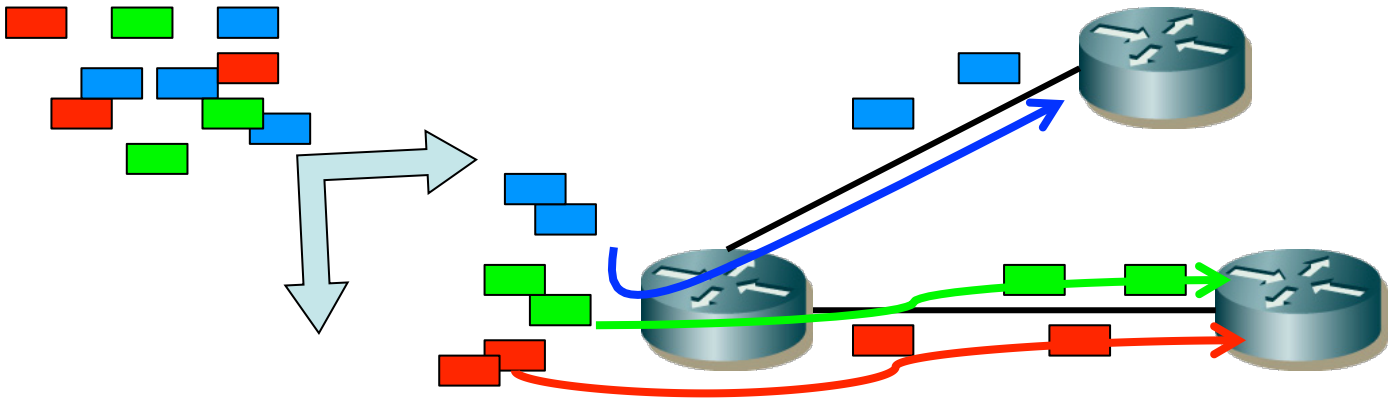
MPLS

Area de Ingeniería Telemática
http://www.tlm.unavarra.es

Redes
4º Ingeniería Informática

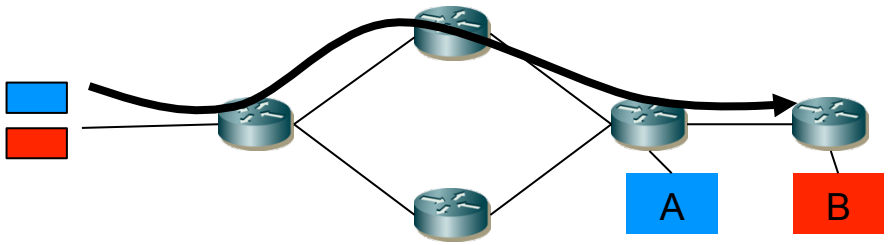
Forwarding IP

- Selección del siguiente salto está compuesto de:
  - Particionar el espacio de paquetes en "Forwarding Equivalence Classes (FECs)"
  - Hacer corresponder cada FEC con un siguiente salto
- Paquetes diferentes que pertenezcan al mismo FEC son indistinguibles respecto al proceso de reenvío
- Paquetes del mismo FEC en el mismo nodo seguirán el mismo camino



FEC

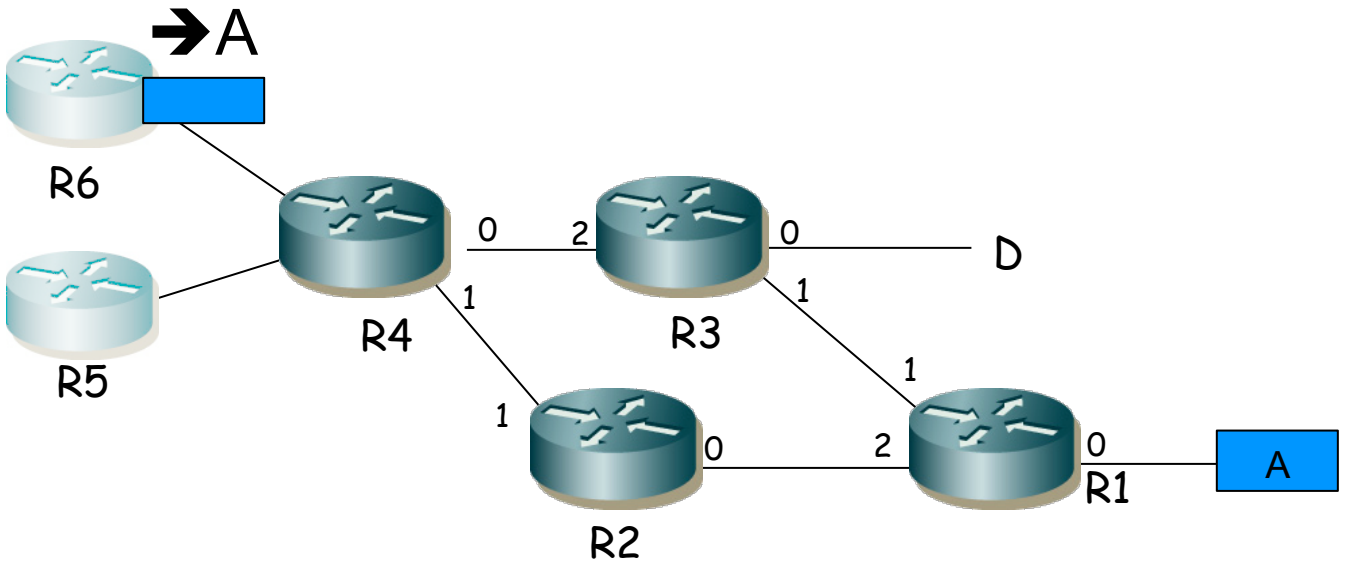
- *Forwarding Equivalence Class*
- Trafico clasificado en el mismo FEC en un nodo sigue el mismo camino
- En forwarding IP convencional
  - El FEC viene determinado por el longest prefix match
  - Cada salto reexamina y asigna el paquete a un FEC
- Problemas:
  - Longest prefix match era costoso (ahora no se hace en CPU)
  - Esas decisiones costosas se debían tomar en cada salto
  - Poco flexible pues se encaminaba solo en función del destino
  - Imposibilidad de elegir rutas alternativas se se deciden en base al menor coste de camino (SPF)
- (…)



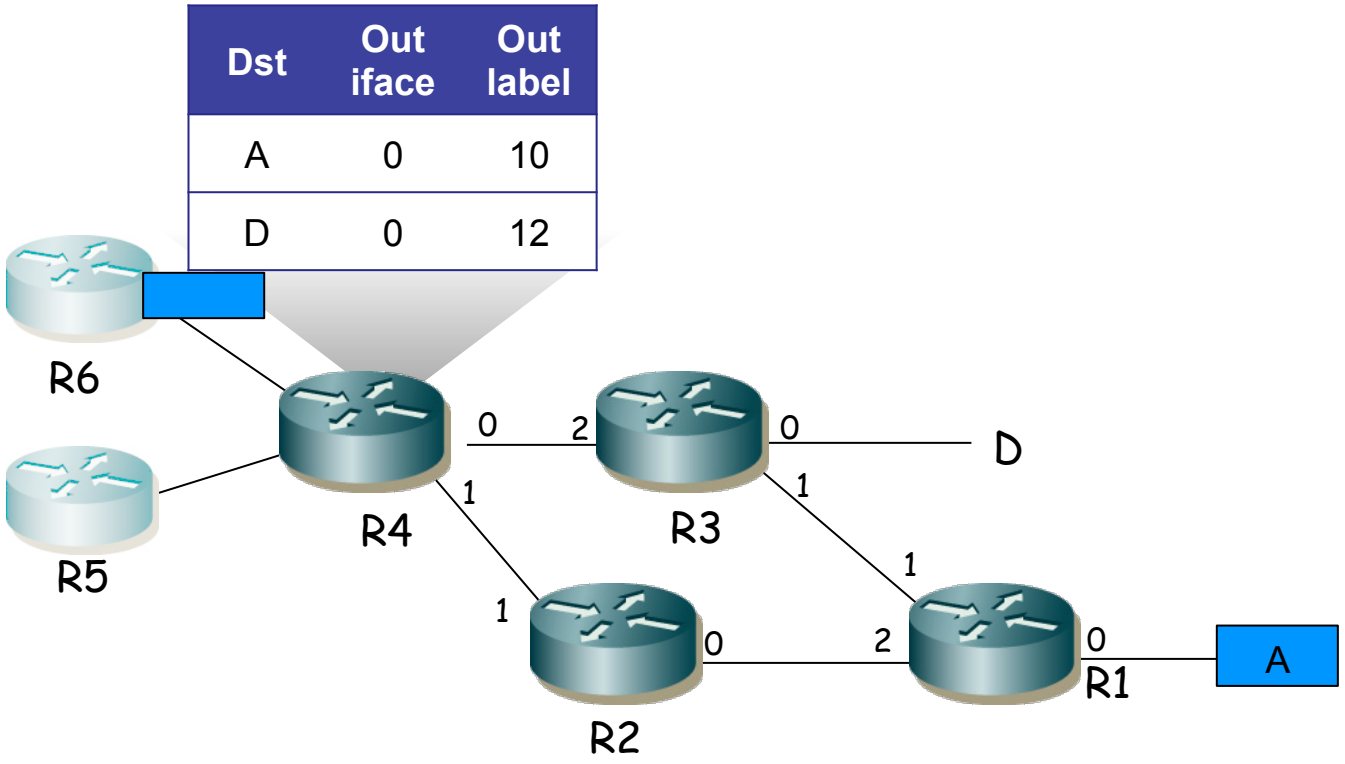
FEC

- *Forwarding Equivalence Class*
- Trafico clasificado en el mismo FEC en un nodo sigue el mismo camino
- En forwarding IP convencional
  - El FEC viene determinado por el longest prefix match
  - Cada salto reexamina y asigna el paquete a un FEC
- MultiProtocol Label Switching (RFC 3031 "***MPLS** Architecture"*)
  - El nodo de entrada a la red (ingress router) hace la asignación de cada paquete a un FEC
  - El FEC se indica mediante una etiqueta que viaja con el paquete
  - En saltos siguientes no hay necesidad de identificar el FEC pues se tiene la etiqueta
  - La etiqueta se emplea como índice en una tabla que especifica un siguiente salto y una nueva etiqueta
  - La etiqueta que traía el paquete se sustituye por la nueva
  - Reenvío MPLS no requiere que los nodos sepan procesar la cabecera del nivel de red (u otro protocolo encapsulado)

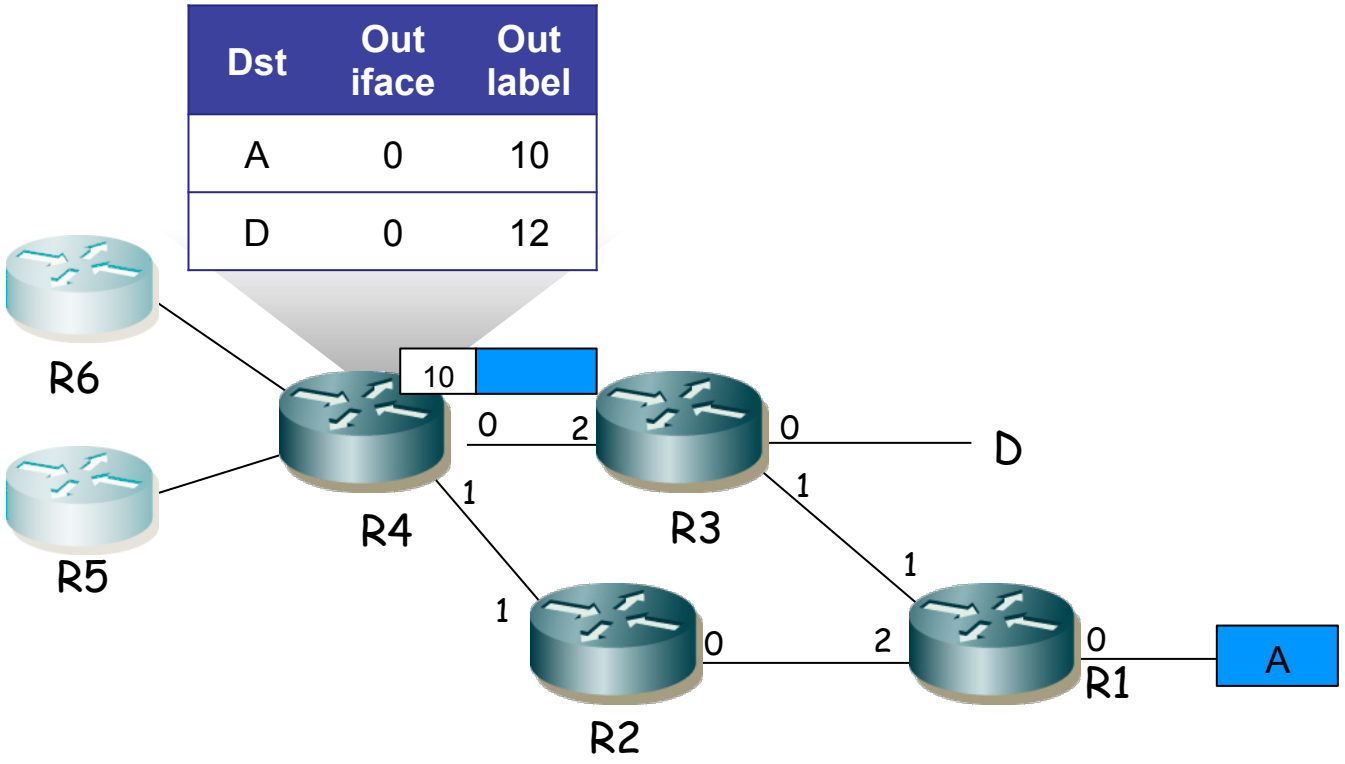
MPLS “forwarding”



MPLS “forwarding”



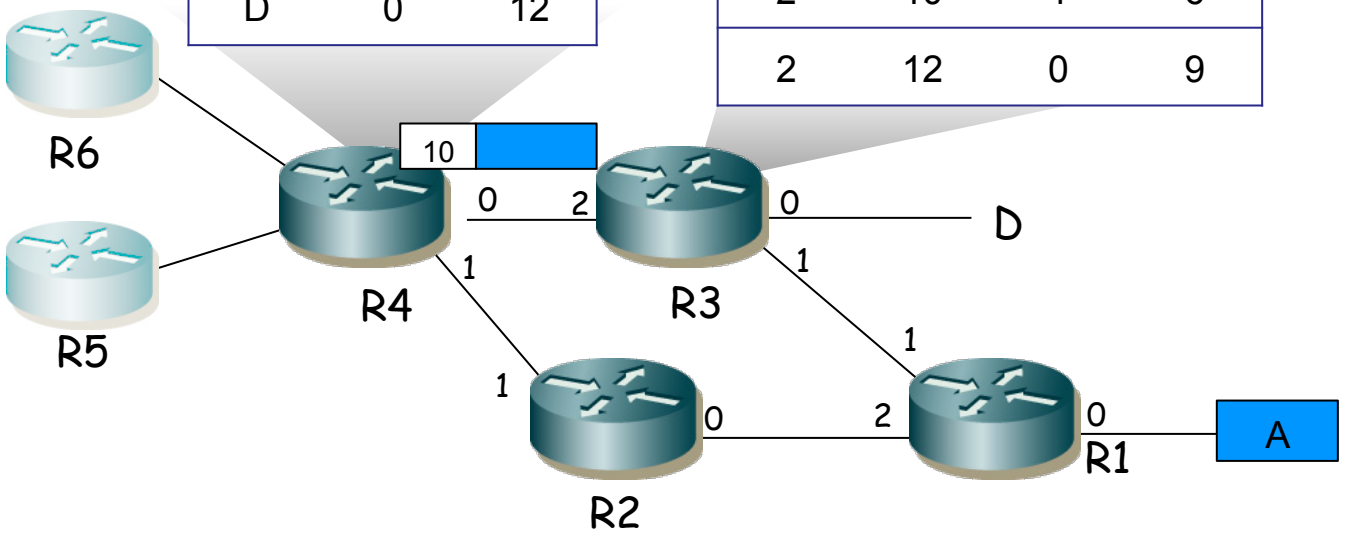
MPLS "forwarding"



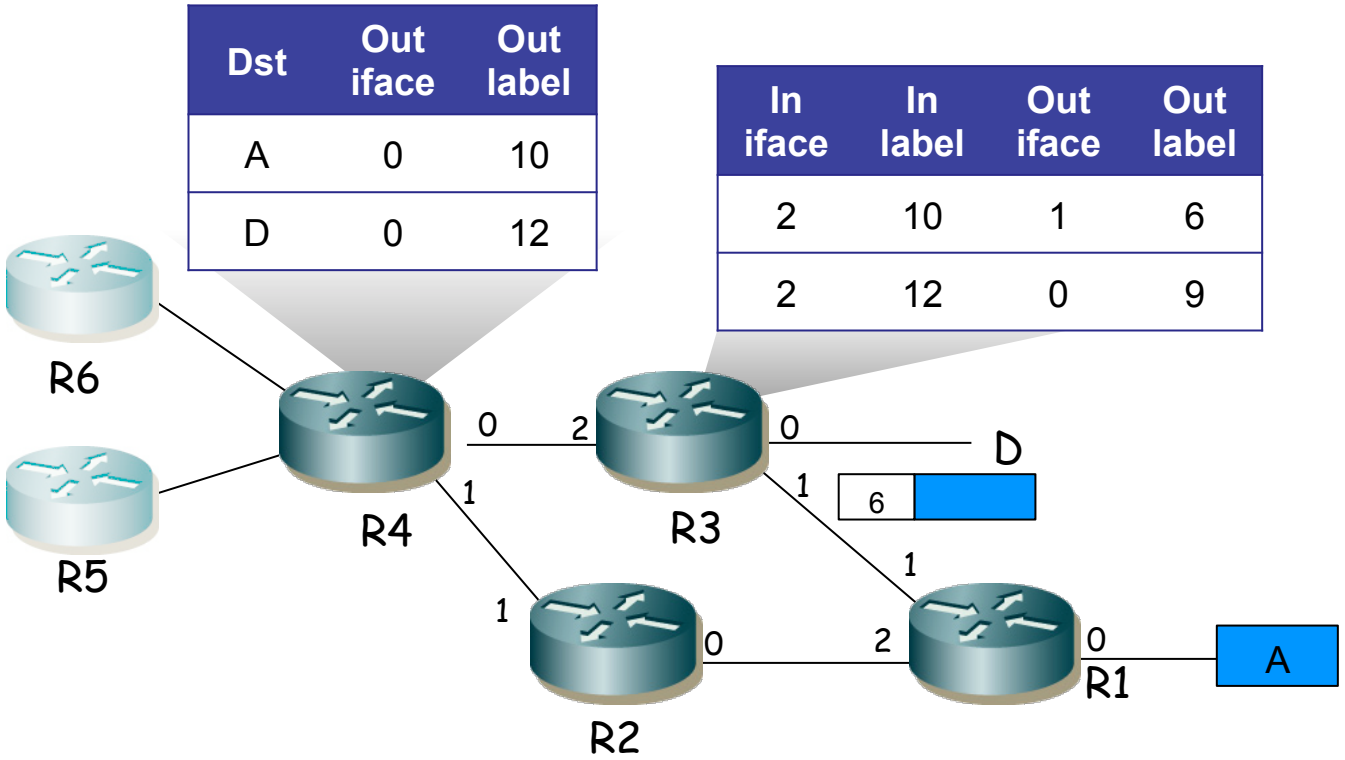
MPLS "forwarding"

Dst	Out iface	Out label
A	0	10
D	0	12

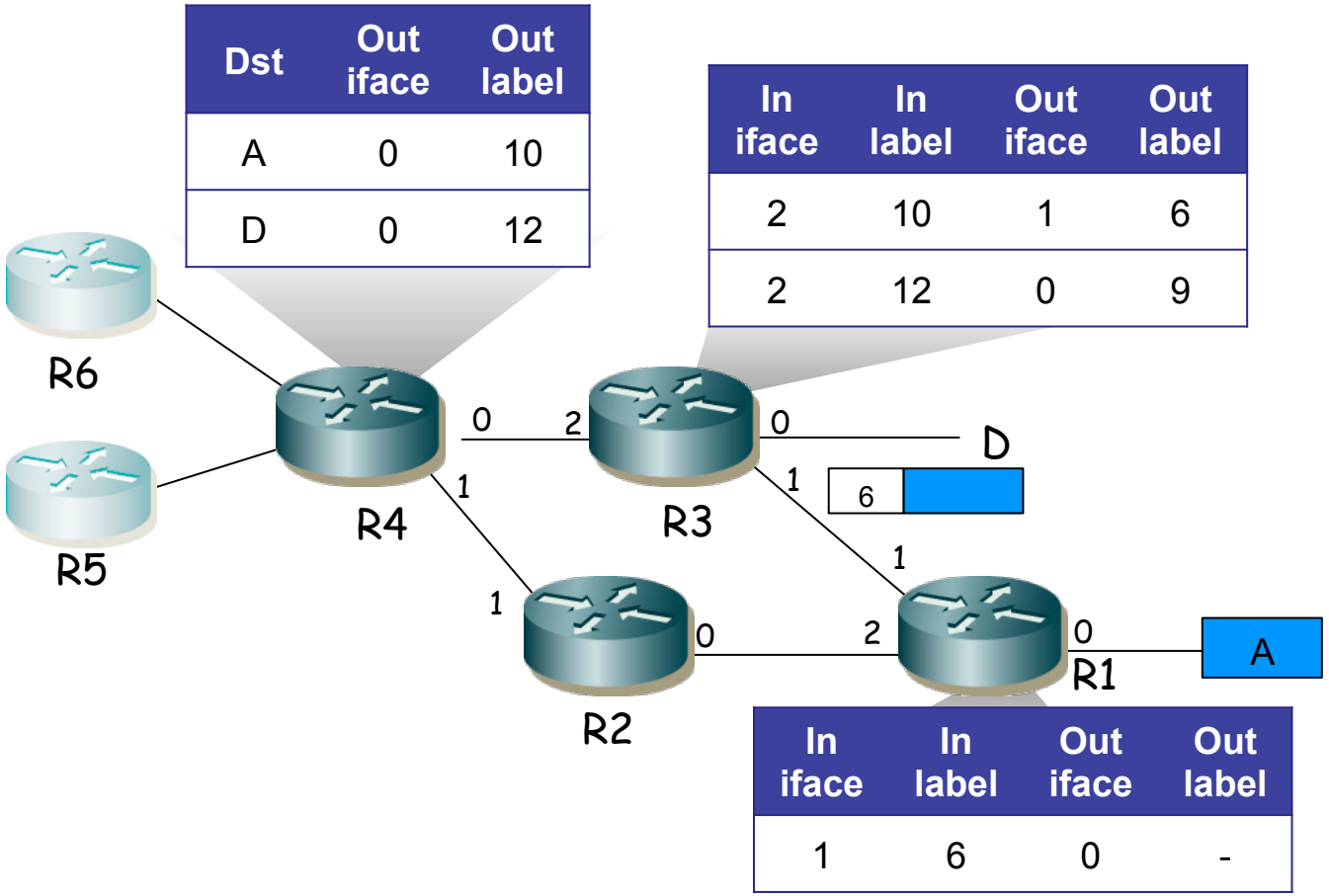
In iface	In label	Out iface	Out label
2	10	1	6
2	12	0	9



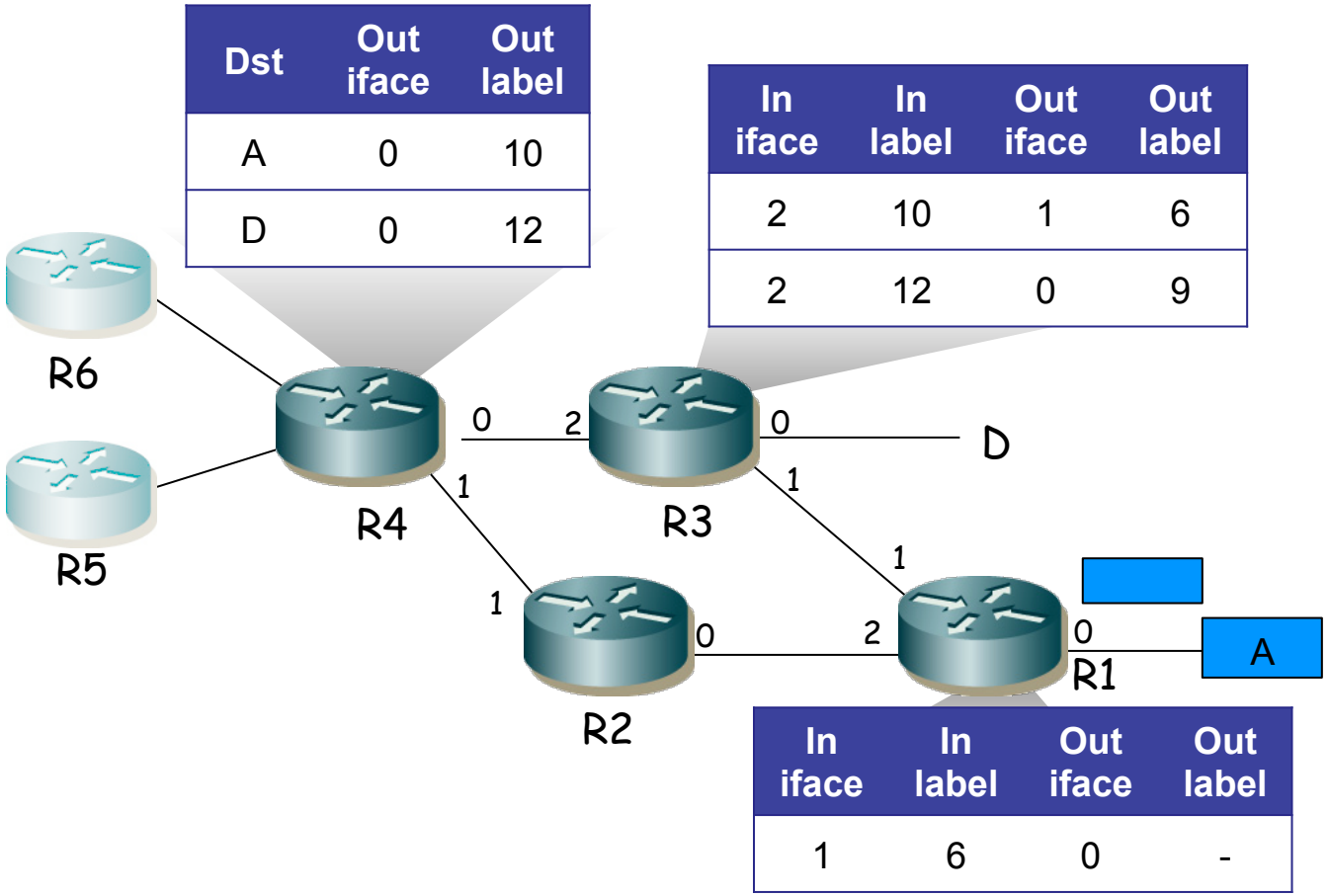
MPLS "forwarding"



MPLS "forwarding"

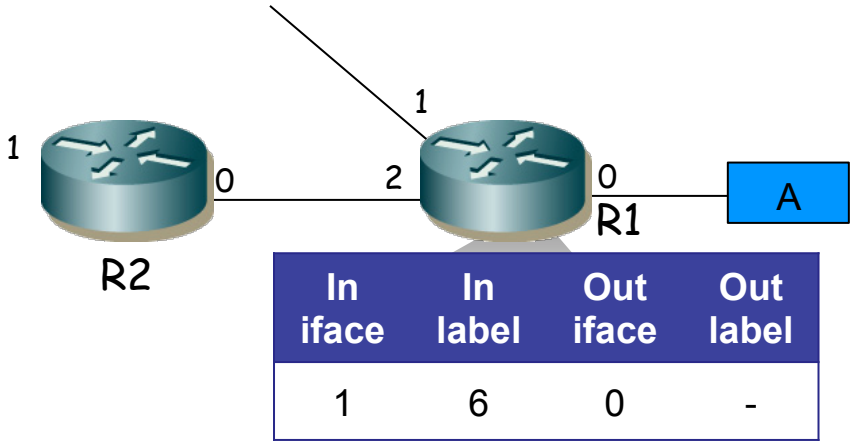


MPLS "forwarding"



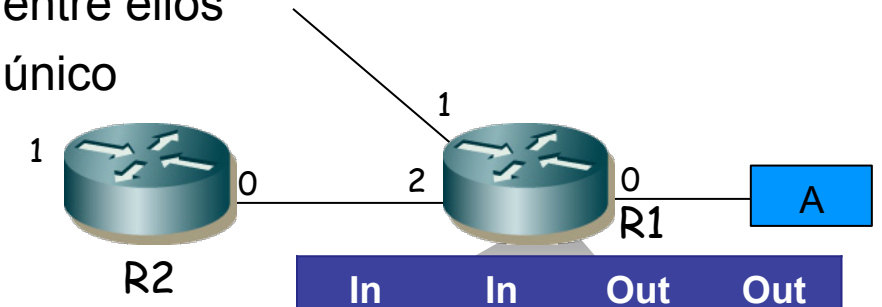
Conceptos

- "MPLS domain": conjunto contiguo de nodos MPLS bajo una misma administración
- "MPLS ingress node": nodo frontera de un dominio en su tarea como entrada de tráfico al mismo
- "MPLS egress node": nodo frontera de un dominio en su tarea como salida de tráfico del mismo
- "Label": etiqueta numérica, corta, longitud fija, identifica a un FEC localmente a un enlace
- "Label Switching Router (LSR)": nodo MPLS capaz de reenviar en base a etiquetas
- "Label Switched Path (LSP)": camino a través de LSRs



Conceptos

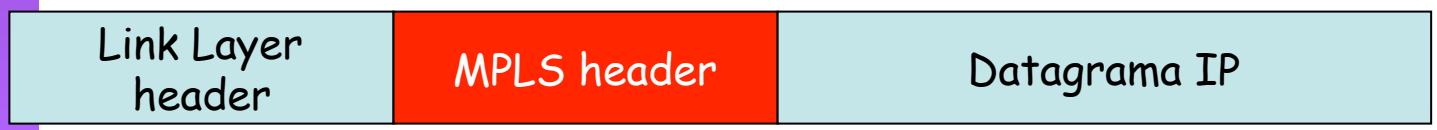
- La etiqueta representa un FEC en paquetes de nodo “upstream” a nodo “downstream”:
- Ejemplo: R2 es el “upstream LSR”, R1 es el “downtream LSR”
- El nodo downstream es quien toma la decisión de asociar una etiqueta a un FEC
- Nodo downstream informa al upstream de la asociación
- Soporta que nodo upstream solicite asociación (label,FEC)
- Un LSR informa a otro mediante un “label distribution protocol”
- Dos LSRs que usan un protocolo de distribución de etiquetas entre ellos son “label distribution peers”
- Si dos LSRs son “label distribution peers” se dice que existe una “label distribution adjacency” entre ellos
- No existe un protocolo único

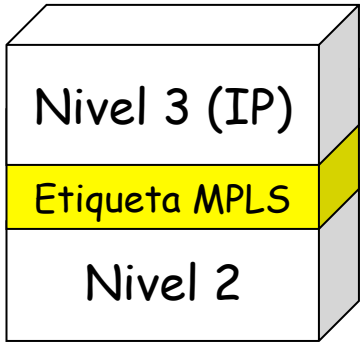


In iface	In label	Out iface	Out label
1	6	0	-

Label Stack

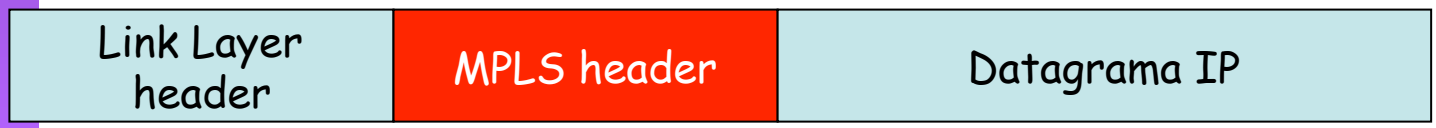
- La localización de la etiqueta depende de la tecnología que transporte los paquetes
- Una posibilidad es emplear un "*shim header*" entre cabecera del nivel de enlace y del protocolo transportado
- Hay otras opciones, por ejemplo si el transporte es sobre ATM se emplea el VPI/VCI como etiqueta
- A veces se dice que es una tecnología de nivel 2.5
- En realidad la etiqueta puede no ser única sino una "pila" de etiquetas (*label stack*) (…)

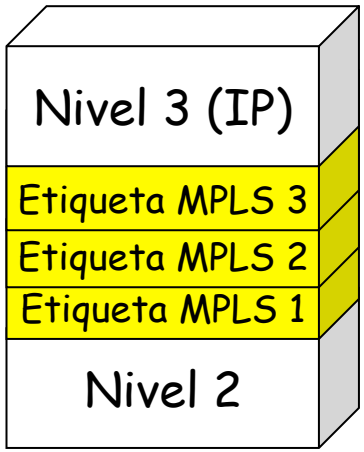




Label Stack

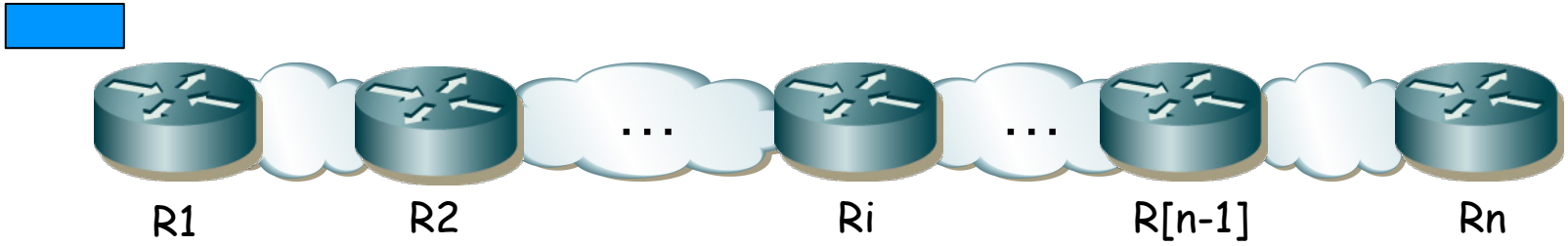
- La localización de la etiqueta depende de la tecnología que transporte los paquetes
- Una posibilidad es emplear un “shim header” entre cabecera del nivel de enlace y del protocolo transportado
- Hay otras opciones, por ejemplo si el transporte es sobre ATM se emplea el VPI/VCI como etiqueta
- A veces se dice que es una tecnología de nivel 2.5
- En realidad la etiqueta puede no ser única sino una “pila” de etiquetas (*label stack*) (…)
- El procesado se basa siempre en la etiqueta superior
- Un paquete sin etiquetar tiene profundidad 0 de pila
- En un LSR se puede emplear espacio de etiquetas:
  - Por interfaz
  - Por plataforma





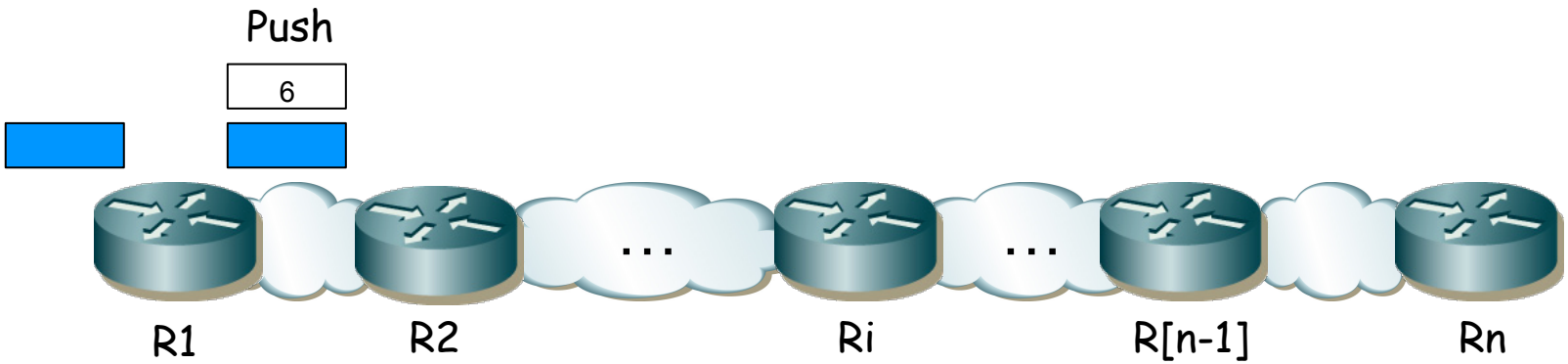
LSP de nivel m

- Secuencia de routers, paquete P con pila de profundidad m-1
- (…)



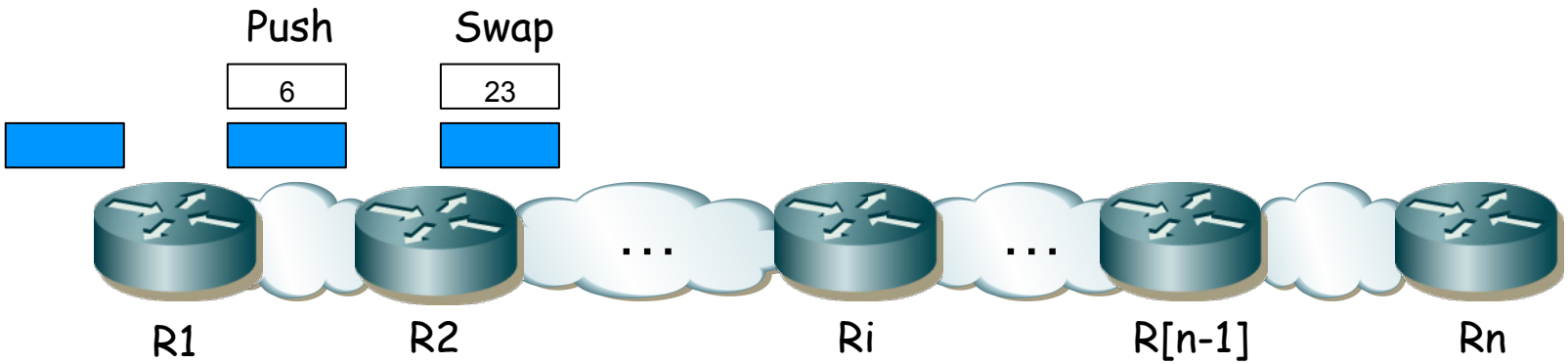
LSP de nivel m

- Secuencia de routers, paquete P con pila de profundidad m-1
- R1: LSP ingress, añade (*push*) una etiqueta a la pila del paquete



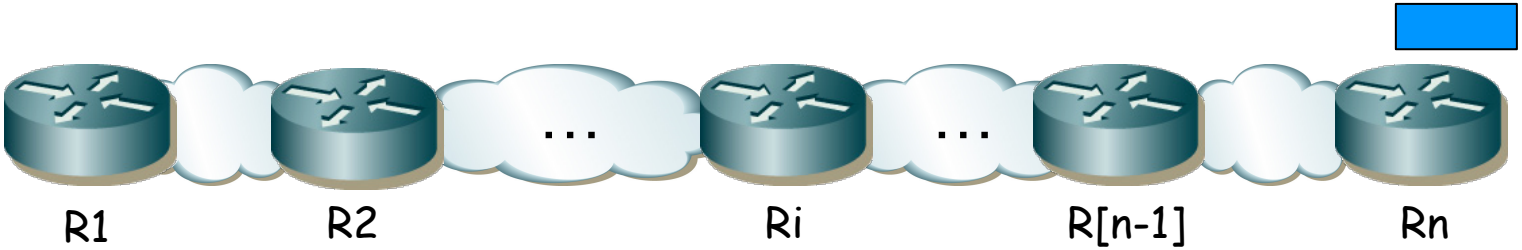
LSP de nivel m

- Secuencia de routers, paquete P con pila de profundidad m-1
- R1: LSP ingress, añade (*push*) una etiqueta a la pila del paquete
- 1<i<n Ri recibe paquete P con una pila de etiquetas de profundidad m
- En el tránsito entre R1 y R[n-1] el paquete P nunca tiene una pila de profundidad menor que m



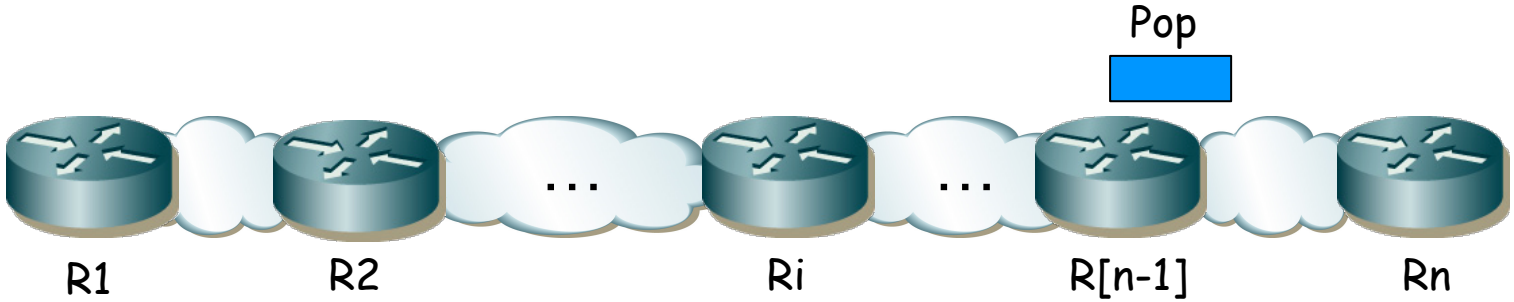
LSP de nivel m

- Secuencia de routers, paquete P con pila de profundidad m-1
- R1: LSP ingress, añade (*push*) una etiqueta a la pila del paquete
- 1<i<n Ri recibe paquete P con una pila de etiquetas de profundidad m
- En el tránsito entre R1 y R[n-1] el paquete P nunca tiene una pila de profundidad menor que m
- Ri transmite P a R[i+1] empleando MPLS, es decir, usando la etiqueta superior de la pila
- Equipos entre Ri y R[i+1], al tomar decisiones de reenvío no se basan en la etiqueta de nivel m ni en cabecera de nivel de red
- LSP egress node será cuando se tome la decisión en función de etiqueta de nivel m-k (k>0) o de métodos “ordinarios”



PHP

- *Penultimate Hop Popping*
- El objetivo es que el paquete P llegue a Rn, luego la etiqueta ha cumplido su función cuando P llega a R[n-1]
- La etiqueta puede ser retirada de la pila en el penúltimo nodo
- La definición anterior de hecho permitía que entre R[n-1] y Rn el paquete llevara una pila de profundidad m-1
- Sin PHP, Rn debe hacer dos búsquedas, una para retirar la etiqueta de profundidad m y otra para tomar la decisión de reenvío
- Con PHP:
  - R[n-1] retira la etiqueta de nivel m y reenvía hacia Rn
  - Rn tendrá como superior la etiqueta de nivel m-1 o si m=1 la cabecera original para tomar la decisión de reenvío
  - Rn no necesita ser un LSR



Túneles

Túneles en IP

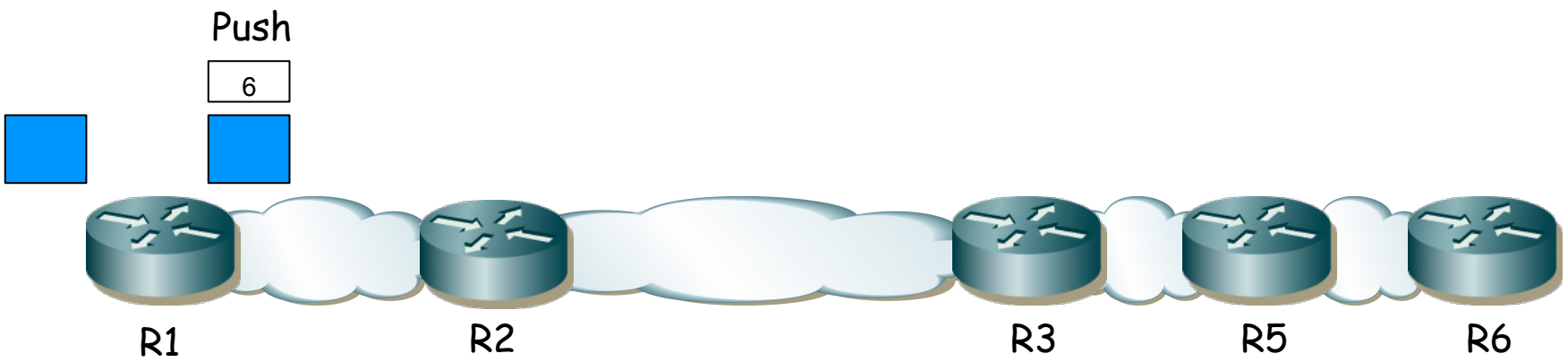
- Para asegurarse que un paquete vaya de un router Ru a otro Rd
- Cuando los routers no son adyacentes
- Ru por ejemplo encapsula el paquete IP dentro de otro paquete IP con dirección destino la de Rd
- Esto crea un túnel de Ru a Rd
- *"Hop-by-Hop Routed Tunnel"*: sigue camino salto a salto de Ru a Rd
- *"Explicitly Routed Tunnel"*: no sigue el camino salto a salto, por ejemplo con source routing

LSP Tunnels

- Se puede implementar un túnel con un LSP
- Los paquetes a enviar por el túnel constituyen un FEC
- "*Hop-by-Hop Routed LSP Tunnel*"
- "*Explicitly Routed LSP Tunnel*"

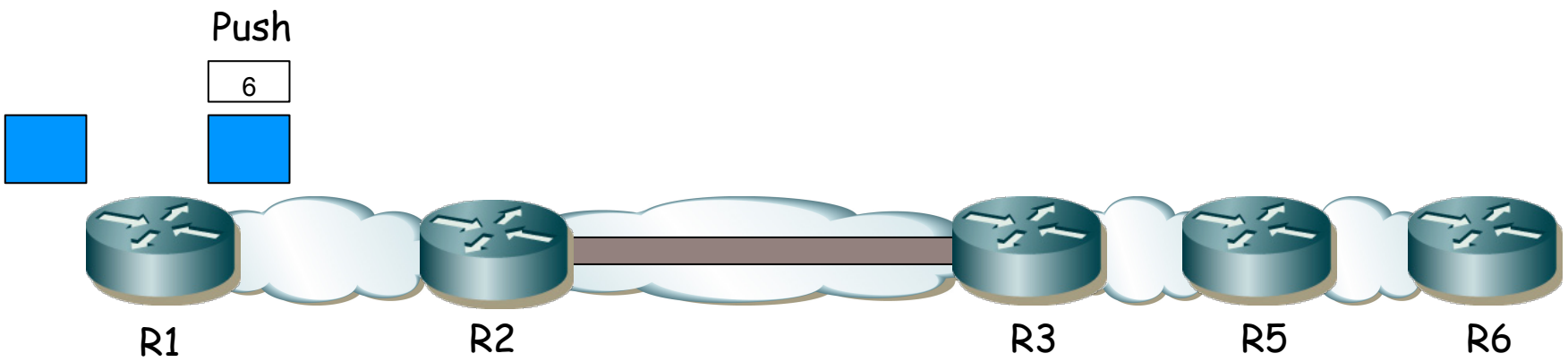
LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4, R5, R6>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- (…)



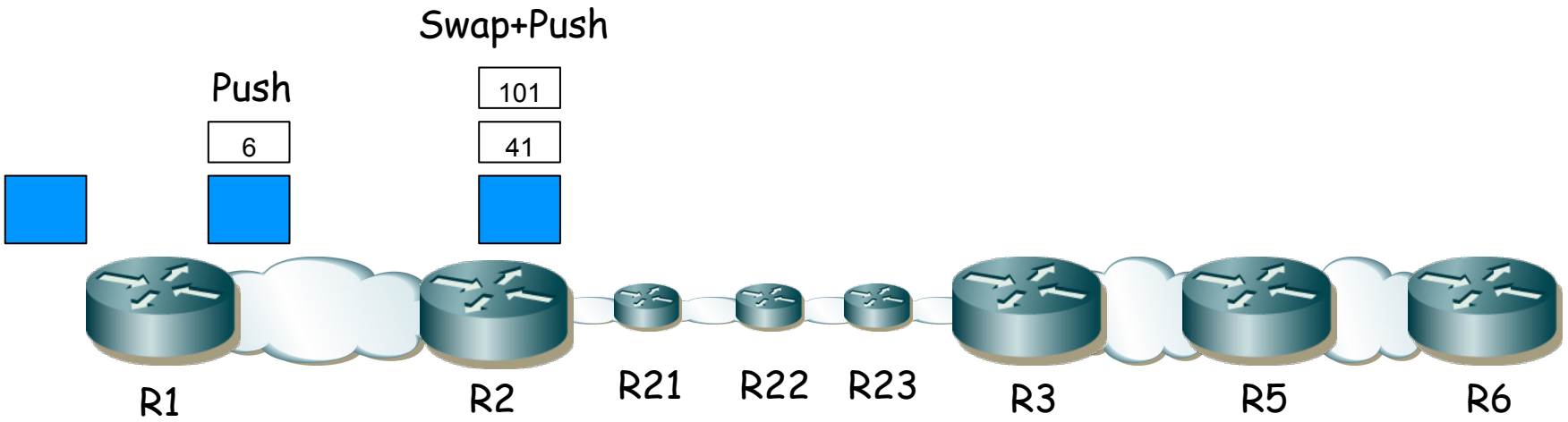
LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4, R5, R6>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- R2 y R3 no están directamente conectados
- R2 y R3 son “vecinos” mediante un túnel LSP (… …)



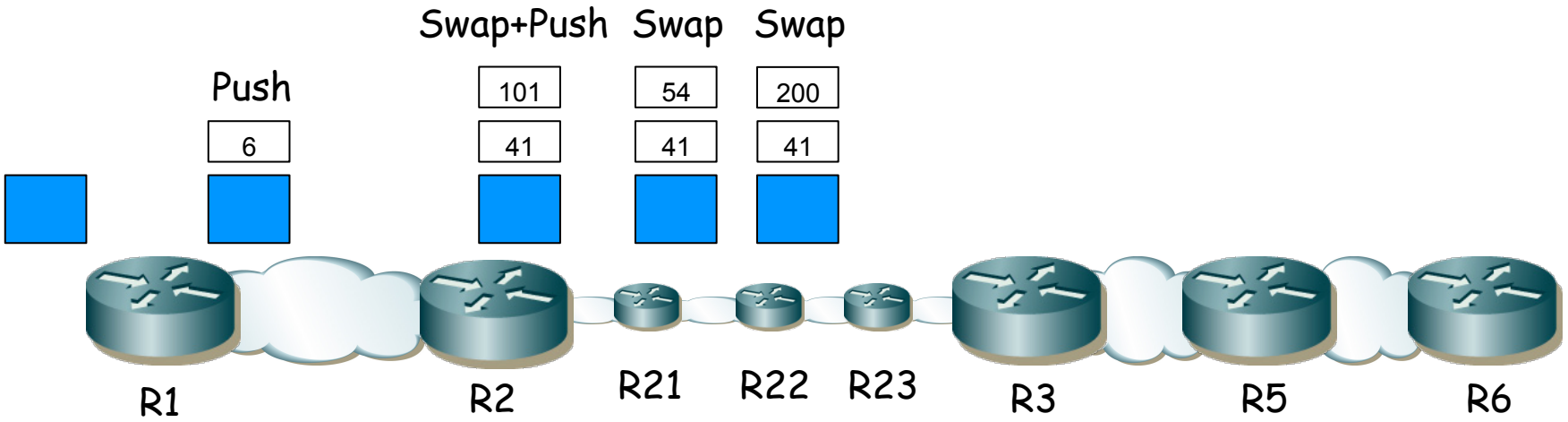
LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4, R5, R6>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- R2 y R3 no están directamente conectados
- R2 y R3 son “vecinos” mediante un túnel LSP
- R2 no solo hace swap de etiqueta sino también push de una nueva para el túnel
- (…)



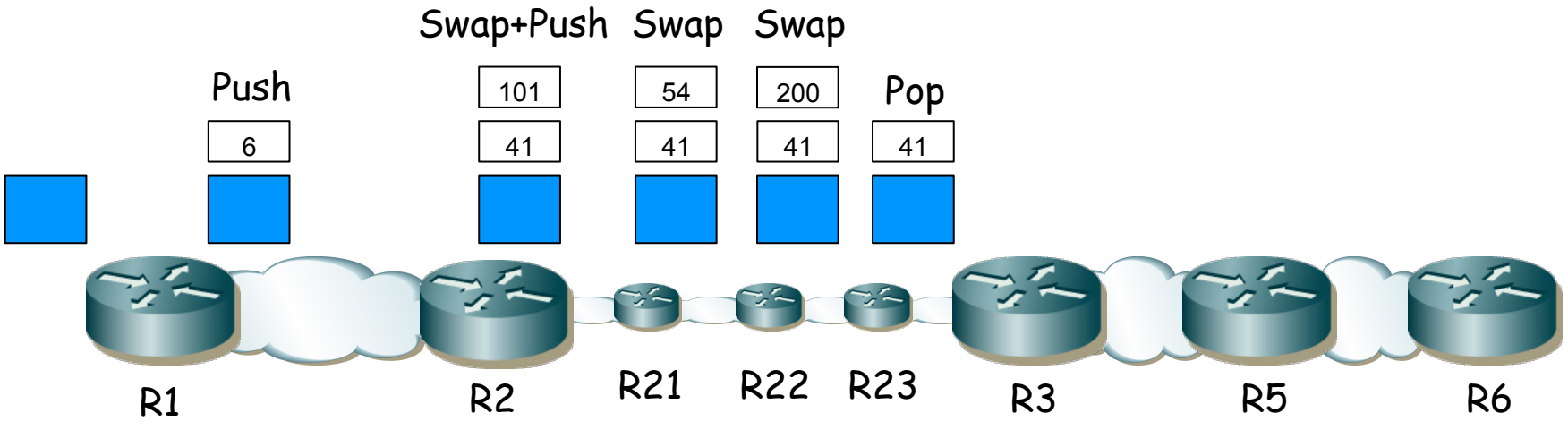
LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4, R5, R6>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- R2 y R3 no están directamente conectados
- R2 y R3 son “vecinos” mediante un túnel LSP
- R2 no solo hace swap de etiqueta sino también push de una nueva para el túnel
- R21 y R22 conmutan en función de la etiqueta de nivel 2
- (…)



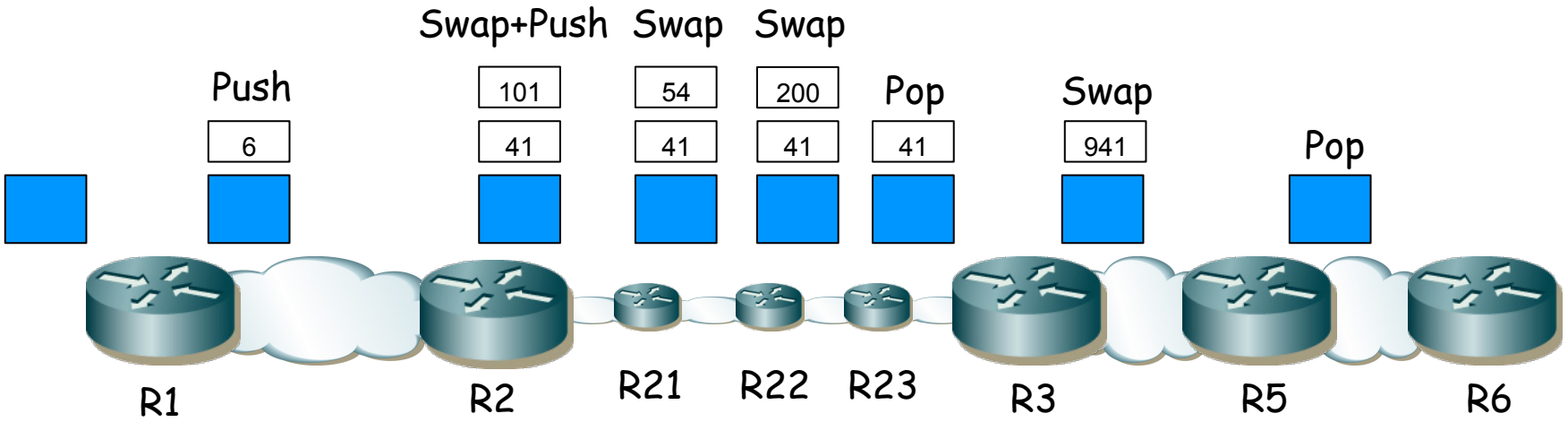
LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4, R5, R6>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- R2 y R3 no están directamente conectados
- R2 y R3 son “vecinos” mediante un túnel LSP
- R2 no solo hace swap de etiqueta sino también push de una nueva para el túnel
- R21 y R22 conmutan en función de la etiqueta de nivel 2
- La etiqueta de nivel 2 es retirada por R23 (PHP) y reenvía el paquete a R3
- (…)



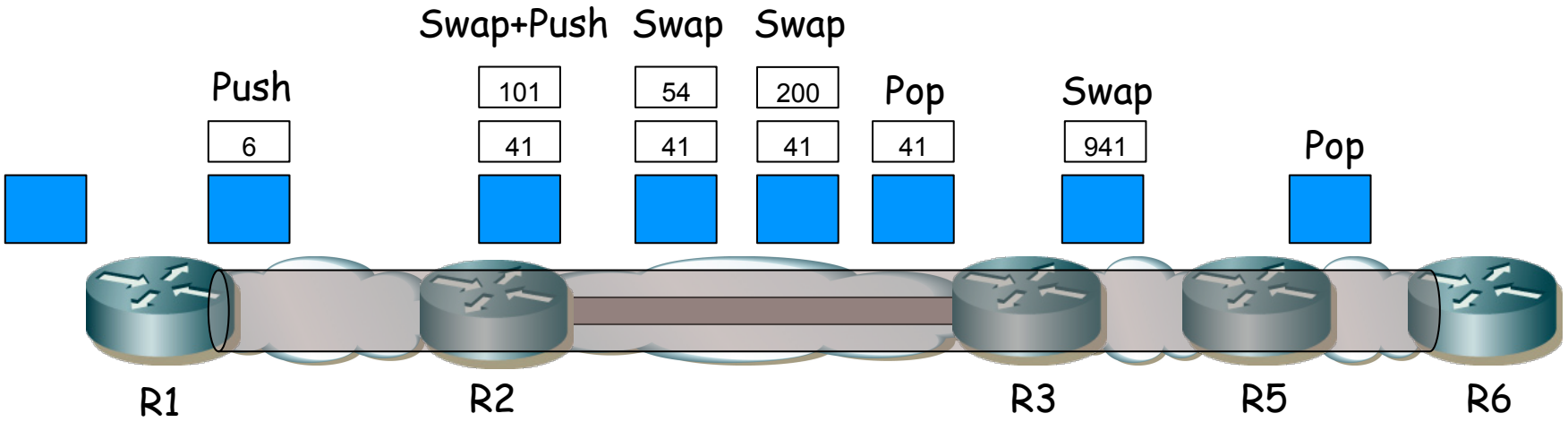
LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4, R5, R6>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- R2 y R3 no están directamente conectados
- R2 y R3 son “vecinos” mediante un túnel LSP
- R2 no solo hace swap de etiqueta sino también push de una nueva para el túnel
- R21 y R22 conmutan en función de la etiqueta de nivel 2
- La etiqueta de nivel 2 es retirada por R23 (PHP) y reenvía el paquete a R3
- R3 recibe el paquete con una sola etiqueta (ha salido del túnel)
- R3 hace Swap
- R4 elimina la etiqueta (PHP) y envía a R5
- Se pueden anidar túneles de esta manera sin límite de profundidad (… …)



LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4, R5, R6>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- R2 y R3 no están directamente conectados
- R2 y R3 son “vecinos” mediante un túnel LSP
- R2 no solo hace swap de etiqueta sino también push de una nueva para el túnel
- R21 y R22 conmutan en función de la etiqueta de nivel 2
- La etiqueta de nivel 2 es retirada por R23 (PHP) y reenvía el paquete a R3
- R3 recibe el paquete con una sola etiqueta (ha salido del túnel)
- R3 hace Swap
- R4 elimina la etiqueta (PHP) y envía a R5
- Se pueden anidar túneles de esta manera sin límite de profundidad (… …)



Selección de ruta

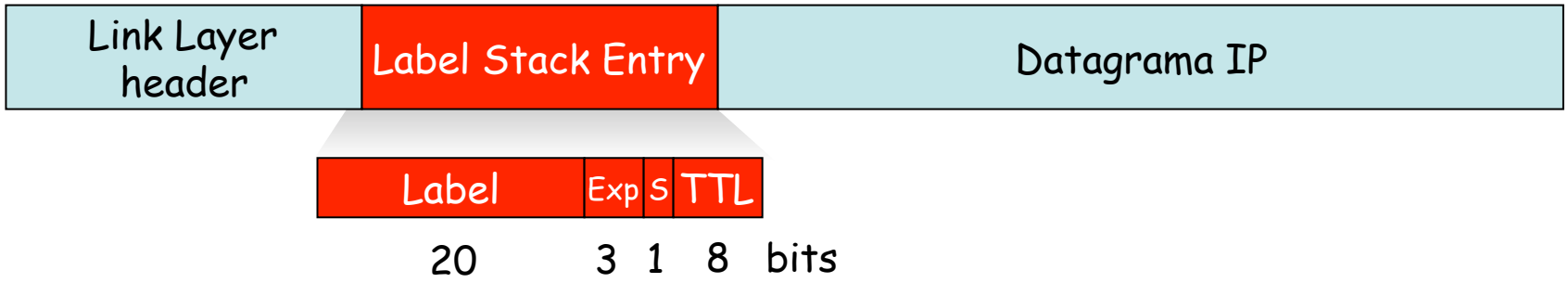
1. Hop by hop routing
   - Cada nodo selecciona de forma independiente el siguiente salto para cada FEC
   - "hop by hop routed LSP"
2. Explicit routing
   - Un LSR (normalmente el ingress o el egress) especifica los LSRs del LSP
   - Puede especificar solo algunos de los LSRs del LSP
   - Si un solo LSR especifica el LSP entero se habla de "*strictly explicitly routed*"
   - Si un solo LSR especifica solo algunos de los LSRs del LSP se habla de "*loosely explicitly routed*"
   - Se especifica al establecer las etiquetas
   - Más eficiente que source routing IP que contiene el camino cada paquete

TTL en IP

- Empleado para:
  - Acotar el efecto de bucles
  - Limitar el alcance de un paquete (traceroute)
- Un paquete en un LSP debería (SHOULD) salir del mismo con el mismo valor de TTL que hubiera tenido de no haber empleado MPLS
- El número de LSRs atravesados debe reflejarse en el TTL del paquete
- Si se emplea un “shim” header:
  - Debe tener un TTL
  - Inicialmente debería tener el valor del TTL del paquete
  - Debería decrementarse en cada LSR
  - Debería copiarse a la salida al paquete original
- Si la etiqueta se codifica en una cabecera de nivel de enlace:
  - Un segmento de LSP que no soporta llevar el TTL se llama “non-TTL LSP segment”
  - Al salir de este segmento debería actualizarse el TTL del paquete
  - Se puede lograr propagando la longitud del LSP al ingress y que éste decremente el TTL *ANTES* de enviar el paquete al segmento “non-TTL”
  - Si se ve que el TTL se agotará, no se conmuta con etiqueta el paquete (se podría hacer reenvío salto a salto convencional)

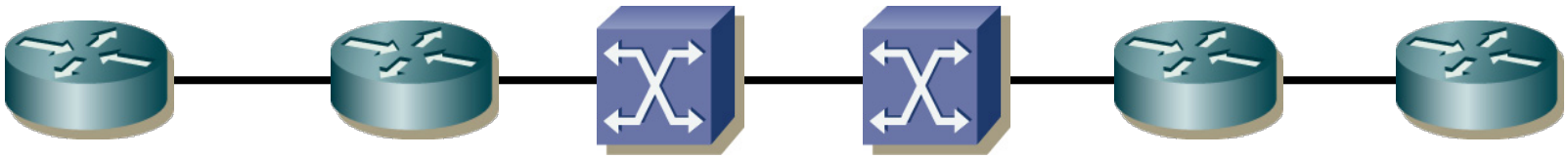
"Shim" header

- RFC 3032 "MPLS Label Stack Encoding"
- Forma de codificación empleada por un LSR para enlaces PPP o LAN
- En general independiente del protocolo encapsulado (con particularidades para IPv4 e IPv6)
- "Label Stack" como una secuencia de "label stack entries"
- La etiqueta superior de la pila es la primera tras la cabecera de nivel de enlace
- Contenido de la entrada:
  - Label : la etiqueta en si (valores 0-15 reservados)
  - Exp : "Experimental Use", ahora TC "Traffic Class" (RFC 5462) empleado para CoS
  - S : "Bottom of Stack", está a 1 en la última entrada de la pila
  - TTL : Time to Live
- Protocolo contenido debe ser acordado o inferirse de la última etiqueta



ATM-LSRs

- RFC 3035 “MPLS using LDP and ATM VC Switching”
- Conmutadores ATM usados como LSRs
- Emplean protocolos de encaminamiento de nivel de red (tipo OSPF, IS-IS) y no los específicos de ATM
- La etiqueta viaja en el VCI o en el VPI/VCI
- En general no se soporta multipunto-a-punto y multipunto-a-multipunto
- No soporta decremento del TTL
- El protocolo transportado va con shim header como únicos datos de la PDU AAL5
- La etiqueta superior estará a 0 porque va en el VPI/VCI (lleva ahí TTL y CoS)
- Para detectar bucles se implementa LDPV (*Loop Detection via Path Vectors*)
- LDPV permite detectar bucles más rápido que simplemente usar el TTL, a costa de mayor sobrecarga



Transporte de MPLS

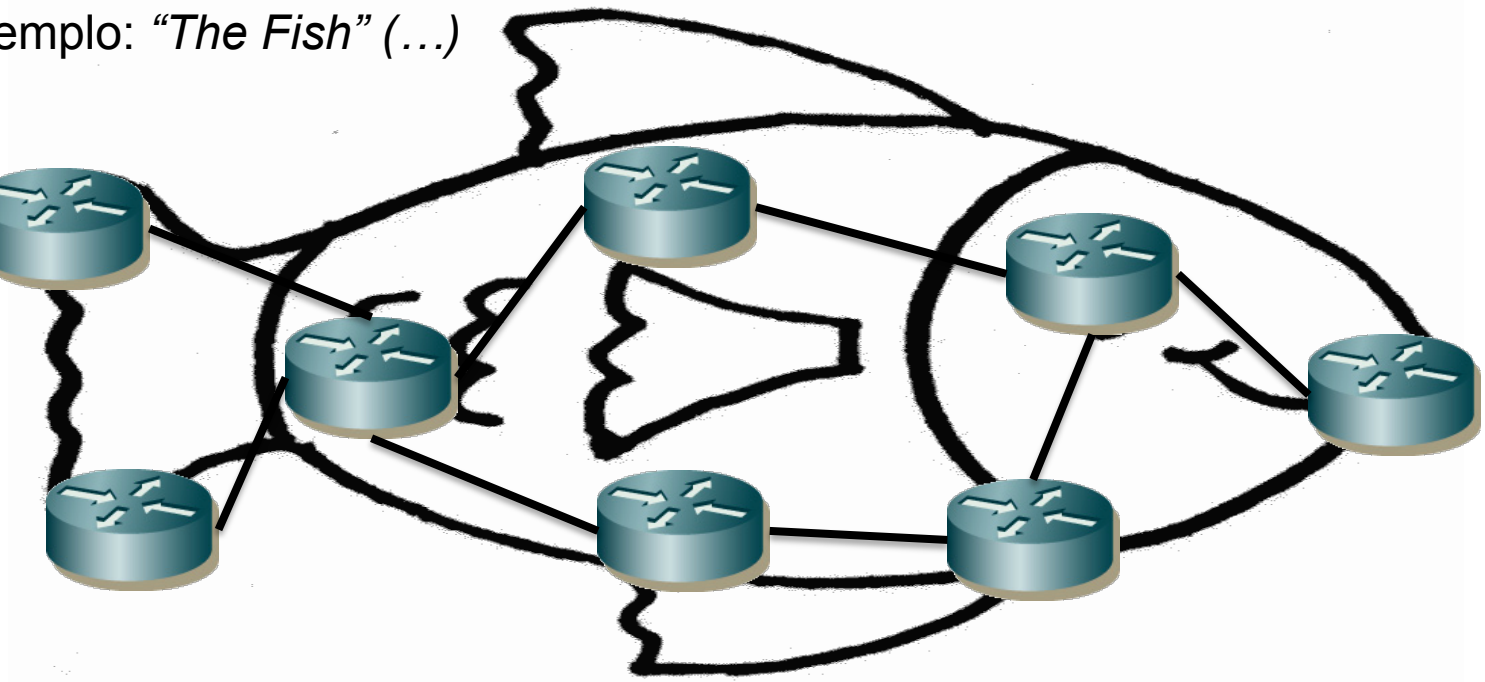
- Sobre ATM (ya hemos visto)
- Sobre PPP (campo protocolo 0x0281 y 0x0283)
- Sobre Ethernet (Ethertypes 0x8847 y 0x8848)
- Sobre HDLC
- Sobre Frame Relay

Layer 2 sobre MPLS

- RFC 4905 “Encapsulation Methods for Transport of Layer 2 Frames over MPLS Networks”
- y RFC 4906 “Transport of Layer 2 Frames Over MPLS”
  - Frame Relay
  - ATM (celdas o PDUs AAL5)
  - Ethernet (simple o 802.1Q)
  - PPP
  - HDLC

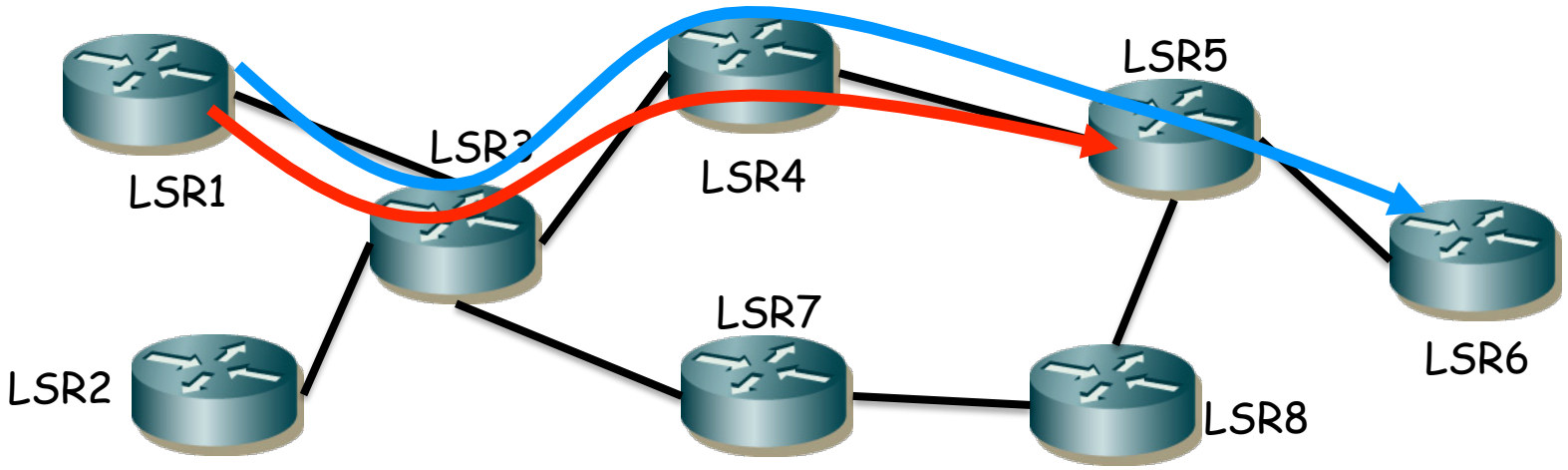
Traffic Engineering

- Network Engineering
  - Construir la red para transportar el tráfico esperado (¡predecir!)
- Traffic Engineering
  - Manipular el tráfico para encajar en la red
  - Prevenir enlaces congestionados y otros infrautilizados
- No podemos contar con predecir los patrones de tráfico
- Seguramente tendremos una red con BW simétricos pero flujos asimétricos
- RFC 2702 - Requirements for Traffic Engineering over MPLS
- Ejemplo: *"The Fish" (…)*



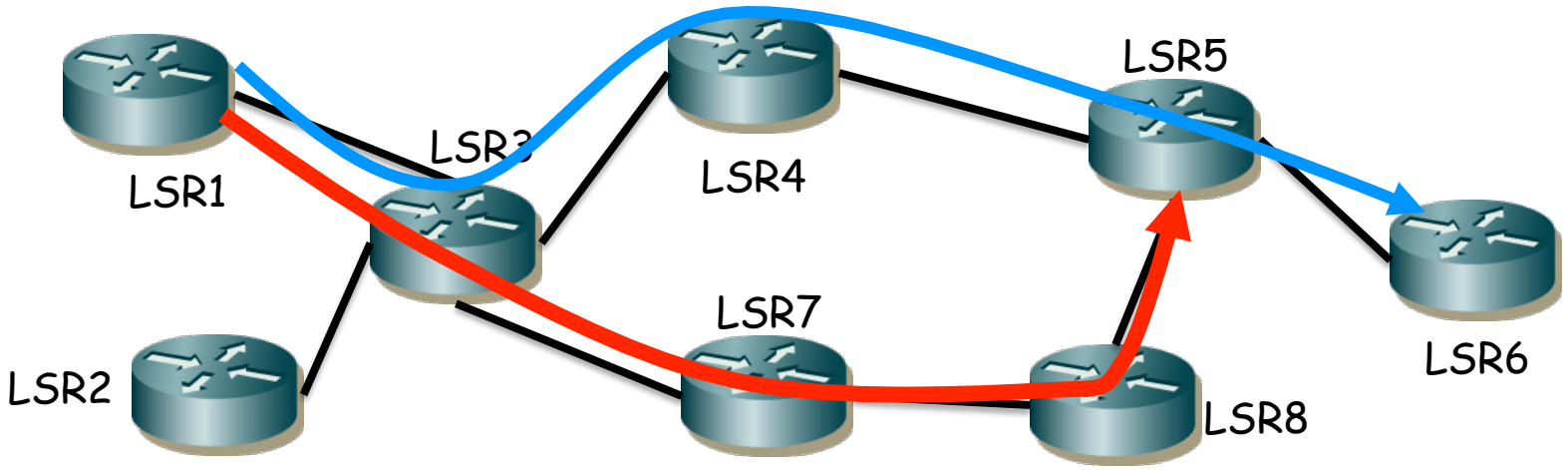
Ejemplo

- LSR5 está en el Shortest Path (SP) de LSR1 a LSR6
- Entonces el SP de LSR1 a LSR5 es parte del camino a LSR6 (principio de optimalidad) (…)



Ejemplo

- LSR5 está en el Shortest Path (SP) de LSR1 a LSR6
- Entonces el SP de LSR1 a LSR5 es parte del camino a LSR6 (principio de optimalidad)
- Querríamos poder emplear rutas alternativas (…)



Explicit routing

- Ingress LSR decide el camino
- Emplea CSPF “Constrained Shortest Path First”
- ¿Cómo?
  - Información: Link State (OSPF-TE, ISIS-TE)
  - Eliminar los enlaces que no cumplen las restricciones
  - Buscar camino más corto en la topología resultante
  - Cambios deben propagarse (por ejemplo BW ocupado)
  - Señalización para LSP con reserva de recursos:
    - CR-LDP: RFC 3212 “Constraint-Based LSP Setup using LDP”
      - Señaliza PDR (Peak Data Rate), PBS (Peak Burst Size), CDR (Commited Data Rate), CBS (Commited Burst Size), EBS (Excess Burst Size)
      - Parámetros para token buckets
    - RSVP-TE: RFC 3209 “Resource Reservation Protocolo – Traffic Engineering”
      - Añade distribución de etiquetas a RSVP

Resumen

- Conmutación por etiquetas
- Transporte de paquetes IP pero también de otros protocolos
- Empleable sobre múltiples tecnologías L2
- Se crean túneles que se pueden anidar
- Permite hacer ingeniería de tráfico