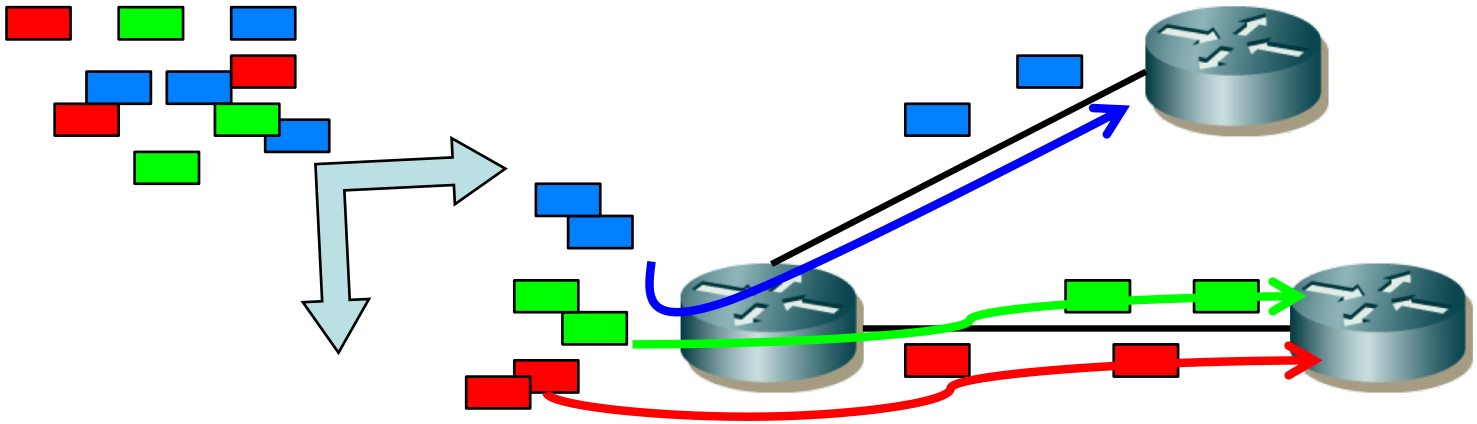
MPLS

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

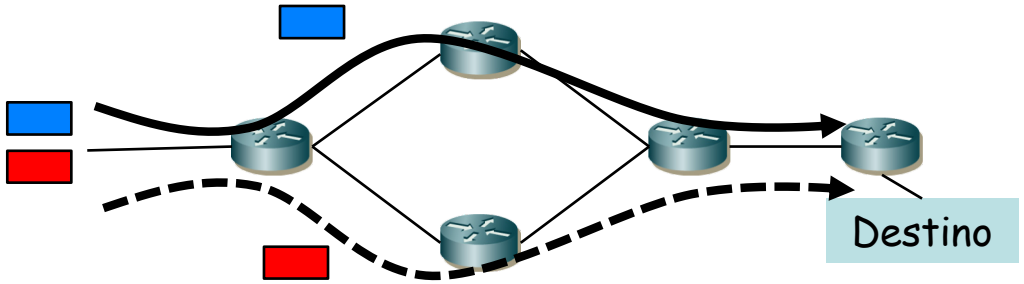
FEC

- *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
- (…)



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 deciden en base al menor coste de camino (SPF)
- (…)



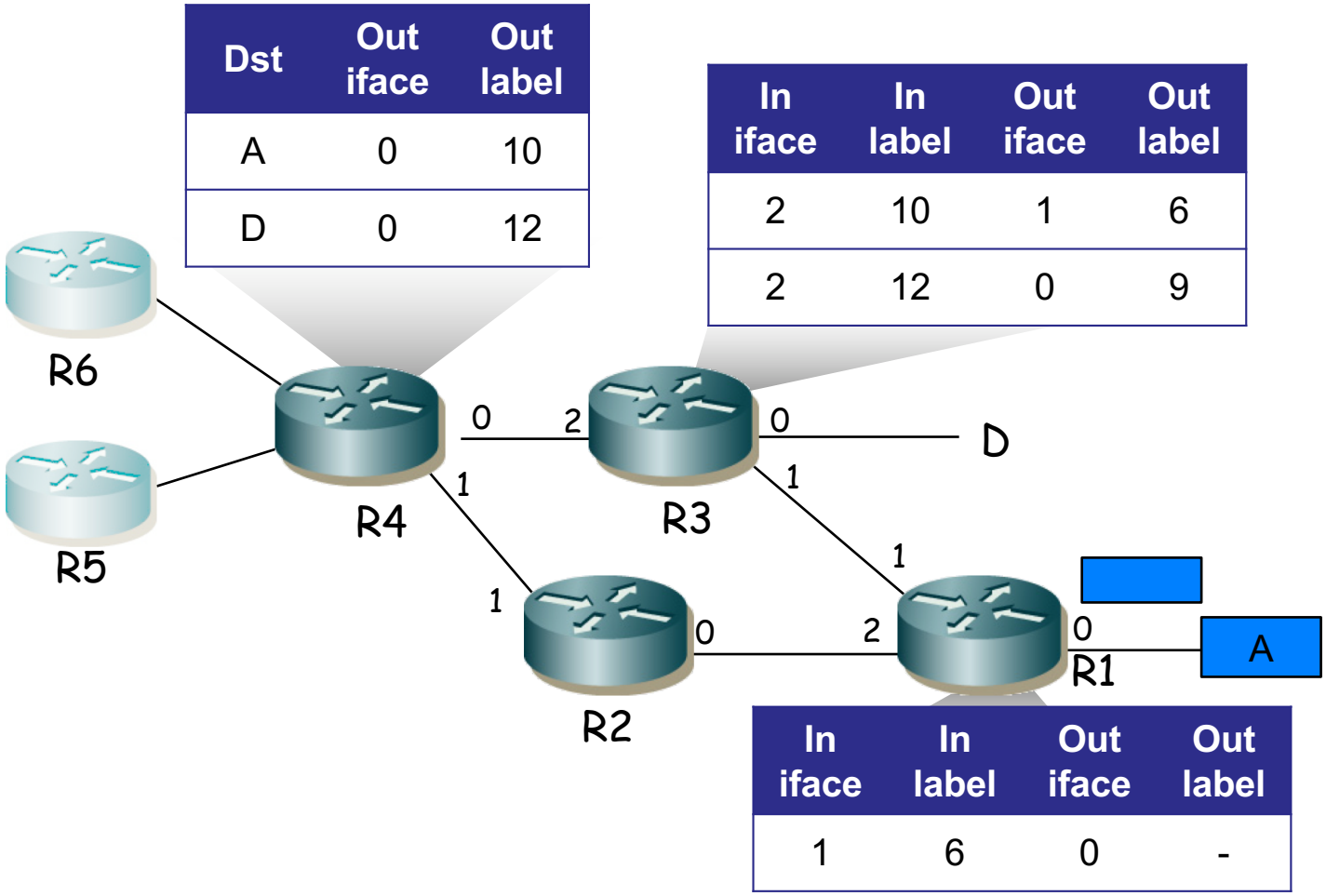
FEC

- *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

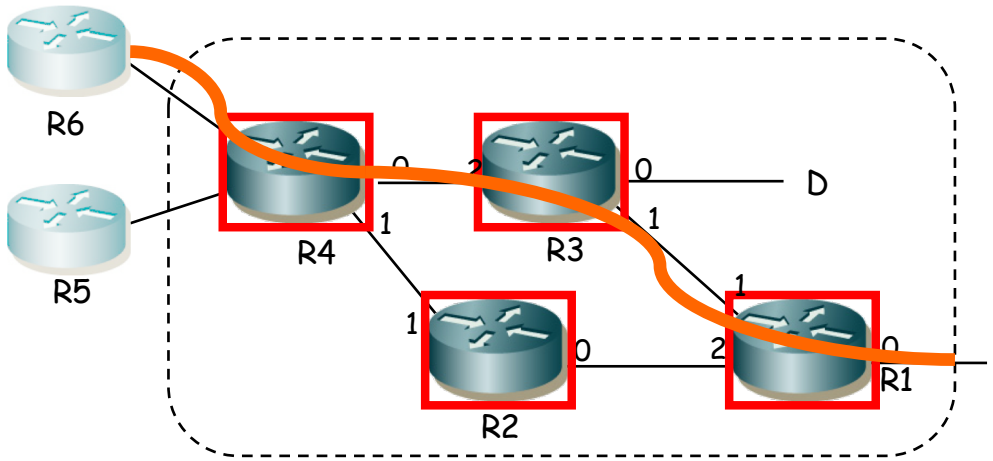
- Inicialmente para ahorrarse el cálculo del *Longest-prefix-match* en los equipos de core
- Hoy en día para hacer *Traffic Engineering*
- Conmutación de paquetes, pero circuitos virtuales
- Heredero de ATM pero con paquetes de tamaño variable
- Inicialmente sin QoS

MPLS "forwarding"



Terminología

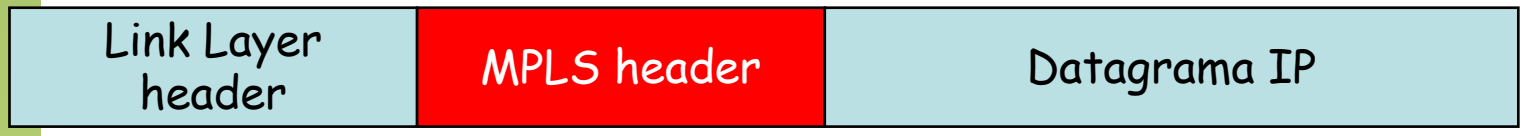
- "MPLS domain": conjunto contiguo de nodos MPLS bajo una misma administración
- "MPLS ingress node": nodo frontera de un dominio en su tarea como entrada de tráfico al mismo
- "MPLS egress node": nodo frontera de un dominio en su tarea como salida de tráfico del mismo
- "Label": etiqueta numérica, corta, 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

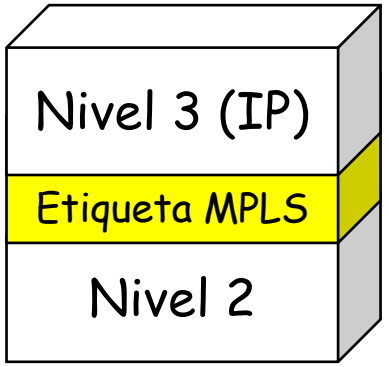


MPLS: Label Stack

Label Stack

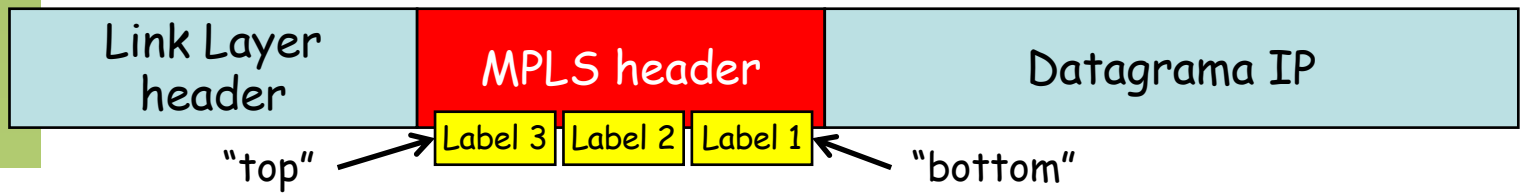
- La localización de la etiqueta depende de la tecnología que transporte los paquetes
- Una posibilidad es emplear un "*shim header*" entre cabecera del nivel de enlace y del protocolo transportado
- Hay otras opciones, por ejemplo si el transporte es sobre ATM se emplea el VPI/VCI como etiqueta
- A 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*) (…)

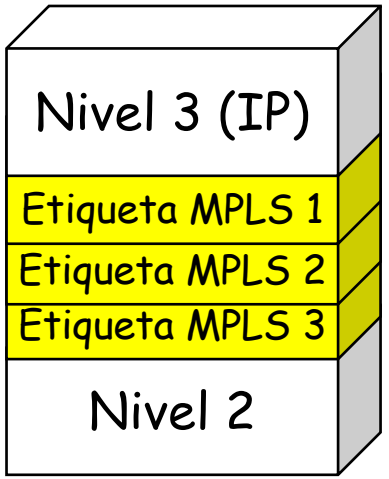




Label Stack

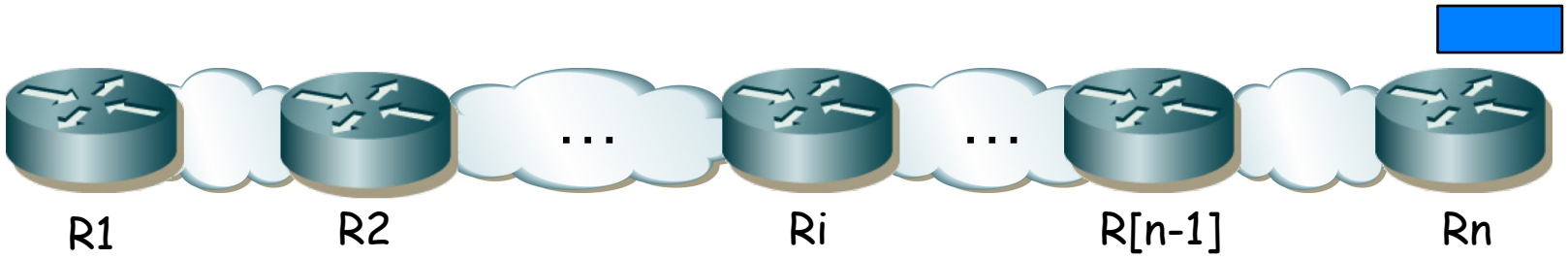
- La parte “superior” (“top”) de la pila comienza a continuación de la cabecera de nivel de enlace
- La parte “inferior” (“bottom”) de la pila está junto a la cabecera de nivel de red
- El procesado se basa siempre en la etiqueta exterior (“top”)
- Un paquete sin etiquetar tiene profundidad 0 de pila
- En un LSR se puede emplear espacio de etiquetas:
  - Por interfaz
  - Por plataforma





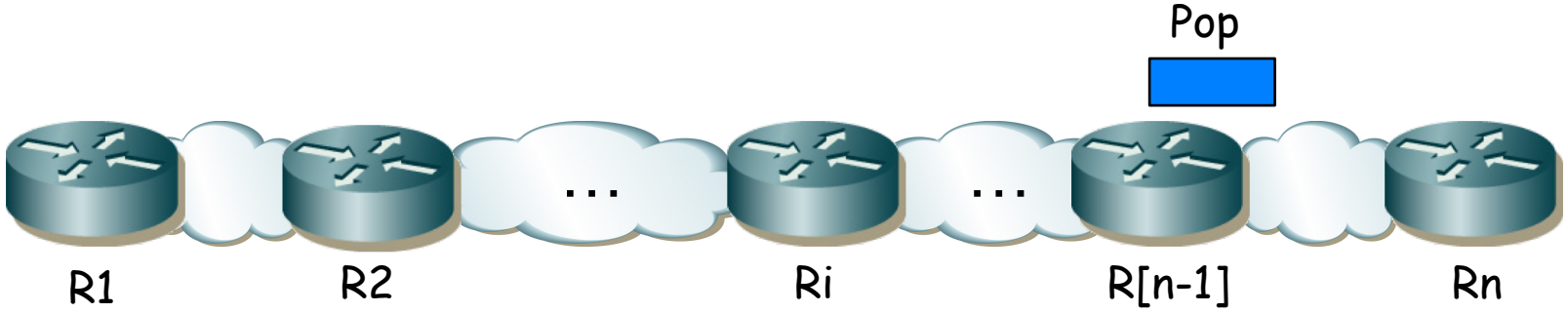
LSP de nivel m

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

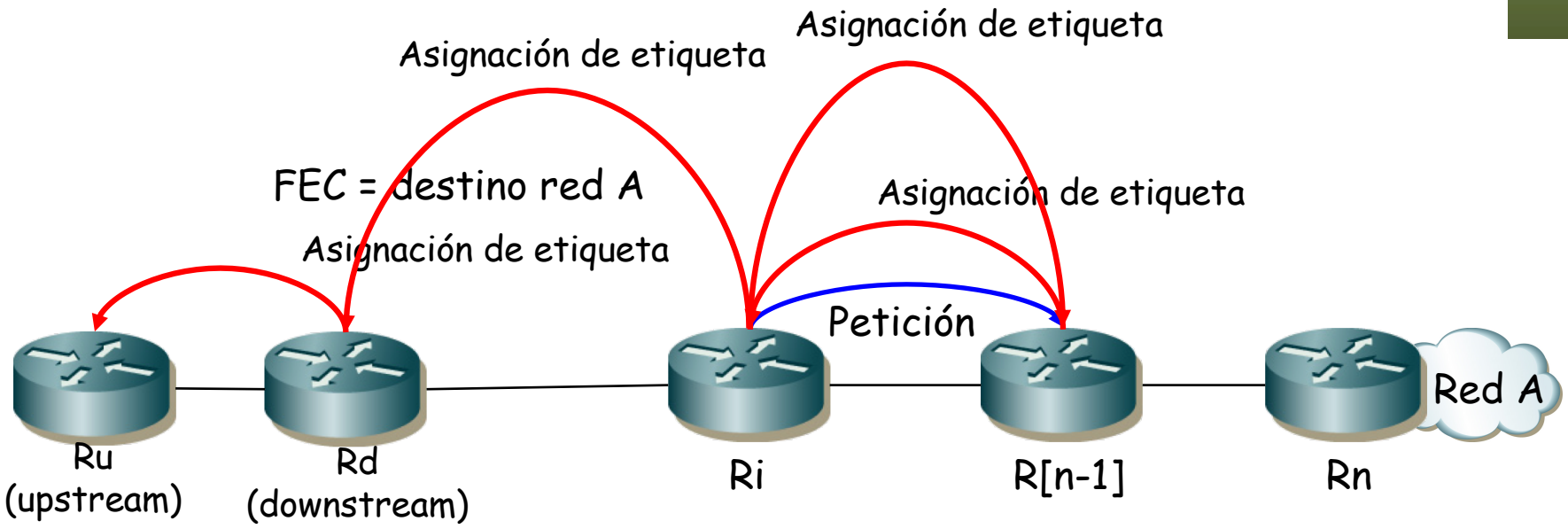


Label distribution

- Empleando un protocolo ya existente
  - Sencillo para protocolos DV
  - Complicado para protocolos LS
  - No se han cambiado IGPs para esto
  - Sí se ha adaptado BGP-4
  - RSVP-TE “Resource Reservation Protocolo – Traffic Engineering” RFC 3209 en realidad para TE
- Creando un protocolo independiente para ello
  - LDP “Label Distribution Protocol” RFC 5036
  - Es tanto el nombre del protocolo como de la categoría

Label distribution

- La etiqueta para un FEC la asigna el LSR downstream e informa al upstream
- LSR que usan un LDP para intercambiar Label/FEC son "LDP peers"
- La asignación puede ser
  - *Downstream-on-demand*: LSR upstream pide la asignación al downstream (...)
  - *Unsolicited downstream:* Nodo envía asignación por su propia iniciativa (...)



LSP Control

- Algunos FECs pueden corresponder con prefijos distribuidos mediante protocolos de encaminamiento dinámico
- La creación de LSPs para estos FECs se puede hacer de dos formas:
  - *Independent LSP Control*
    - Cada LSR, al reconocer un FEC, toma una decisión independiente de asociar una etiqueta al FEC
    - LSR distribuye la asociación a sus “LDP peers”
  - *Ordered LSP Control*
    - Un LSR solo asocia una etiqueta a un FEC si es el egress LSR para ese FEC o si ha recibido una asociación de su siguiente salto
    - Necesario para hacer Traffic Engineering
- Son interoperables pero si no usan todos *Ordered Control* el efecto final es como si usaran *Independent Control*

Agregación

- MPLS soporta agregación
- Un conjunto de FECs con etiquetas diferentes, al llegar a un nodo forman un solo FEC con una sola etiqueta
- *Label Merging*
- Un equipo puede no soportarlo (por ejemplo usando conmutadores ATM como LSRs se entremezclan celdas de diferentes PDUs)
- Se puede hablar de un “Multipoint-to-Point LSP Tree”

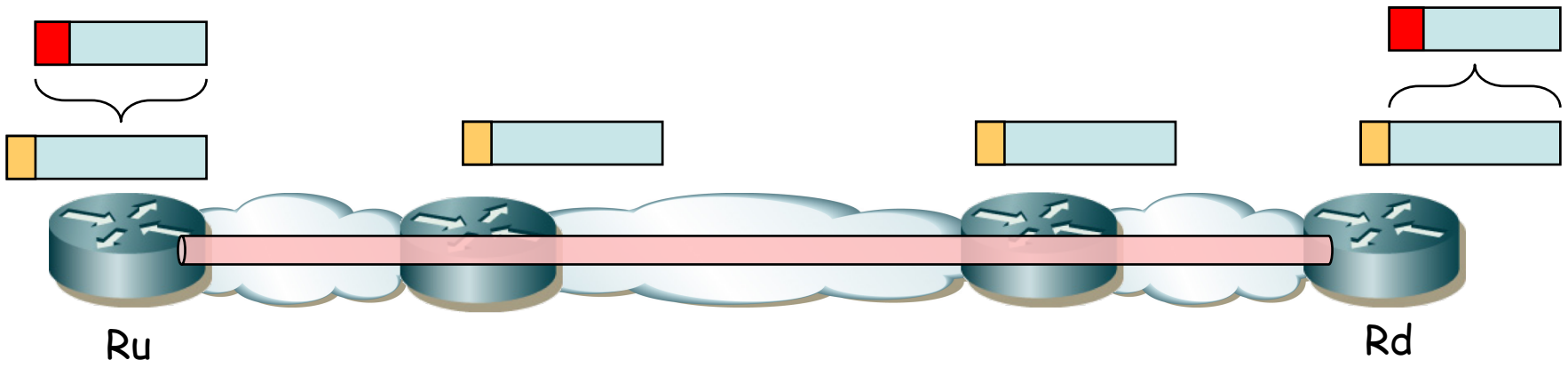
Selección de ruta

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

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
- (...)



Túneles

Túneles en IP

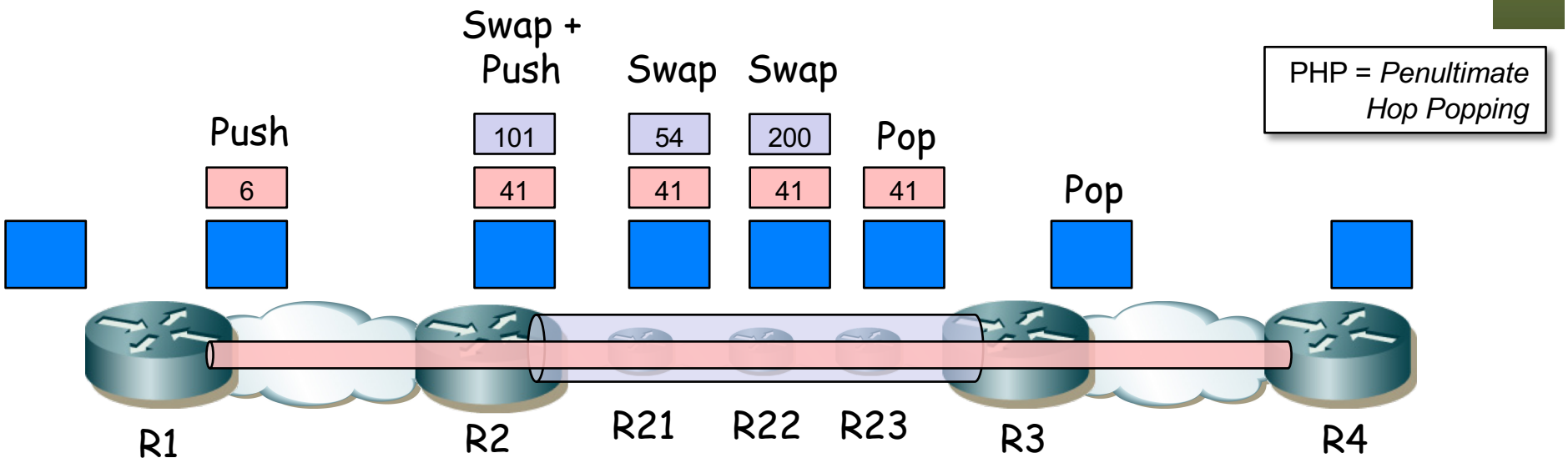
- Para asegurarse que un paquete vaya de un router Ru a otro Rd
- Cuando los routers no son adyacentes
- Ru por ejemplo encapsula el paquete IP dentro de otro paquete IP con dirección destino la de Rd
- Esto crea un túnel de Ru a Rd
- *"Hop-by-Hop Routed Tunnel"*: sigue camino salto a salto de Ru a Rd
- *"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*"
- Y un LSP se puede meter en un túnel (...)

LSP Tunnels dentro de LSPs

- Por ejemplo LSP <R1, R2, R3, R4>
- 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 conmuta 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 elimina la etiqueta (PHP) y envía a R4
- Se pueden anidar túneles de esta manera sin límite de profundidad

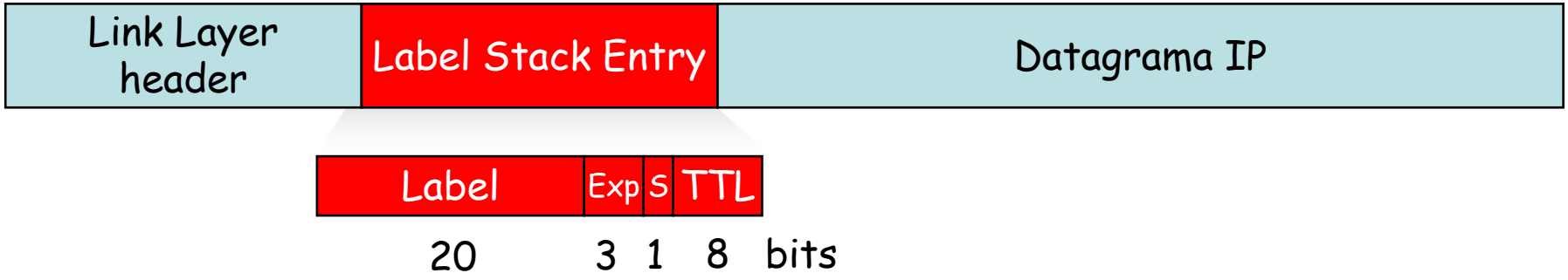


TTL en IP

- 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

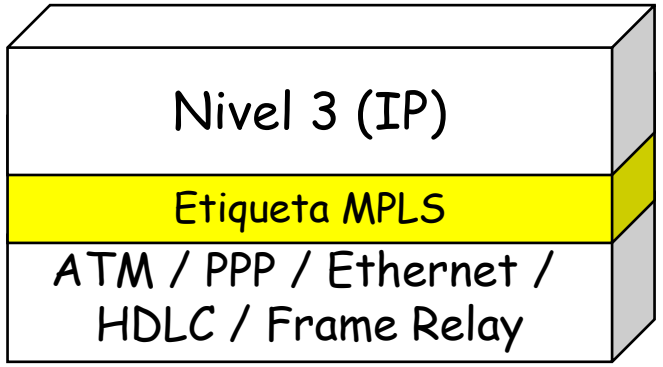
- RFC 3032 "MPLS Label Stack Encoding"
- Forma de codificación empleada por un LSR para enlaces PPP o LAN
- En general independiente del protocolo encapsulado (con particularidades para IPv4 e IPv6)
- "Label Stack" como una secuencia de "label stack entries"
- La etiqueta superior de la pila es la primera 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



MPLS: Transporte

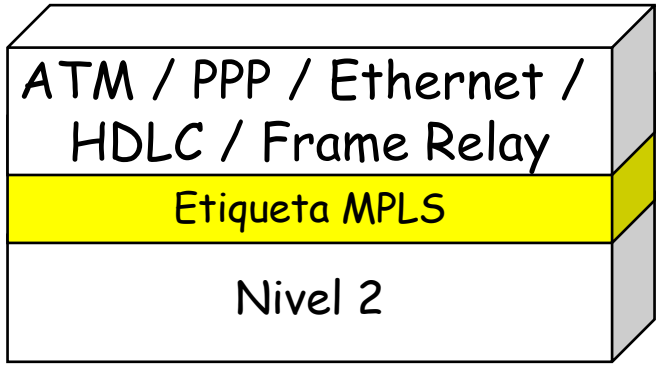
Transporte de MPLS

- Sobre ATM (Etiqueta en el VPI/VCI)
- 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
- Por supuesto, sobre ese nuevo layer 2, lo que queramos...



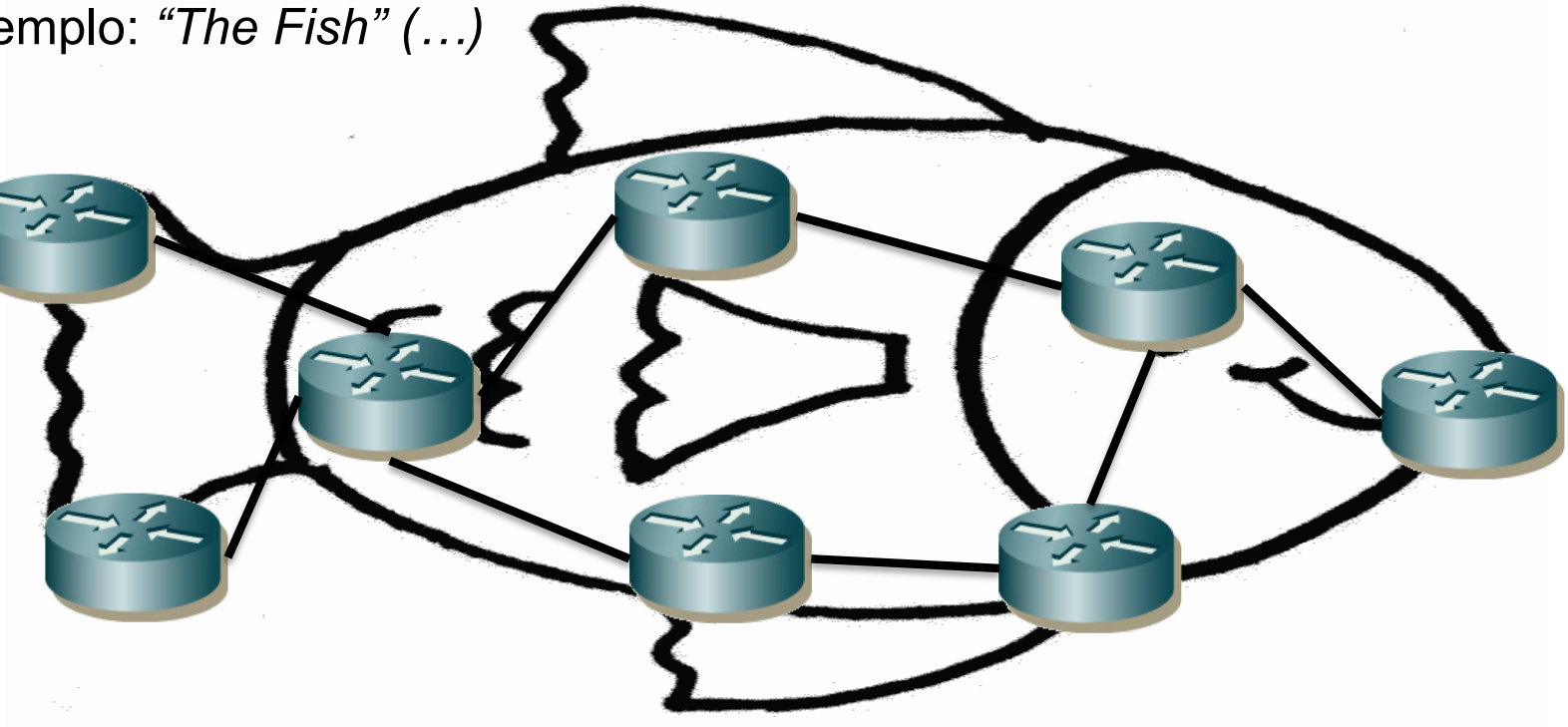
Traffic Engineering

Traffic Engineering (TE)

- RFC 3272 (Overview and Principles of Internet Traffic Engineering)
- *".. that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks."*
- *"[TE] encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic."*
- Existe desde las redes telefónicas clásicas
- Proceso:
  - *Measurement*: desde el nivel de paquete al de flujo, usuario, agregado de tráfico o red
  - *Modeling, Analysis and Simulation*
  - *Optimization*: desde real-time optimization a network planning

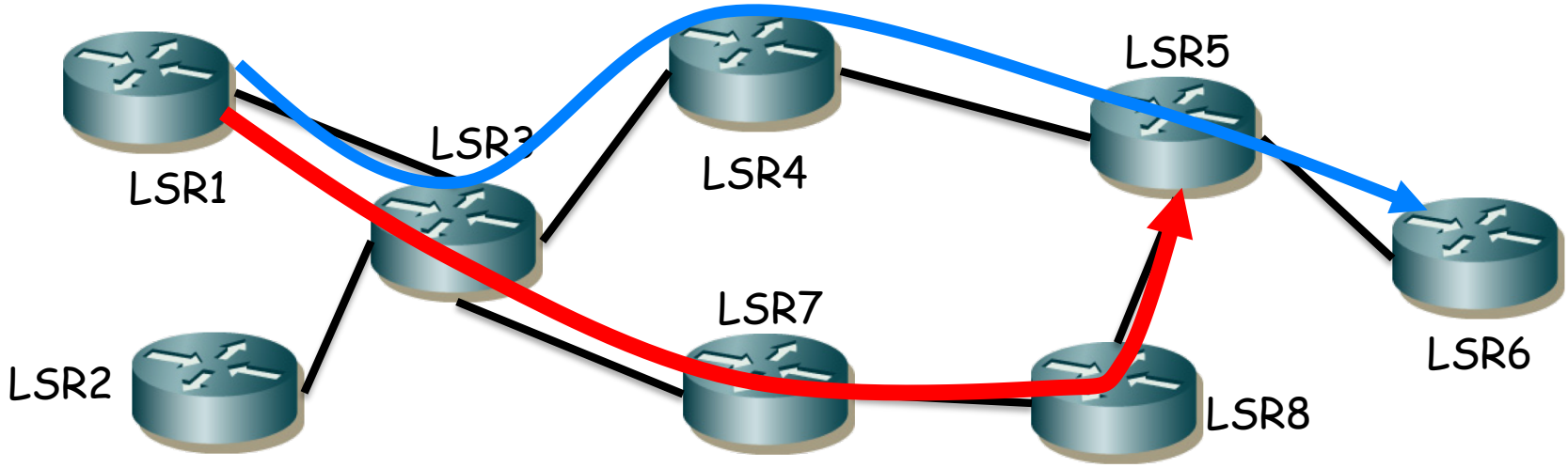
Traffic Engineering

- 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)
- Querríamos poder emplear rutas alternativas (…)

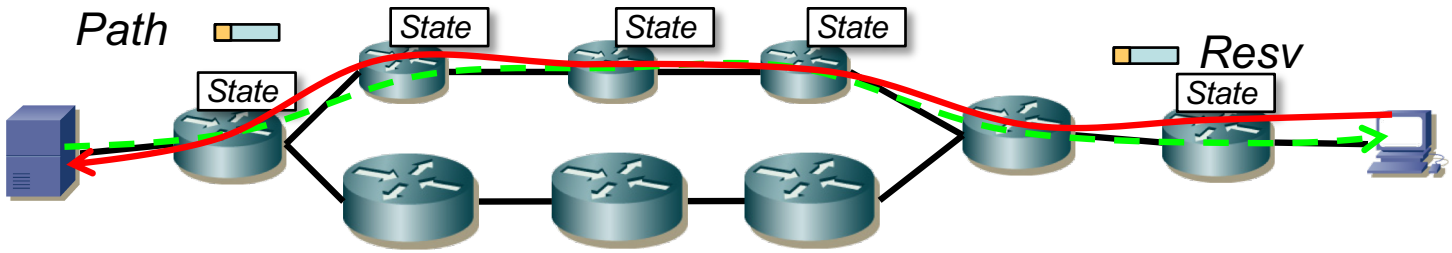


Explicit routing

- Ingress LSR (o un *Path Computation Element*) decide el camino mediante CBR (“Constraint-Based Routing”)
- En concreto 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”
      - (…)

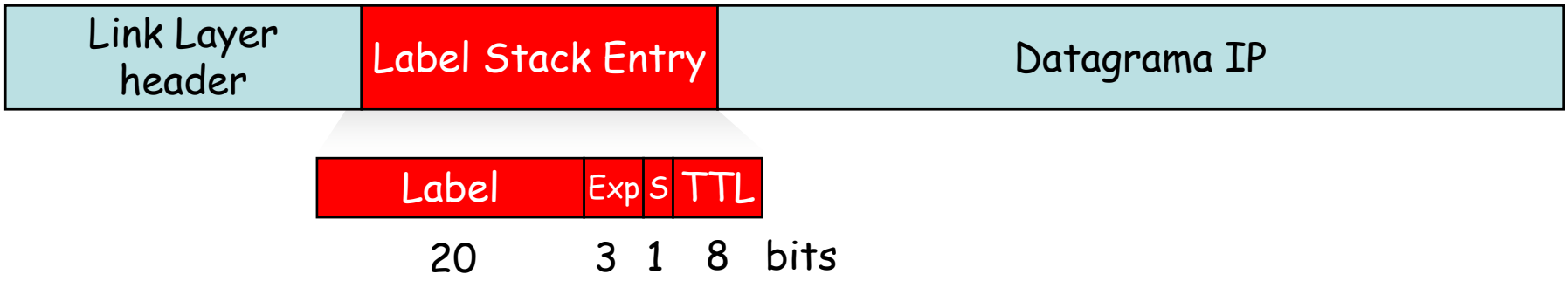
RSVP-TE

- RSVP
  - El mensaje *Path* lo envía la fuente y sigue la ruta calculada por los protocolos de encaminamiento
  - El mensaje *Resv* emplea ese estado para seguir el camino inverso
  - Si no hay recursos suficientes falla la reserva
  - Falla incluso si existe otro camino que sí disponga de recursos
- RSVP-TE añade un objeto EXPLICIT_ROUTE que permite especificar los nodos del camino deseado (*strict* o *loose*)
- Las reservas pueden ser compartidas entre varios LSPs, lo cual permite un *make-before-break*
- RSVP-TE añade la distribución de etiquetas y reserva bidireccional



MPLS : Exp field

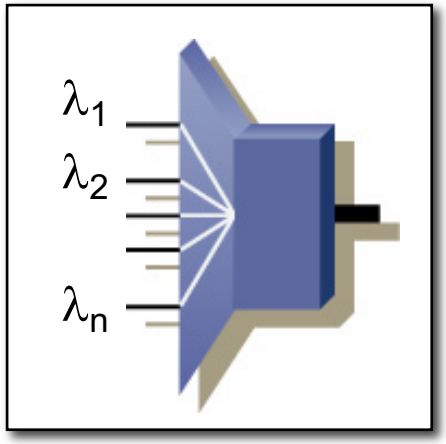
- 3 bits en una entrada de etiqueta
- Definido inicialmente (RFC 3032) para uso experimental
- RFC 5462 lo renombra a *"Traffic Class field"*

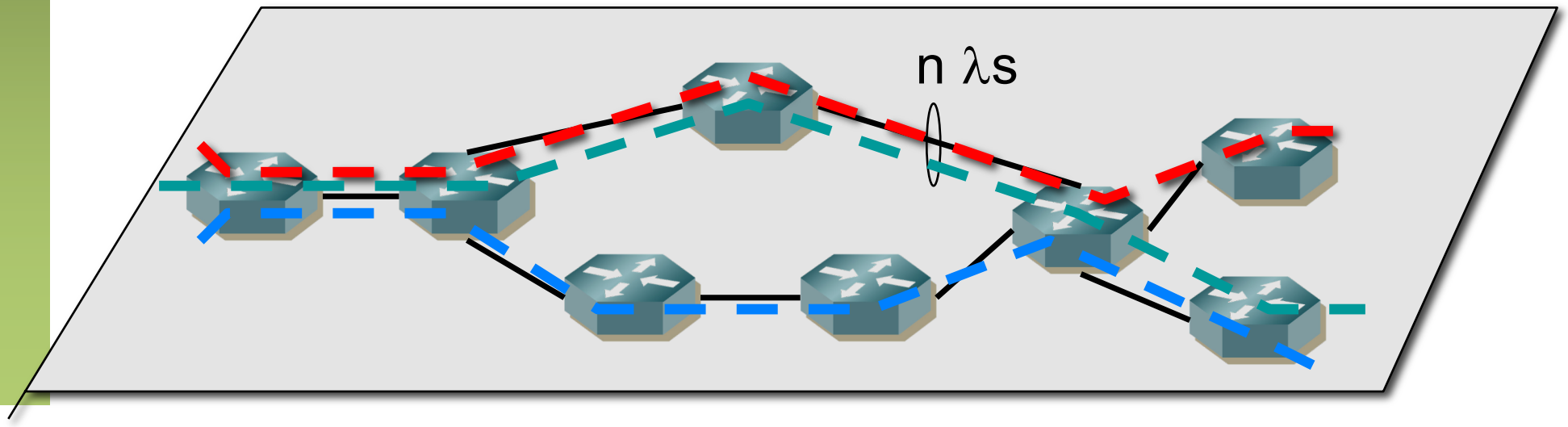


GMPLS

Lightpaths

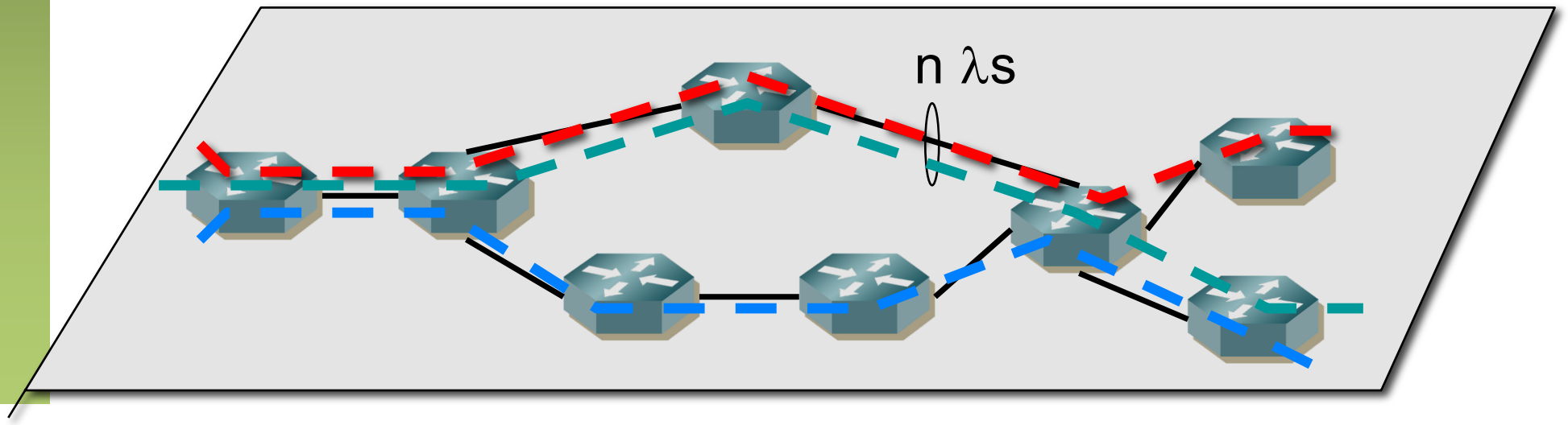
- DWDM
- Wavelength routing (. . .)
- OADM : Optical Add Drop Multiplexer
- ROADM : Reconfigurable OADM
- Con o sin conversión de longitud de onda





GMPLS

- *Generalized MultiProtocol Label Switching* (IETF)
- Aplicación de conceptos de MPLS a redes de transporte que **NO** son de conmutación de **paquetes**
- WDM funcionamiento similar a MPLS con fibra de entrada y wavelength (etiqueta) de entrada
- Inicialmente surgió con esa idea MPλS
- Se amplió para *fiber switching*, TDM, layer 2 switching, etc. ("Generalización")
- NO es reutilizable la parte de MPLS en que puede asignar etiquetas a entradas en tablas de rutas (LDP)
- SÍ aplican las soluciones para *Traffic Engineering*

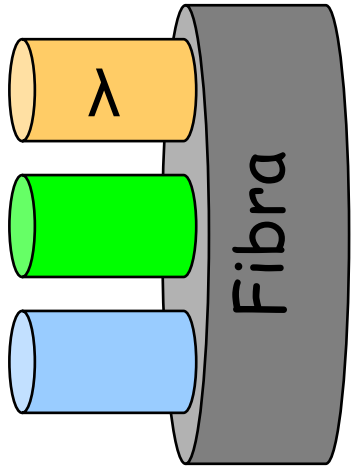


Switching types

- PSC
  - *Packet Switch Capable*
  - MPLS routers
  - Identifican paquetes y los conmutan independientemente
- LSC
  - *Lambda Switch Capable*
  - Un optical cross connect
  - Extrae wavelengths independientes y las conmuta
  - No es capaz de “mirar” dentro de las mismas, trabaja solo en nivel fotónico
- TDMC
  - *Time Division Multiplex Capable*
  - Es capaz de reconocer y conmutar slots temporales

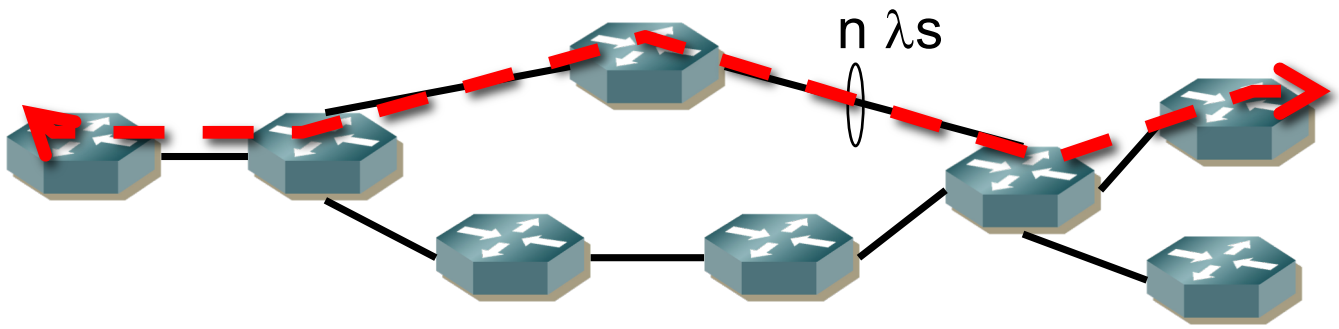
Bandwidth

- En MPLS se puede trabajar con alta granularidad (bytes por segundo)
- En GMPLS con redes de transporte la conmutación está relacionada con recursos físicos
- Si el equipo conmuta wavelengths y soporta de 2.5, 10 y 40 Gbps, ¡ esa es toda la granularidad que soporta !



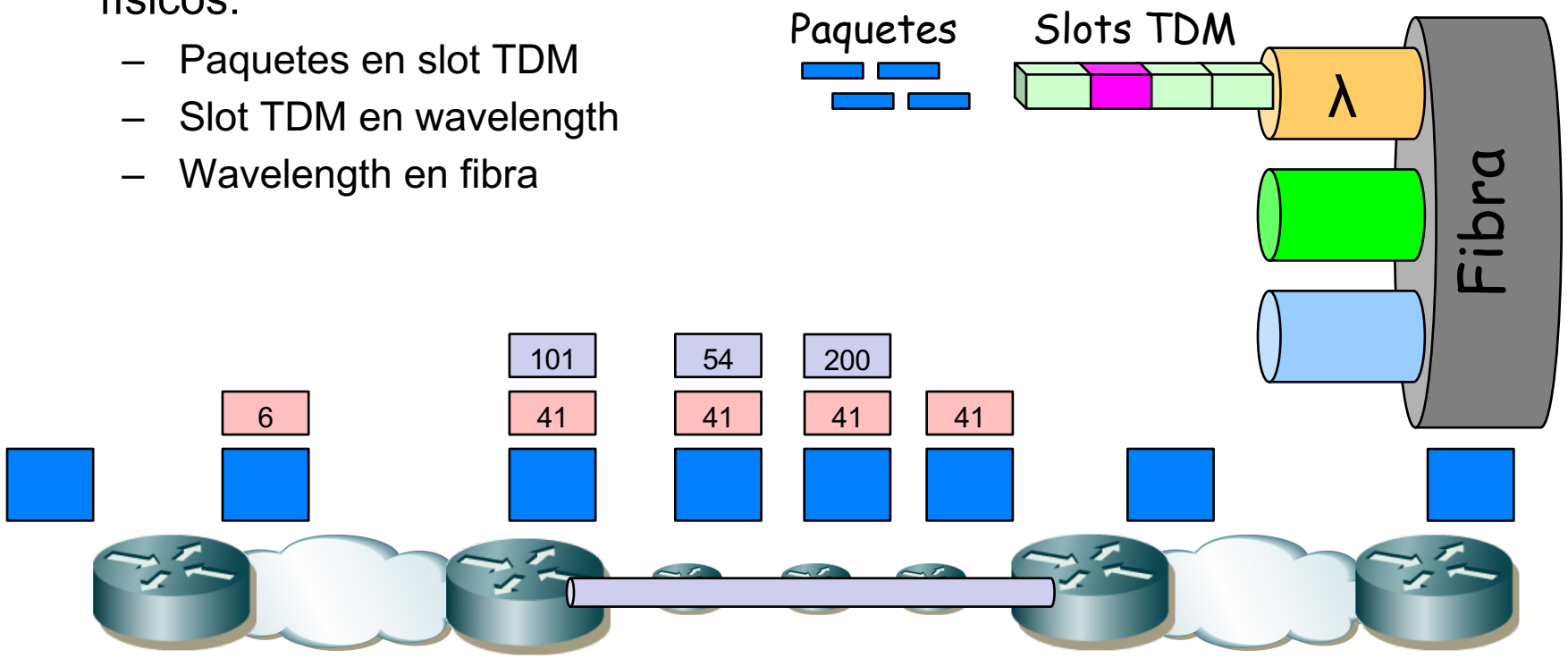
Bidireccionalidad

- LSPs MPLS son unidireccionales
- Se puede hacer bidireccional estableciendo dos LSPs, pero son independientes en el establecimiento
- Interesan LSPs bidireccionales (mismo camino) para que ambos sentido "compartan destino" (*fate sharing*) ante fallos
- Los servicios ofrecidos por redes de transporte suelen ser bidireccionales
- GMPLS añade soporte para establecimiento de LSPs bidireccionales



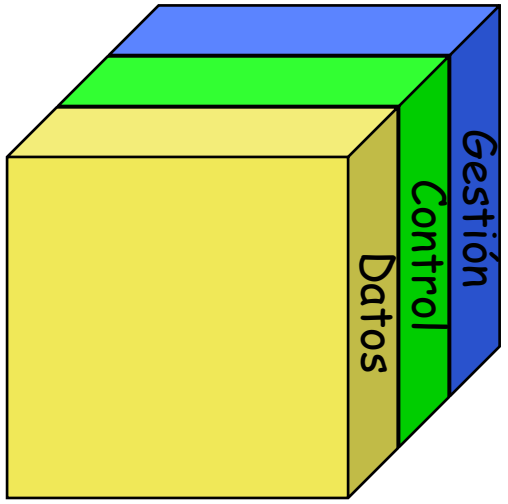
Label Stacking

- MPLS permite hacer túneles de profundidad teóricamente ilimitada
- No es posible con redes de transporte donde la etiqueta está asociada a un recurso físico
- Por ejemplo, un LSP basado en una wavelength (wavelength es la etiqueta) si se transporta en otro LSP de wavelength no puede transportar ambas "etiquetas"
- Existe la posibilidad de hacer una jerarquía basada en los recursos físicos:
  - Paquetes en slot TDM
  - Slot TDM en wavelength
  - Wavelength en fibra



Planos

- En conmutación de paquetes el plano de control (señalización) y de datos pueden compartir enlaces (señalización en banda)
- En redes de transporte los nodos no pueden extraer la señalización del flujo de datos
- Conmutadores pueden no reconocer paquetes
- Por ejemplo, un optical cross-connect no puede hacer conversión OE para extraer de una wavelength mensajes de control
- Soluciones:
  - Canal de datos uso exclusivo para control (wavelength, slot, etc)
  - Emplear enlaces/redes independientes
  - Se puede usar *overhead bytes* (en TDM)
- Fallo de plano de datos ya no se detecta por dejar de recibir mensajes de control
- Mensajes de control necesitan hacer referencia a canales de datos (ya no está claro simplemente por ser compartidos)



Control y señalización

- Entre *signaling controllers*
- Pueden estar separados de los conmutadores de datos
- Protocolo de control o gestión comunicará ambos
- MPLS inicialmente no fijaba el protocolo de señalización y aparecieron
  - CR-LDP
  - RSVP-TE
- RFC 3468 toma finalmente una decisión a favor de RSVP-TE