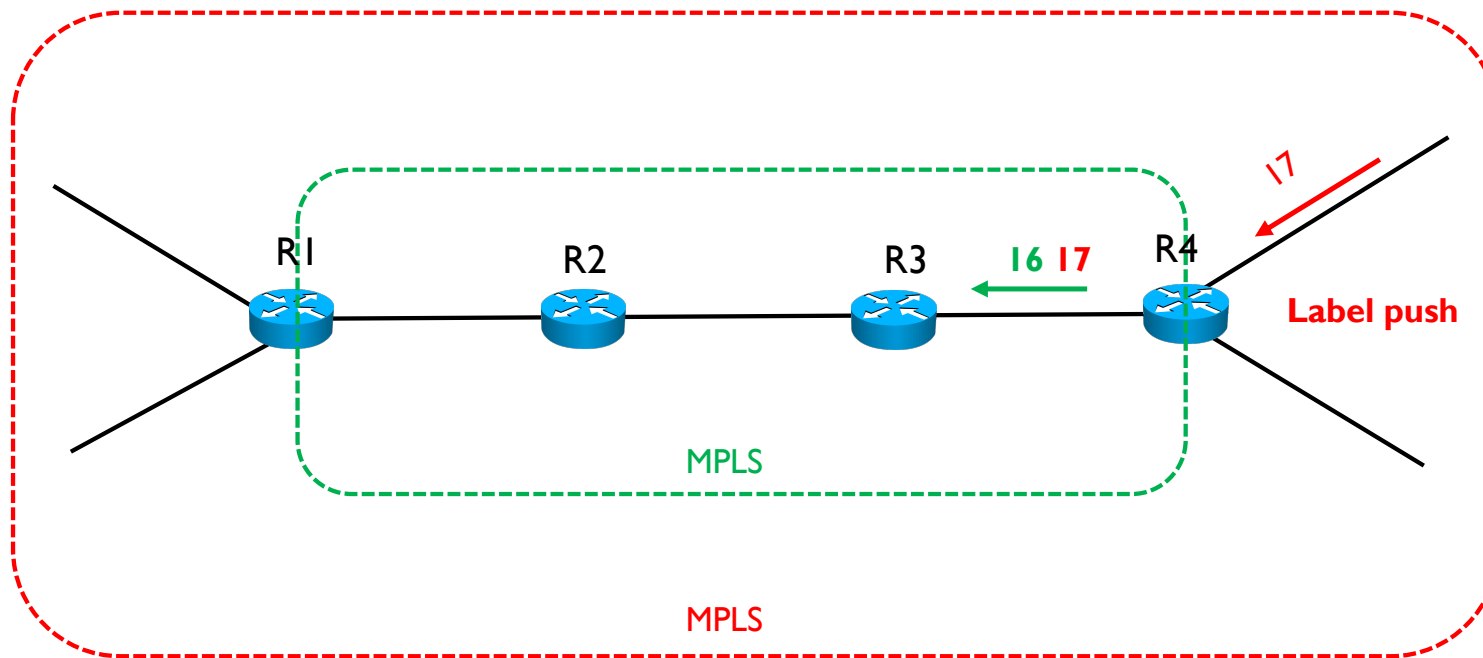
- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP



| R1 | | R2 | | R3 | | R4 | |
|-------------|--------------|-------------|--------------|-------------|--------------|-------------|--------------|
| Local label | Output label | Local label | Output label | Local label | Output label | Local label | Output label |
| 58 | pop | 27 | 58 | 16 | 27 | 17 | push |
| 17 | 23 | | | | | 18 | push |
| 18 | 47 | | | | | push | 16 |

4.6.3 Label stack

- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP

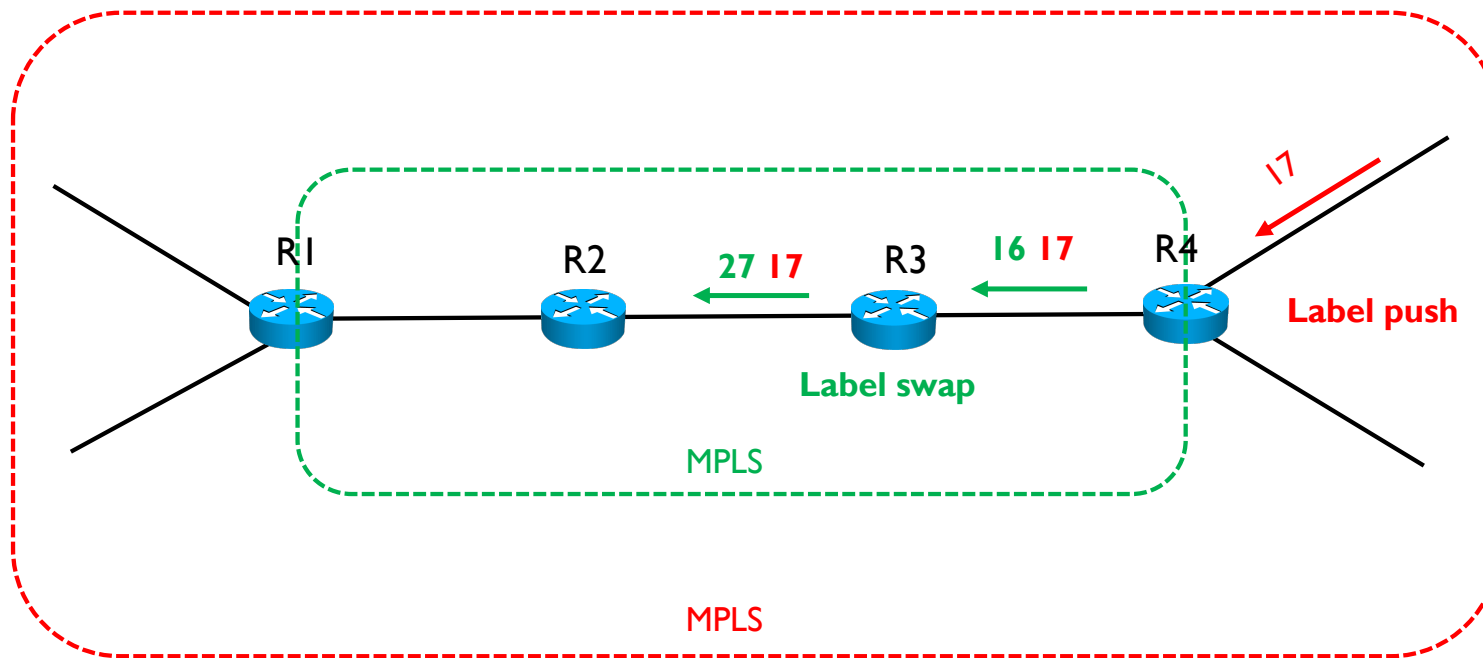


| R1 | | R2 | | R3 | | R4 | |
|-------------|--------------|-------------|--------------|-------------|--------------|-------------|--------------|
| Local label | Output label | Local label | Output label | Local label | Output label | Local label | Output label |
| 58 | pop | 27 | 58 | 16 | 27 | 17 | push |
| 17 | 23 | | | | | 18 | push |
| 18 | 47 | | | | | push | 16 |



4.6.3 Label stack

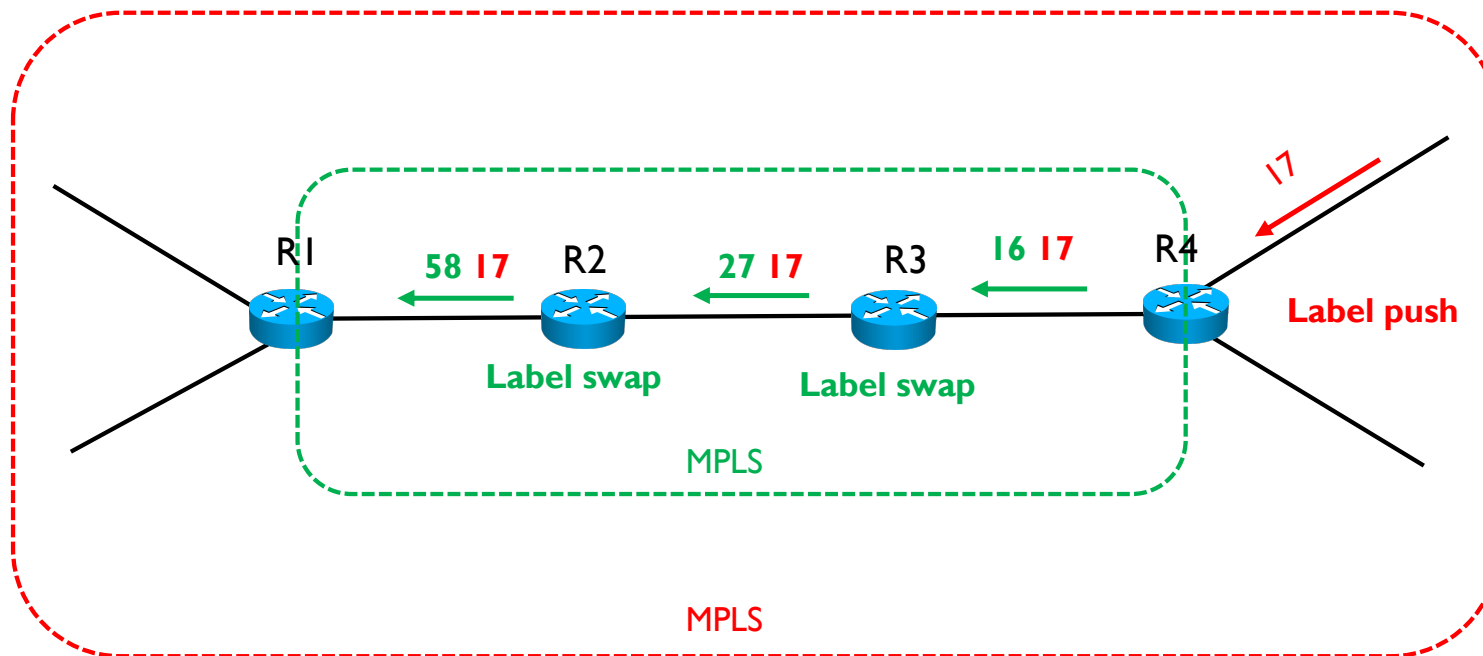
- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP



| R1 | | R2 | | R3 | | R4 | |
|-------------|--------------|-------------|--------------|-------------|--------------|-------------|--------------|
| Local label | Output label | Local label | Output label | Local label | Output label | Local label | Output label |
| 58 | pop | 27 | 58 | 16 | 27 | 17 | push |
| 17 | 23 | | | | | 18 | push |
| 18 | 47 | | | | | push | 16 |

4.6.3 Label stack

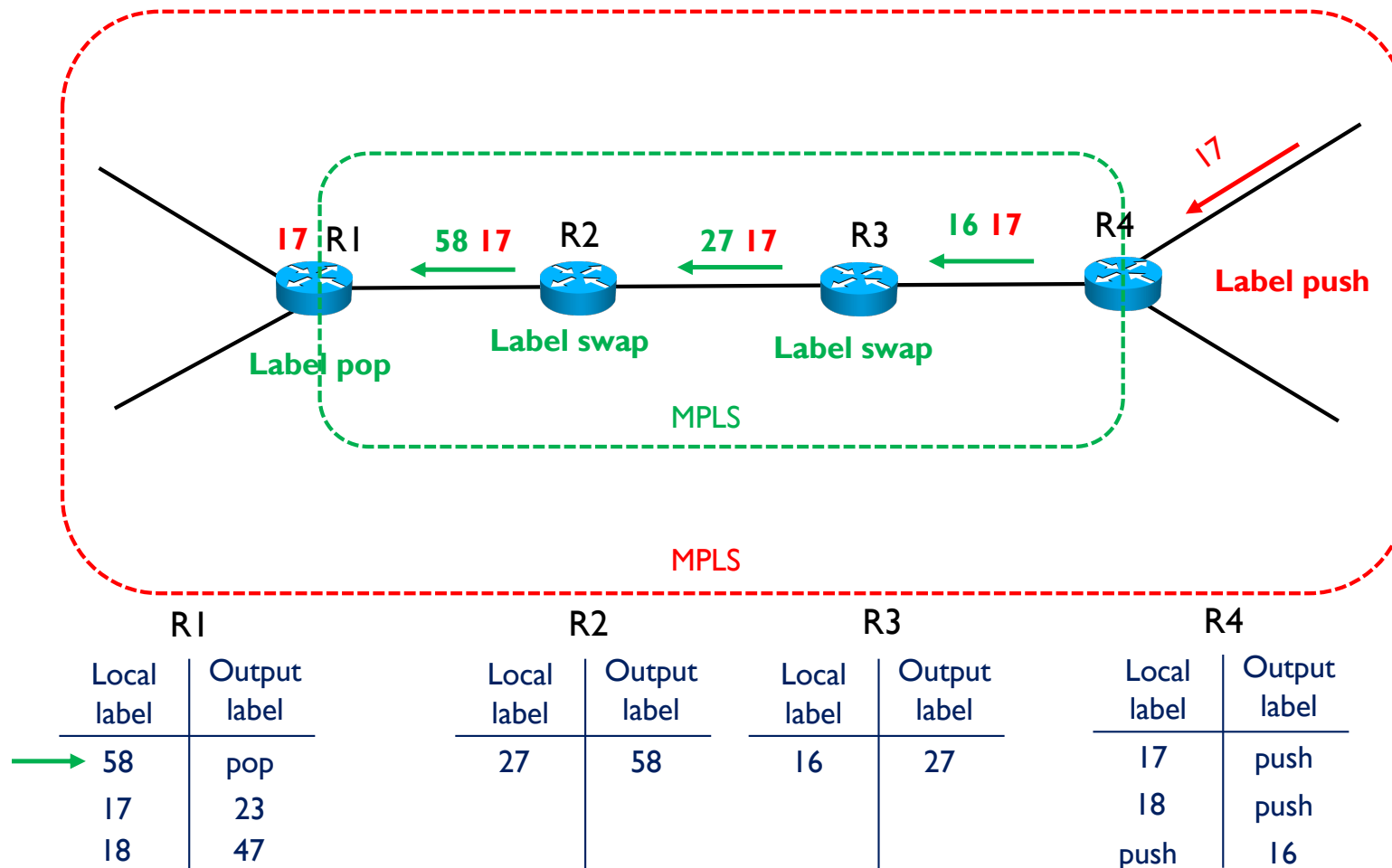
- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP



| R1 | | R2 | | R3 | | R4 | |
|-------------|--------------|-------------|--------------|-------------|--------------|-------------|--------------|
| Local label | Output label | Local label | Output label | Local label | Output label | Local label | Output label |
| 58 | pop | 27 | 58 | 16 | 27 | 17 | push |
| 17 | 23 | | | | | 18 | push |
| 18 | 47 | | | | | push | 16 |

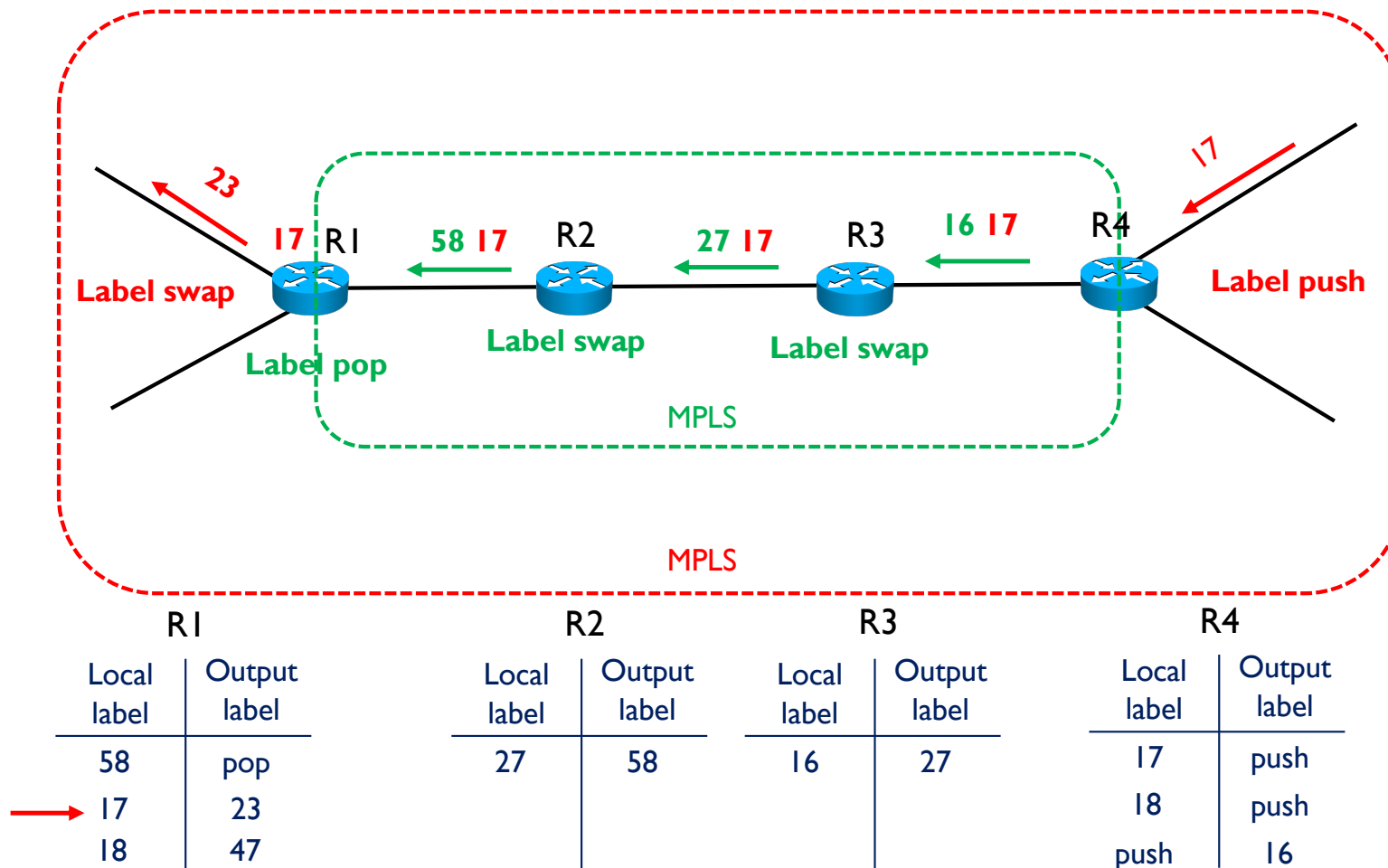
4.6.3 Label stack

- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP



4.6.3 Label stack

- ▶ Una red MPLS puede encapsular otra red MPLS
- ▶ Agregación de caminos LSP



4. MultiProtocol Label Switching (MPLS)

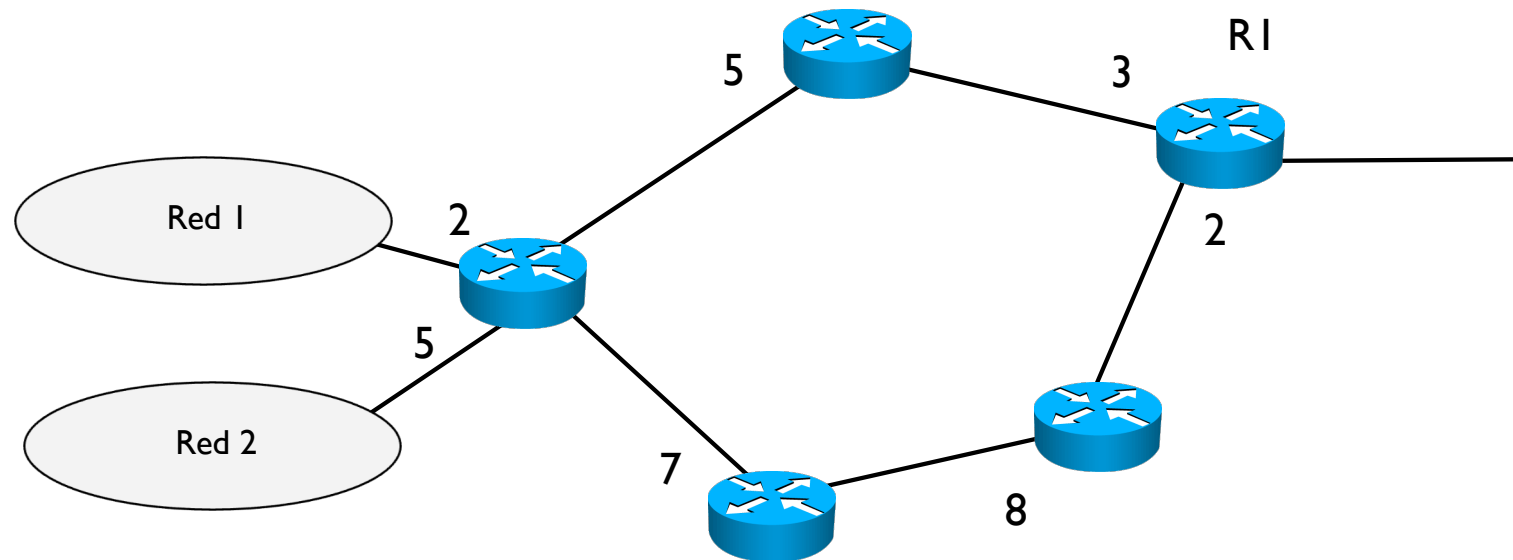
1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. Label Distribution Protocol (LDP)
6. Ejemplo de funcionamiento
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. **MPLS con extensiones de TE**
8. Ejemplo de funcionamiento
9. MPLS fast reroute

4.7 – MPLS con extensiones TE

Funcionamiento hasta ahora

► Infraestructura con coste OSPF

- Llega una petición de crear un camino para llegar de RI a la red I
- RI sabe que la mejor ruta con estos costes es

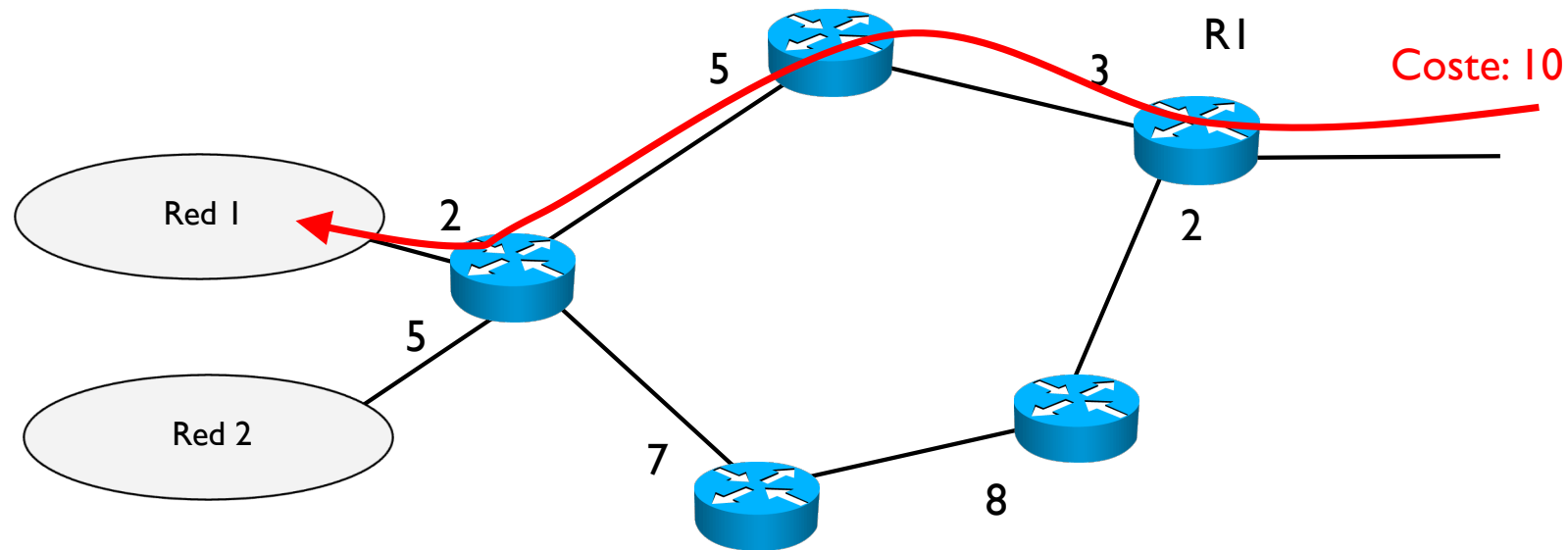


4.7 – MPLS con extensiones TE

Funcionamiento hasta ahora

► Infraestructura con coste OSPF

- Llega una petición de crear un camino para llegar de RI a la red 1
- RI sabe que la mejor ruta con estos costes es

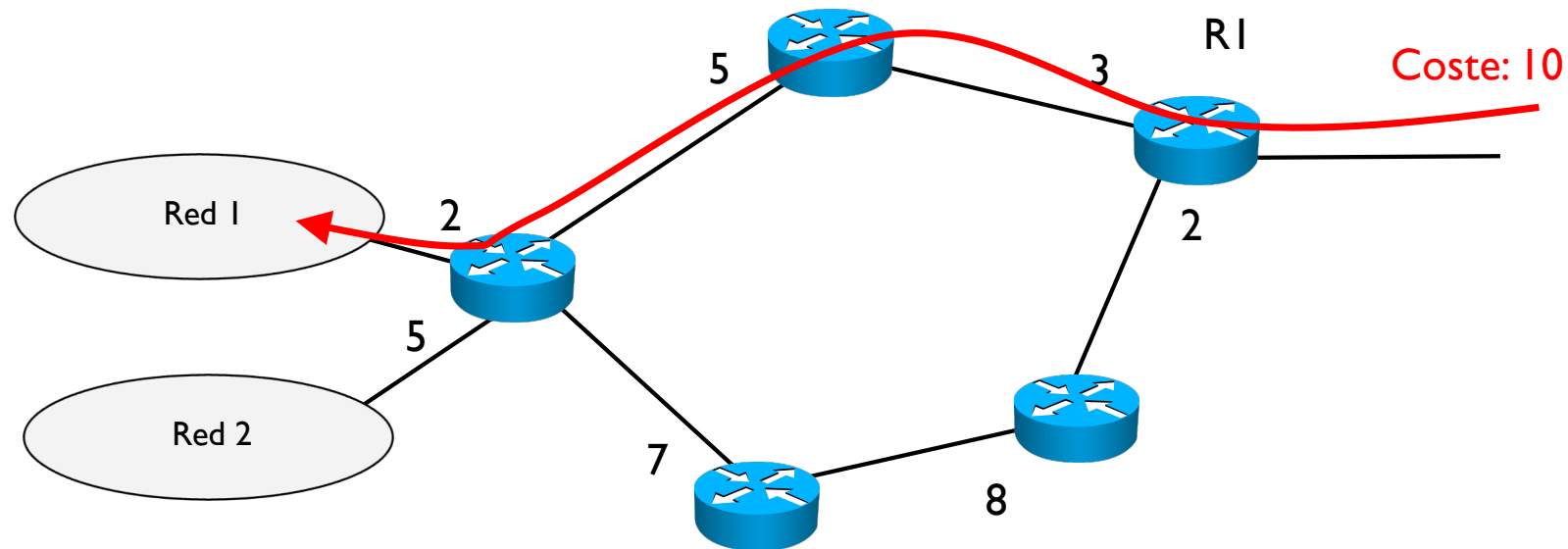


4.7 – MPLS con extensiones TE

Funcionamiento hasta ahora

► Infraestructura con coste OSPF

- Llega una petición de crear un camino para llegar de RI a la red 1
- RI sabe que la mejor ruta con estos costes es
- Ahora llega otra petición para un camino de RI a la red 2

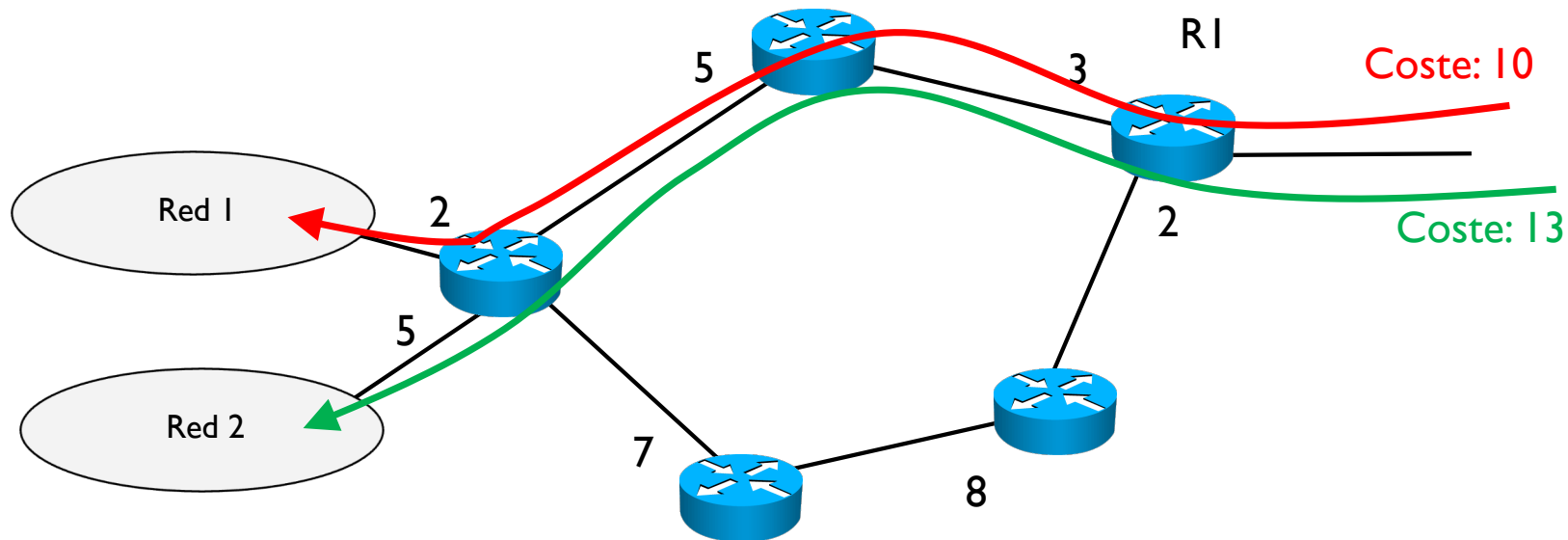


4.7 – MPLS con extensiones TE

Funcionamiento hasta ahora

► Infraestructura con coste OSPF

- Llega una petición de crear un camino para llegar de RI a la red 1
- RI sabe que la mejor ruta con estos costes es
- Ahora llega otra petición para un camino de RI a la red 2: la mejor ruta sigue siendo la que pasa por los mismos routers

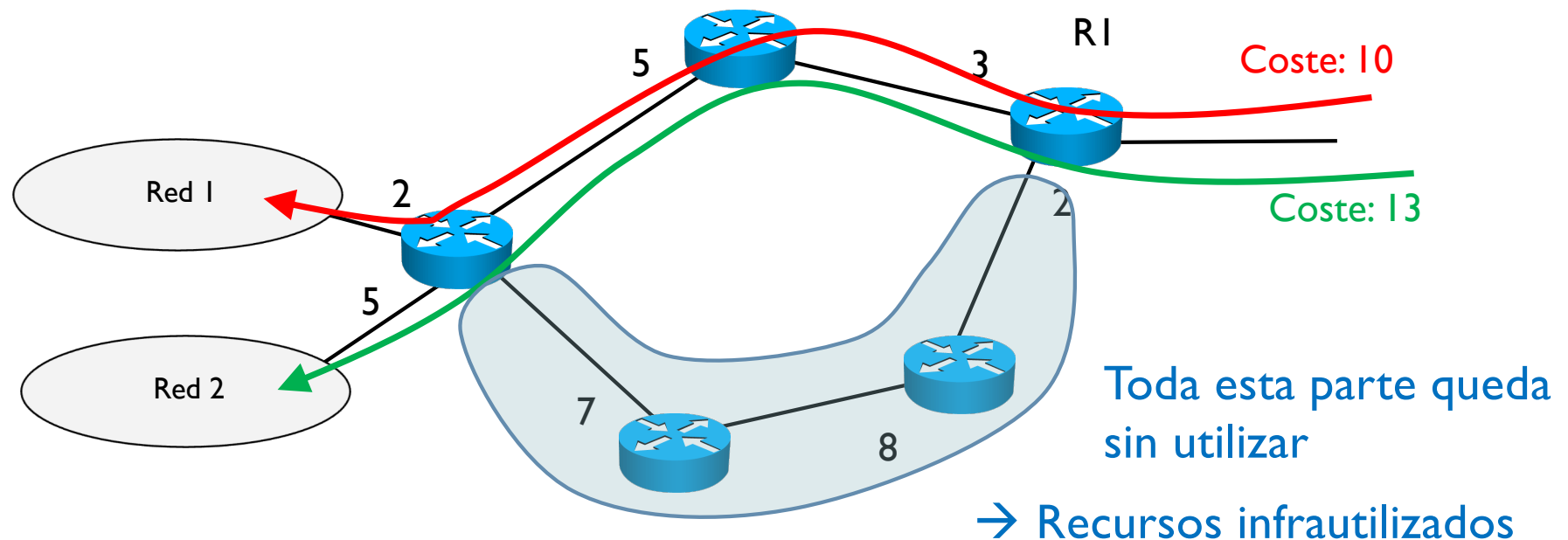


4.7 – MPLS con extensiones TE

Funcionamiento hasta ahora

► Infraestructura con coste OSPF

- Llega una petición de crear un camino para llegar de RI a la red 1
- RI sabe que la mejor ruta con estos costes es
- Ahora llega otra petición para un camino de RI a la red 2: la mejor ruta sigue siendo la que pasa por lo mismos routers



4.7 – MPLS con extensiones TE

Traffic Engineering

- ▶ Extensiones de Ingeniería de Tráfico
- ▶ Idea general
 - ▶ Optimizar el uso de los recursos de la infraestructura
- ▶ Objetivo
 - ▶ Moviendo el tráfico donde hay recursos
 - ▶ Creando espacio para más tráfico (más clientes, más beneficios)
- ▶ ¿Como conseguir eso?

4.7 – MPLS con extensiones TE

Traffic Engineering

- ▶ Extensiones de Ingeniería de Tráfico
- ▶ Idea general
 - ▶ Optimizar el uso de los recursos de la infraestructura
- ▶ Objetivo
 - ▶ Moviendo el tráfico donde hay recursos
 - ▶ Creando espacio para más tráfico (más clientes, más beneficios)
- ▶ ¿Como conseguir eso?
 - ▶ Métricas dinámicas

4.7 – MPLS con extensiones TE

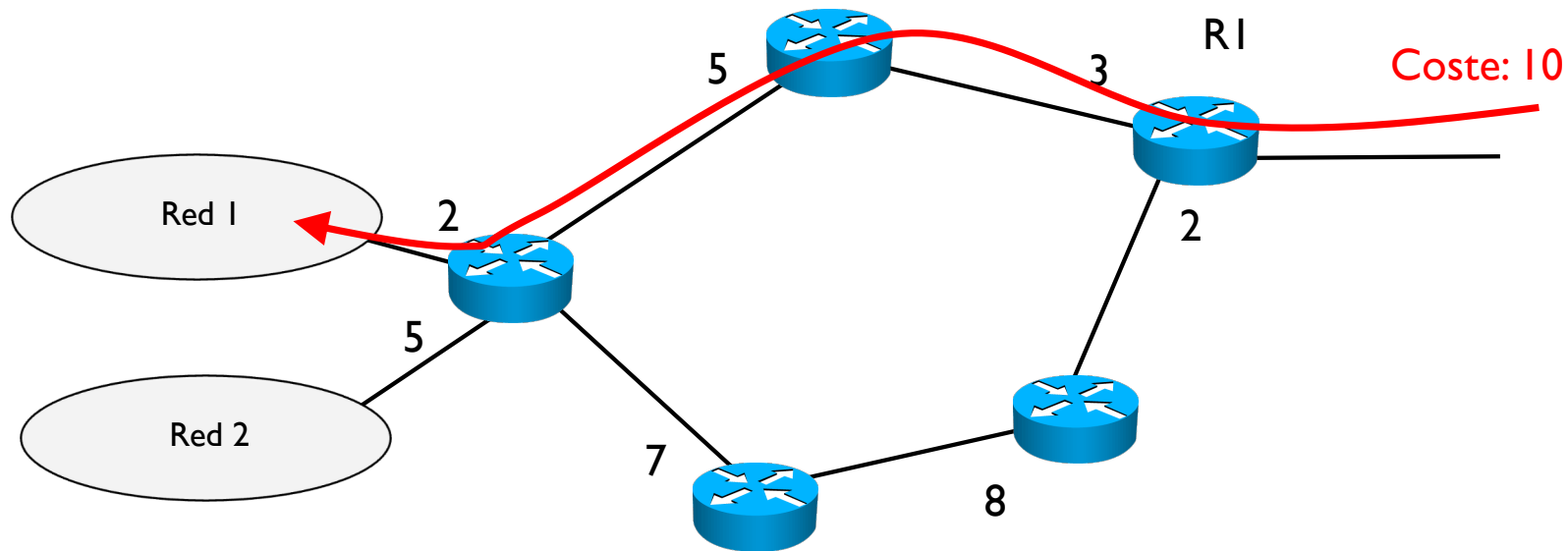
Traffic Engineering

- ▶ Además no tienen porque indicar solo una métrica
- ▶ Se permite el uso de cualquier tipo de métrica, por ejemplo
 - ▶ Ancho de banda restante
 - ▶ Latencia
 - ▶ Disponibilidad
 - ▶ Probabilidad de pérdida de paquetes
 - ▶ Variación del retardo entre paquetes
 - ▶ Etc.
- ▶ Incluido una combinación de ellos en una función pesada

$$f(coste) = w_1 \times C_1 + w_2 \times C_2 + \dots + w_n \times C_n$$

4.7 – MPLS con extensiones TE Traffic Engineering

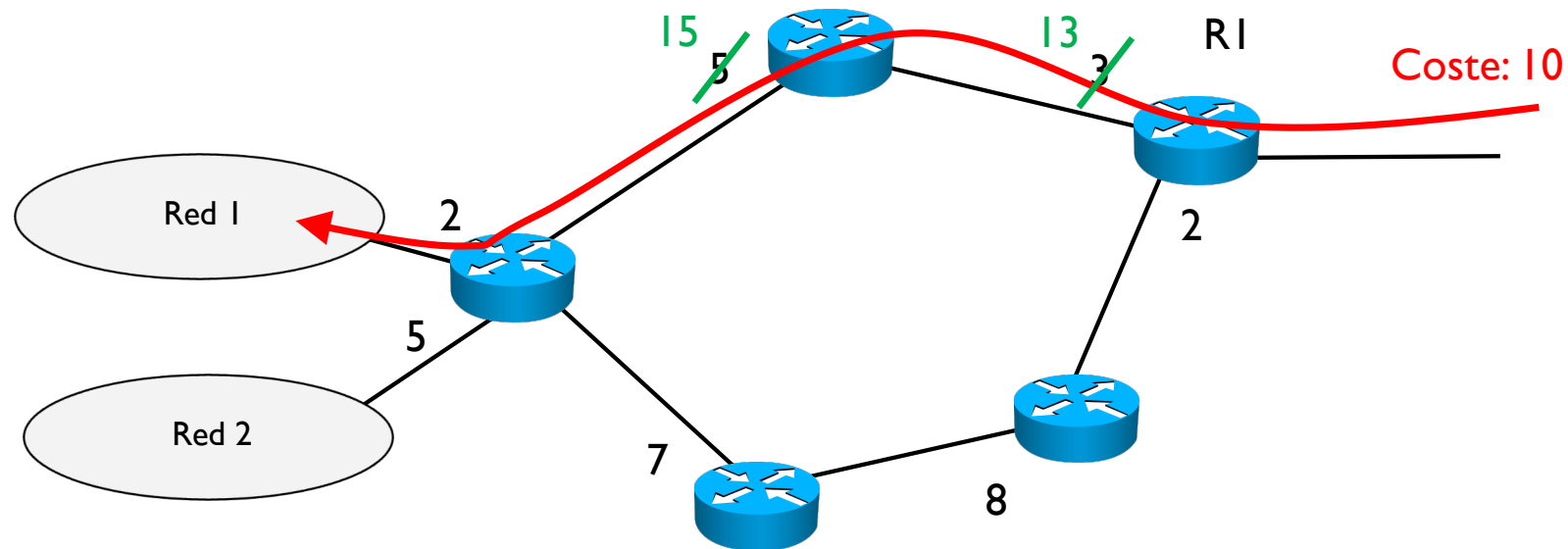
- ▶ **Infraestructura con costes dinámicos**
 - ▶ Mismo caso que antes
 - ▶ Cuando pero se establece el tráfico de RI a la red 1, cambian los costes



4.7 – MPLS con extensiones TE Traffic Engineering

► Infraestructura con costes dinámicos

- Mismo caso que antes
- Cuando pero se establece el tráfico de RI a la red I, cambian los costes

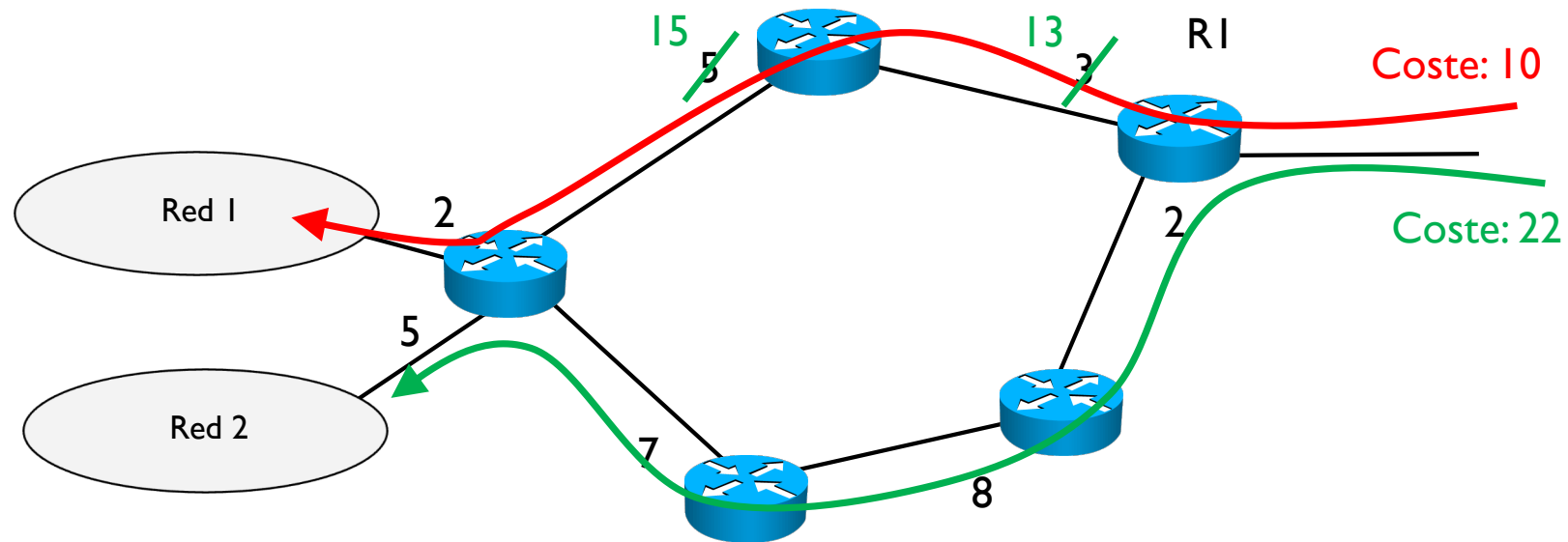


4.7 – MPLS con extensiones TE

Traffic Engineering

► Infraestructura con costes dinámicos

- Mismo caso que antes
- Cuando pero se establece el tráfico de RI a la red 1, cambian los costes
- Cuando llega la otra petición, ahora la mejor ruta es otra



4.7 – MPLS con extensiones TE

Traffic Engineering

- ▶ En el contexto que estamos tratando, estas extensiones afectan tres protocolos
 - ▶ OSPF-TE: protocolo de encaminamiento
 - ▶ MPLS-TE: protocolo de forwarding
 - ▶ RSVP-TE: protocolo de distribución de etiquetas
-
- ▶ Más una clase nueva de algoritmos de encaminamiento llamados Constraint-Based Routing (CBR)

4.7 – MPLS con extensiones TE

OSPF-TE

- ▶ Protocolo de encaminamiento igual que antes
- ▶ Se añade pero
 - ▶ Una métrica puede cambiar en el tiempo (en principio no decide OSPF como)
 - ▶ Cuando una métrica cambia, hay que notificar con un LSA todos los demás routers del nuevo valor de esta métrica
 - ▶ De esta forma, los routers siempre tienen una visión única y global del sistema
- ▶ Notar que
 - ▶ El cambio de valor de una métrica no afecta las entradas en las tablas ya creadas
 - ▶ Solo afectará las nuevas entradas

4.7 – MPLS con extensiones TE

MPLS-TE

- ▶ Se ocupa de crear las entradas en la tabla de forwarding de etiquetas como antes
- ▶ Además pero, permite que los clientes pidan unos determinados requerimientos a la hora de crear los LSP
- ▶ Se usa entonces un CBR para determinar estos caminos
 - ▶ CBR es una clase de algoritmos de encaminamiento basados en restricciones (constraints)
 - ▶ Datos los requerimientos de los clientes (las restricciones), un CBR es capaz de encontrar el mejor camino que proporciona, como mínimo, estos requerimientos

4.7 – MPLS con extensiones TE

RSVP-TE

- ▶ Conocido el estado global y actual de la infraestructura distribuyendo métricas dinámicas → OSPF-TE
- ▶ Conocido los requerimientos de los clientes y determinado el mejor camino → MPLS-TE
- ▶ Falta establecer este camino
 - ▶ Distribuir las etiquetas MPLS a lo largo de este camino → RSVP-TE
 - ▶ Verificar que este camino es realmente valido

4.7 – MPLS con extensiones TE

RSVP-TE

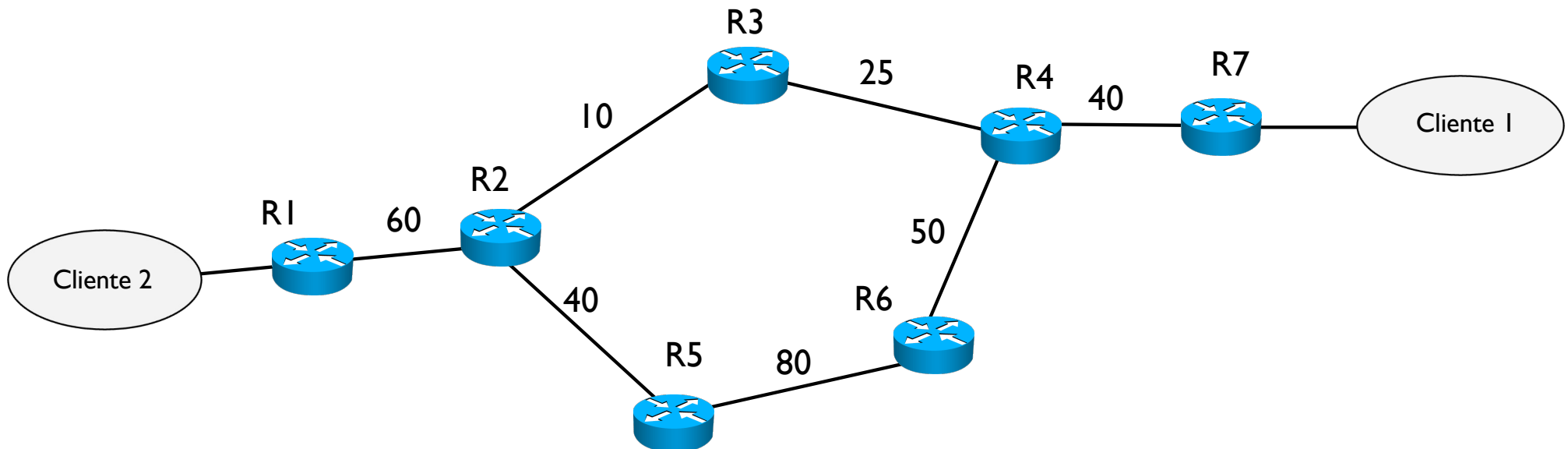
- ▶ Resource Reservation Protocol with Traffic Engineering
- ▶ Es un protocolo de distribución de etiquetas (como LDP)
- ▶ RSVP señala entonces el camino del destino al origen del LSP (sentido upstream)
 - ▶ Cada router genera una etiqueta local que pasa al siguiente router (como LDP)
 - ▶ Pero además lleva la información sobre los requerimientos específico del camino
 - ▶ Con estos requerimientos, cada router verifica que efectivamente puede proporcionarlos

4. MultiProtocol Label Switching (MPLS)

1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. Label Distribution Protocol (LDP)
6. Ejemplo de funcionamiento
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. MPLS con extensiones de TE
8. **Ejemplo de funcionamiento**
9. MPLS fast reroute

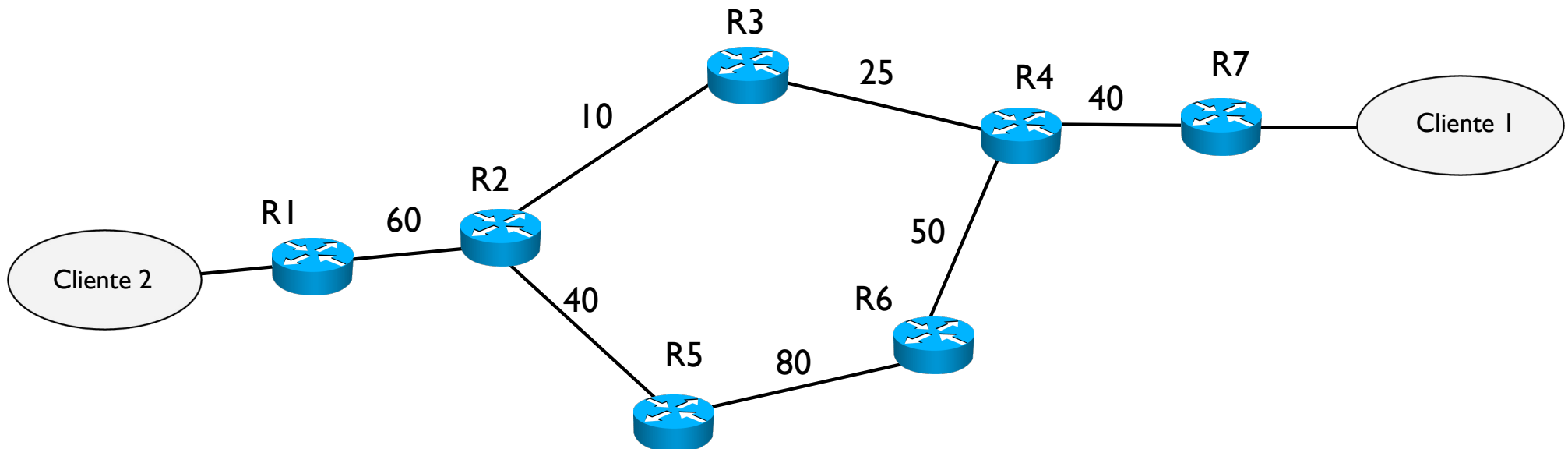
4.8 - Ejemplo de funcionamiento

- ▶ Se usa ancho de banda restante como única métrica
 - ▶ Suponiendo una métrica inicial de 100 Mbps, un valor X en un link-state significa que quedan disponible X Mbps (se están usando $100 - X$ Mbps en otros LSPs)
 - ▶ Cada petición de LSP tendrá como restricción el ancho de banda necesario
- ▶ OSPF-TE se ocupa de distribuir estas métricas a todos los routers
 - ▶ En la figura solo se muestran los valores que se usarán en el ejemplo



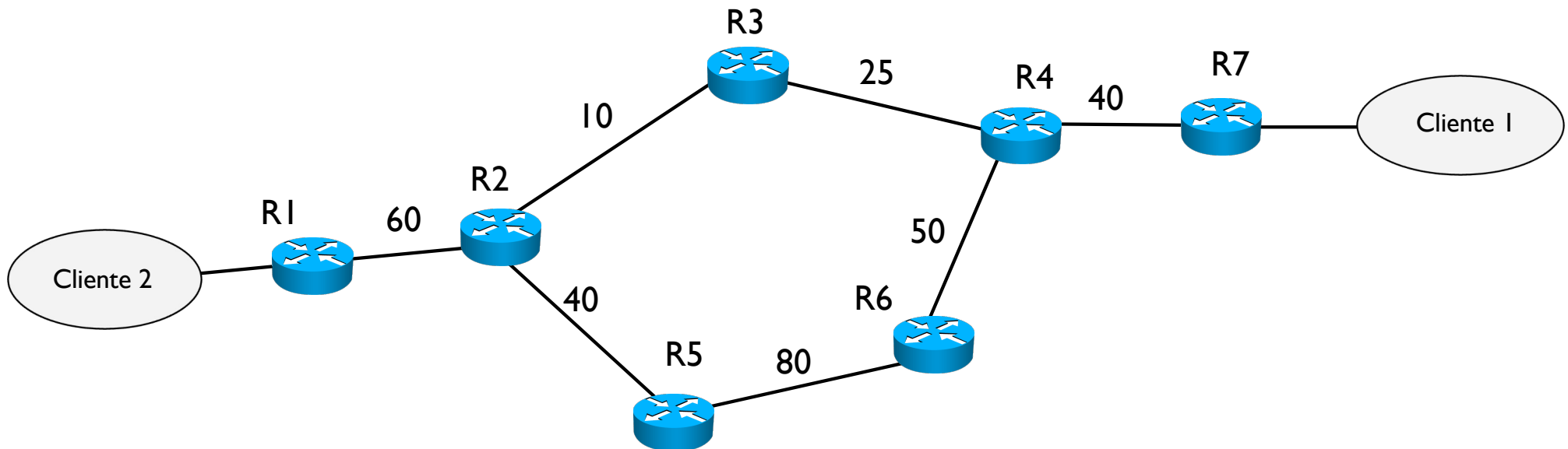
4.8 - Ejemplo de funcionamiento

- ▶ Suponemos que Cliente 1 quiere establecer un LSP con Cliente 2 para descargarse datos a 20 Mbps
 - ▶ La dirección es de Cliente 2 a Cliente 1
- ▶ Esta petición llega al sistema de gestión de la infraestructura
 - ▶ Este sistema (que no tratamos) comanda a R1 de empezar la gestión para computar y establecer este LSP de 20 Mbps hacia R7



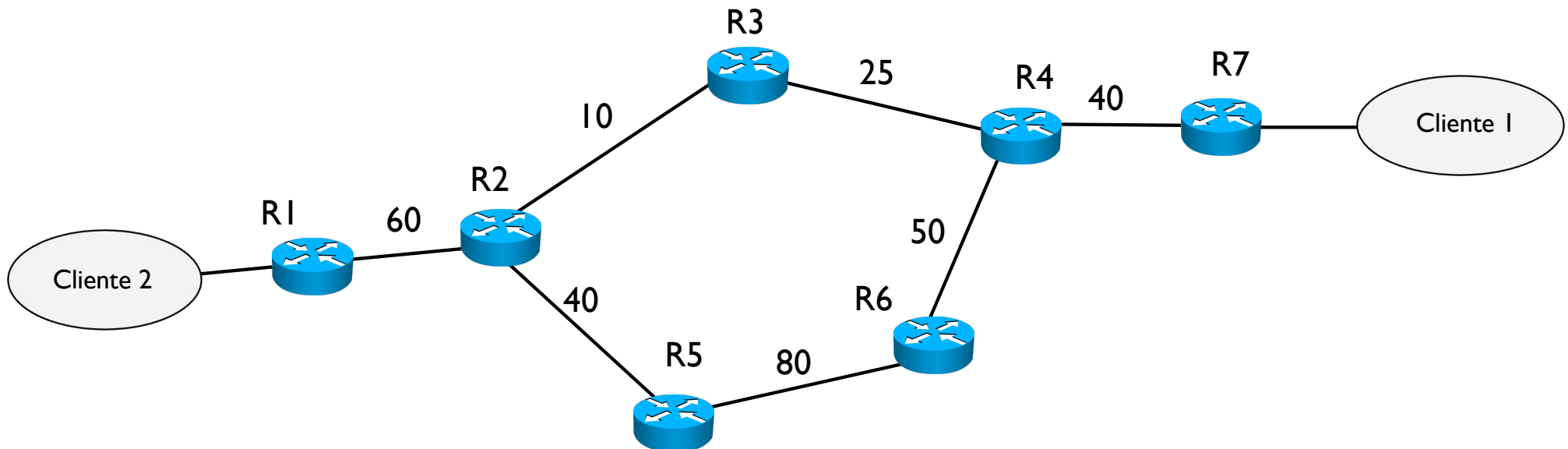
4.8 - Ejemplo de funcionamiento

- ▶ R1 tendrá la visión completa del sistema actual (gracias a OSPF-TE)
- ▶ R1 ejecuta el CBR y determina que solo hay disponible un camino R1-R5-R6-R4-R7
- ▶ R1 lanza RSVP-TE para crear el camino y definir las etiquetas en cada router



4.8 - Ejemplo de funcionamiento

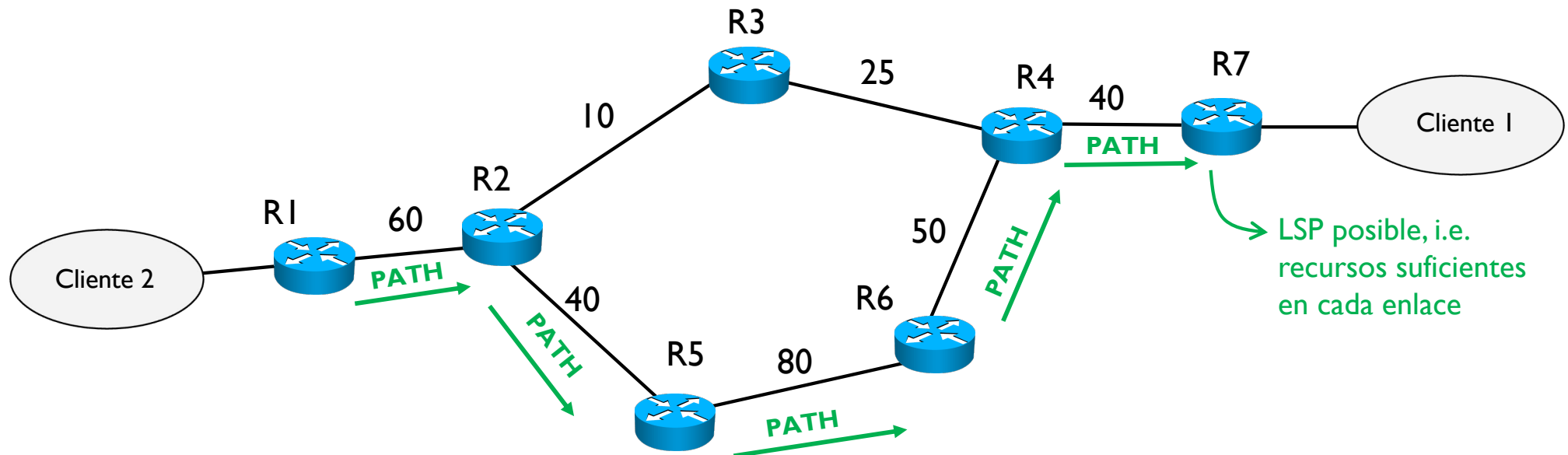
- ▶ RSVP-TE usa
 - ▶ el mensaje PATH de R1 a R7 para hacer pre-reserva de los recursos y verificar que el camino es posible
 - ▶ el mensaje RESV de vuelta de R7 a R1 para hacer efectiva las reservas, definir las etiquetas locales y las entradas en las tablas MPLS
 - ▶ el mensaje PATH_error si encuentra algún problema en el camino de ida (el camino no es posible)



4.8 - Ejemplo de funcionamiento

► RSVP-TE usa

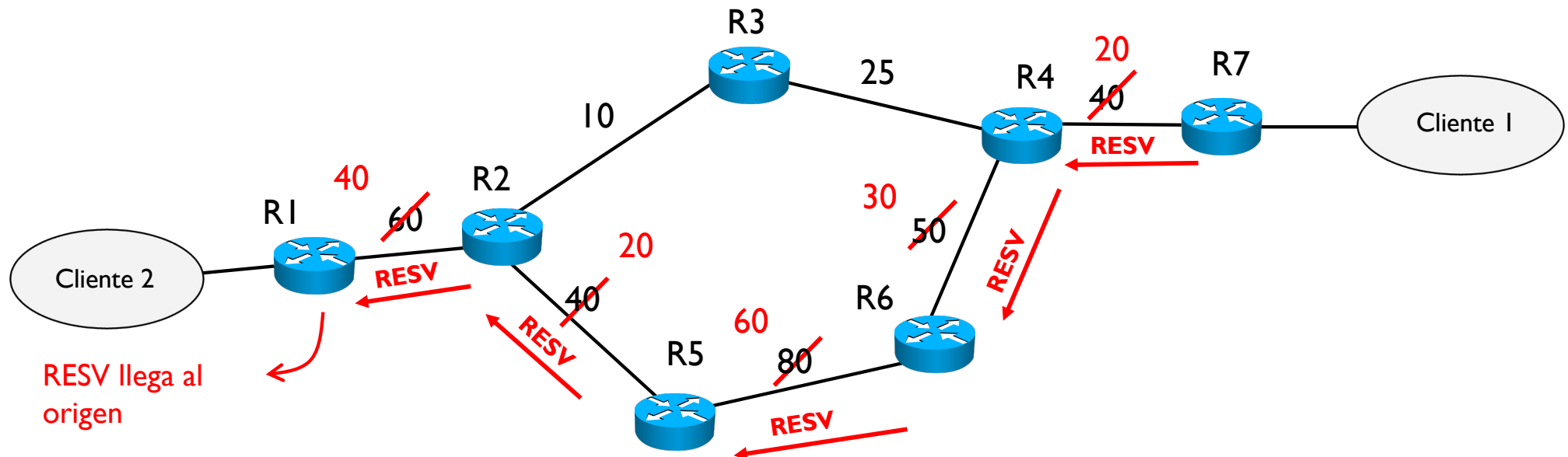
- el mensaje PATH de R1 a R7 para hacer pre-reserva de los recursos y verificar que el camino es posible
- el mensaje RESV de vuelta de R7 a R1 para hacer efectiva las reservas, definir las etiquetas locales y las entradas en las tablas MPLS
- el mensaje PATH_error si encuentra algún problema en el camino de ida (el camino no es posible)



4.8 - Ejemplo de funcionamiento

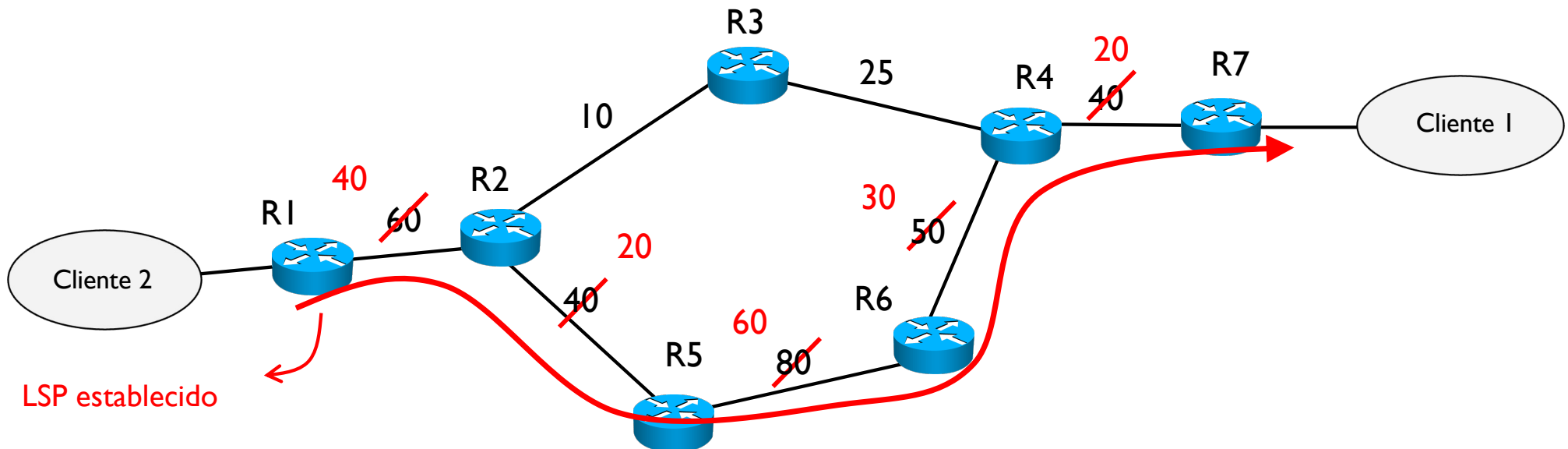
► RSVP-TE usa

- el mensaje PATH de R1 a R7 para hacer pre-reserva de los recursos y verificar que el camino es posible
- el mensaje RESV de vuelta de R7 a R1 para hacer efectiva las reservas, definir las etiquetas locales y las entradas en las tablas MPLS
- el mensaje PATH_error si encuentra algún problema en el camino de ida (el camino no es posible)



4.8 - Ejemplo de funcionamiento

- ▶ RSVP-TE usa
 - ▶ el mensaje PATH de R1 a R7 para hacer pre-reserva de los recursos y verificar que el camino es posible
 - ▶ el mensaje RESV de vuelta de R7 a R1 para hacer efectiva las reservas, definir las etiquetas locales y las entradas en las tablas MPLS
 - ▶ el mensaje PATH_error si encuentra algún problema en el camino de ida (el camino no es posible)

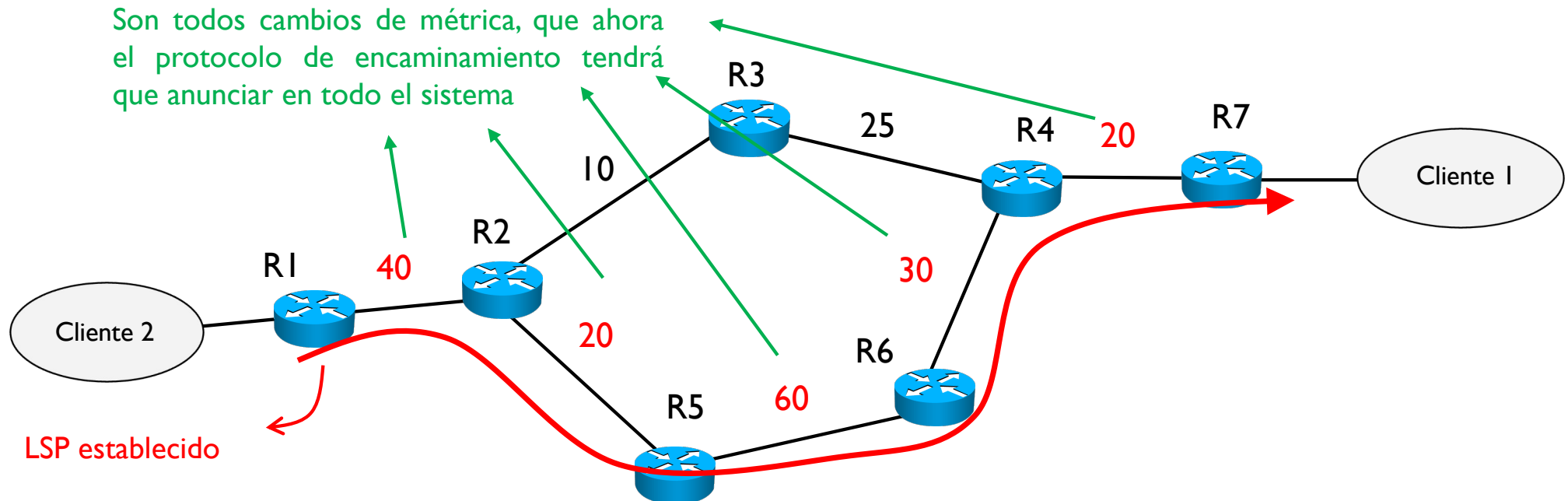


4.8 - Ejemplo de funcionamiento

► RSVP-TE usa

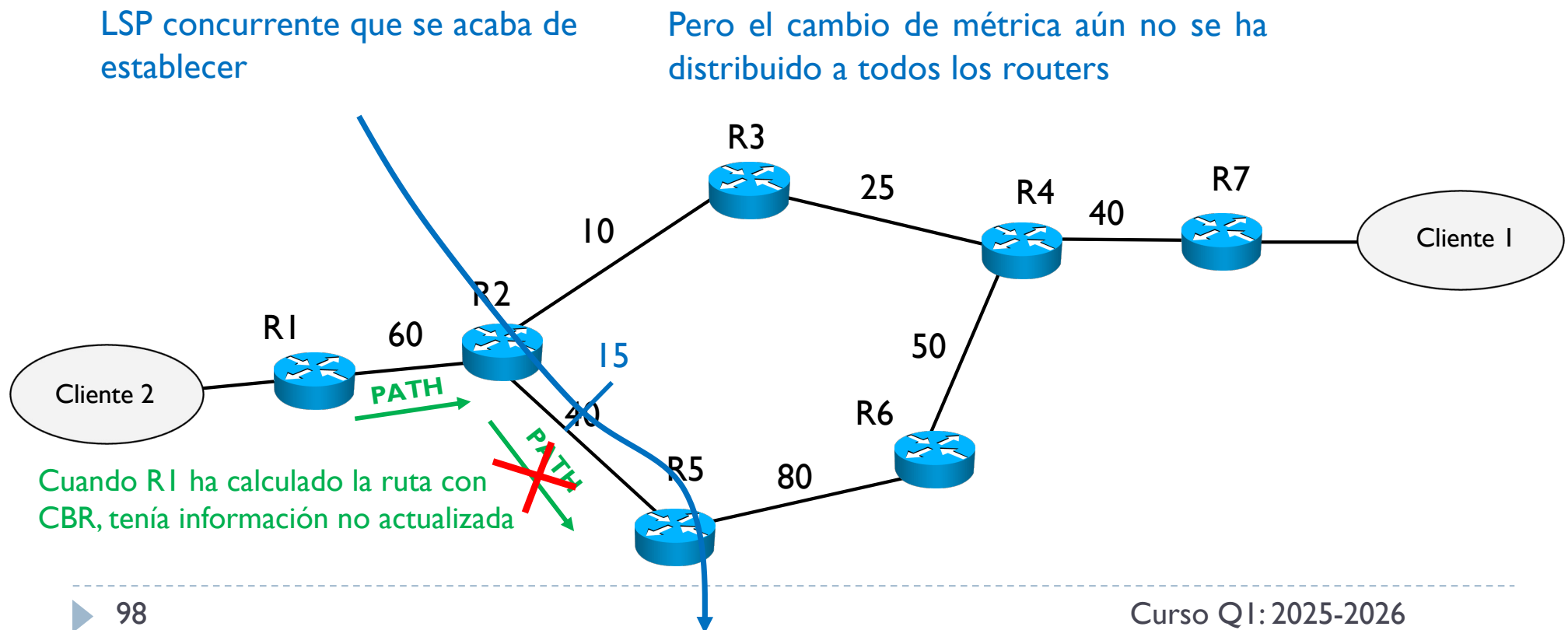
- el mensaje PATH de R1 a R7 para hacer pre-reserva de los recursos y verificar que el camino es posible
- el mensaje RESV de vuelta de R7 a R1 para hacer efectiva las reservas, definir las etiquetas locales y las entradas en las tablas MPLS
- el mensaje PATH_error si encuentra algún problema en el camino de ida (el camino no es posible)

Son todos cambios de métrica, que ahora el protocolo de encaminamiento tendrá que anunciar en todo el sistema



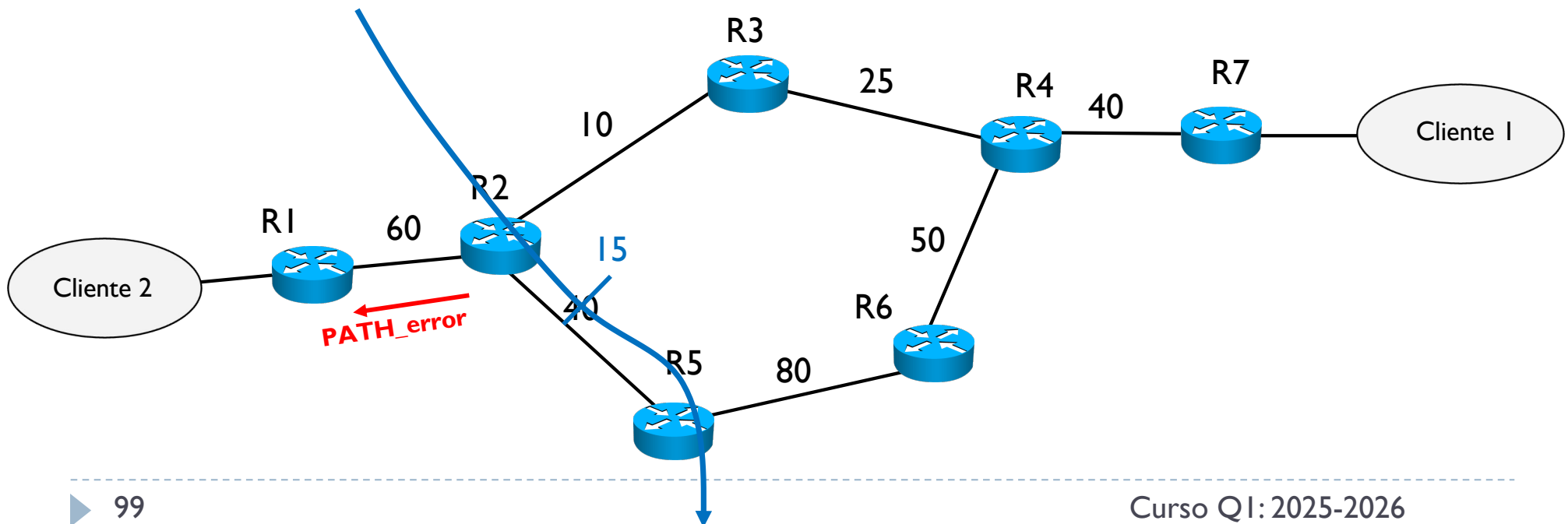
4.8 - Ejemplo de funcionamiento

- ▶ Es posible que la información que tenga cada router no sea completamente actualizada
 - ▶ Puede que haya dos o más peticiones concurrentes y que el nuevo estado de cada link no haya aún llegado a R1



4.8 - Ejemplo de funcionamiento

- ▶ Para eso el PATH verifica que cada link cumpla con la restricción
 - ▶ Si se verifica que no hay suficiente, se genera un PATH_error de vuelta al origen
 - ▶ Según el tipo de servicio contratado, se puede o bien bloquear el LSP, o bien volver a intentarlo por otro camino o bien establecer un LSP con menos ancho de banda



4. MultiProtocol Label Switching (MPLS)

1. Introducción a MPLS
2. Terminología
3. Formato de una etiqueta
4. Estructura de las tablas MPLS
5. Label Distribution Protocol (LDP)
6. Ejemplo de funcionamiento
 - ▶ Con Penultimate Hop Popping (PHP)
 - ▶ Label Stack
7. MPLS con extensiones de TE
8. Ejemplo de funcionamiento
9. **MPLS fast reroute**

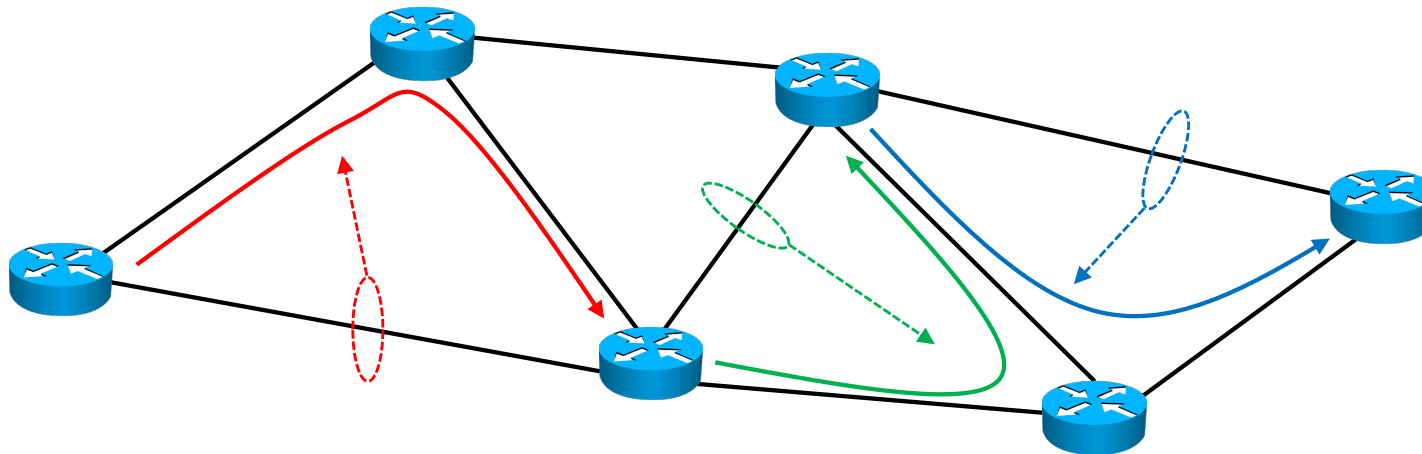
4.9 - MPLS fast reroute

- ▶ En caso de algún fallo en el sistema
- ▶ Redes IP
 - ▶ Cuando hay un fallo, se distribuye el nuevo estado, se calculan nuevos caminos y se crean nuevas entradas en las tablas
 - ▶ Método de tipo reactivo
 - ▶ La convergencia puede ser lenta
- ▶ Redes MPLS
 - ▶ Se calculan caminos alternativos desde el principio que se activan solo cuando se detecta un fallo
 - ▶ Método de tipo proactivo
 - ▶ La convergencia es más rápida

4.9 - MPLS fast reroute

► Protección del enlace

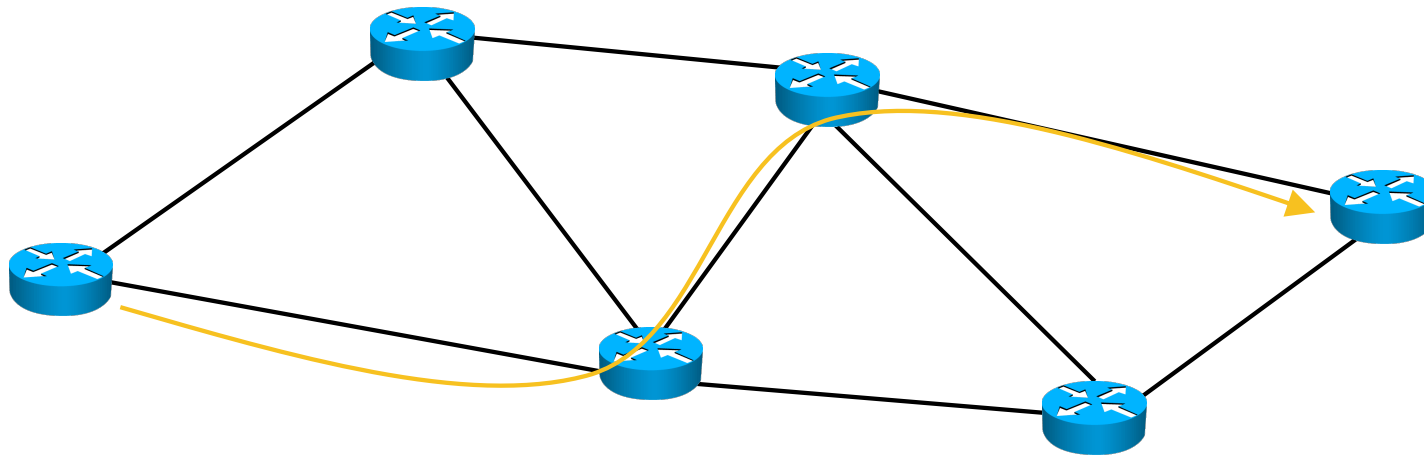
- Cada enlace se protege con un LSP de backup
- Este LSP de backup tiene la misma pareja de routers que conecta este enlace, pero usa un camino alternativo a la conexión directa
- Todos los LSP que pasan por un enlace que cae, se re-direccionan en el LSP de backup, para luego volver al LSP original



4.9 - MPLS fast reroute

► Protección del enlace

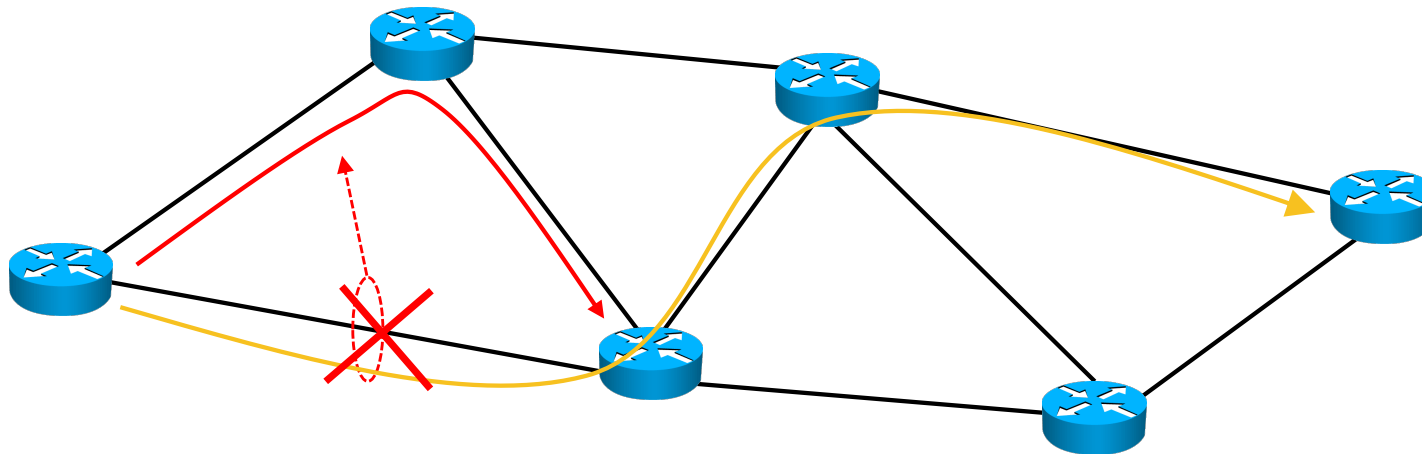
- Cada enlace se protege con un LSP de backup
- Este LSP de backup tiene la misma pareja de routers que conecta este enlace, pero usa un camino alternativo a la conexión directa
- Todos los LSP que pasan por un enlace que cae, se re-direccionan en el LSP de backup, para luego volver al LSP original



4.9 - MPLS fast reroute

► Protección del enlace

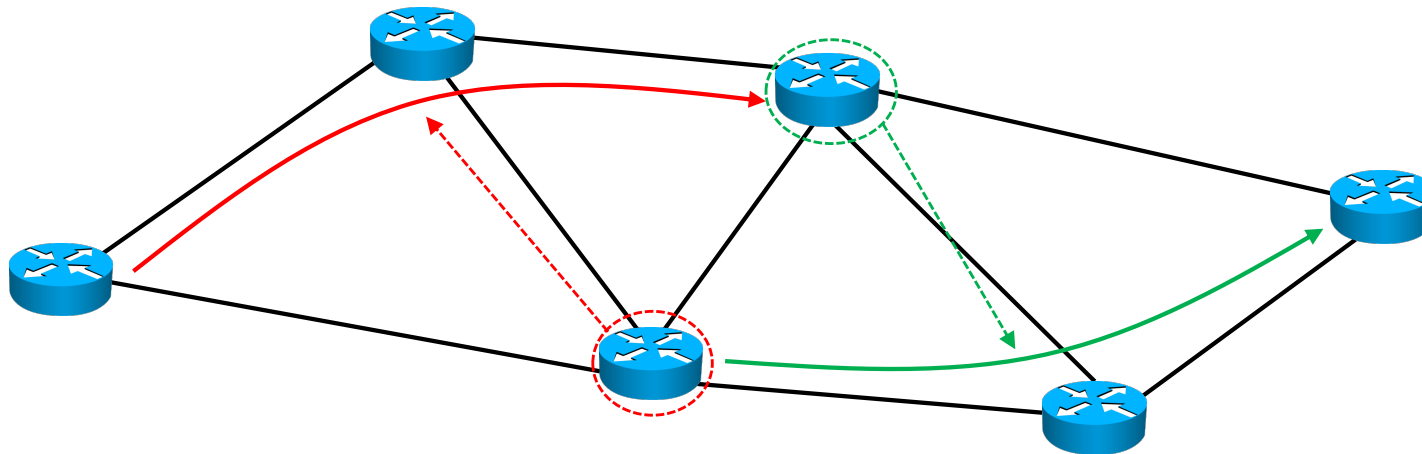
- Cada enlace se protege con un LSP de backup
- Este LSP de backup tiene la misma pareja de routers que conecta este enlace, pero usa un camino alternativo a la conexión directa
- Todos los LSP que pasan por un enlace que cae, se re-direccionan en el LSP de backup, para luego volver al LSP original



4.9 - MPLS fast reroute

► Protección del nodo

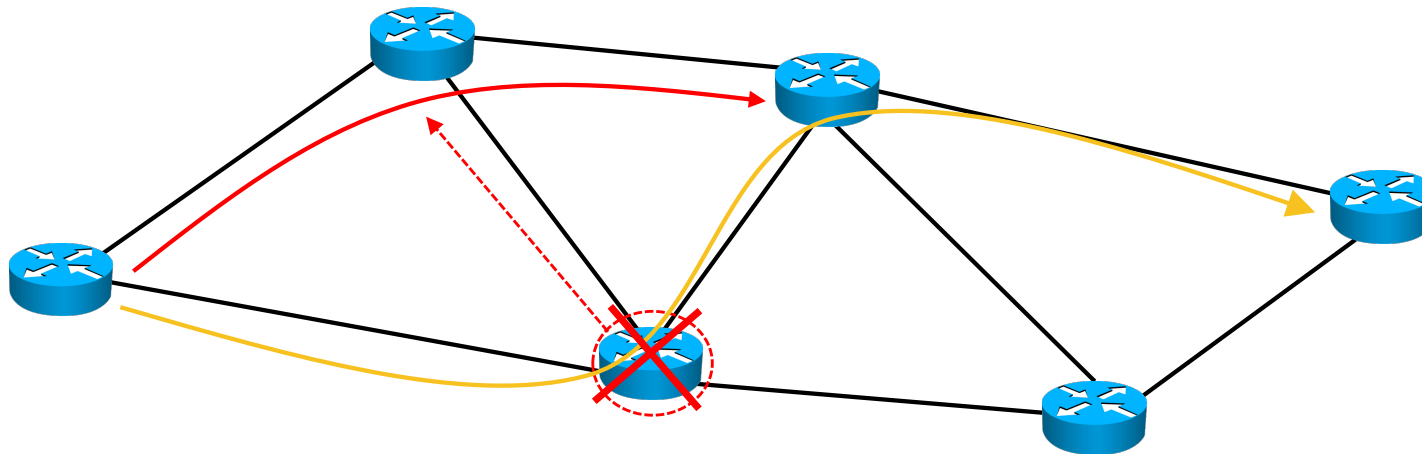
- Cada nodo (router) se protege con un LSP de backup
- Este LSP de backup tiene que proteger todos los LSP que pasan por el nodo
- Todos los LSP que pasan por un nodo que cae, se re-direccionan en el LSP de backup, para luego volver al LSP original



4.9 - MPLS fast reroute

► Protección del nodo

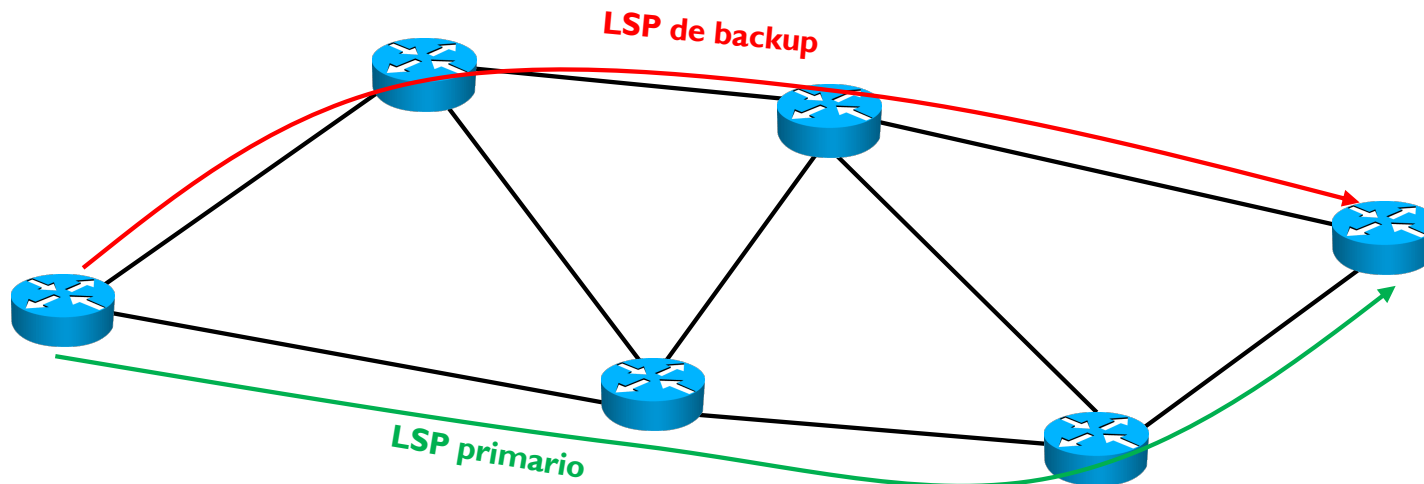
- Cada nodo (router) se protege con un LSP de backup
- Este LSP de backup tiene que proteger todos los LSP que pasan por el nodo
- Todos los LSP que pasan por un nodo que cae, se re-direccionan en el LSP de backup, para luego volver al LSP original



4.9 - MPLS fast reroute

► Protección del LSP

- En este caso, a la hora de establecer un LSP (llamado primario), se crea otro LSP de backup que sea disjunto del primario
 - Disjunto: no debe compartir ningún enlace o nodo con el primario (excepto origen y destino)
- Si cae un nodo o un enlace, se pasa a usar enteramente el LSP de backup
- Un LSP de backup puede ser único de un primario o puede estar compartido entre varios



Xarxes de Computadors II

Tema 4. Multiprotocol Label Switching (MPLS)

Davide Careglio