

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

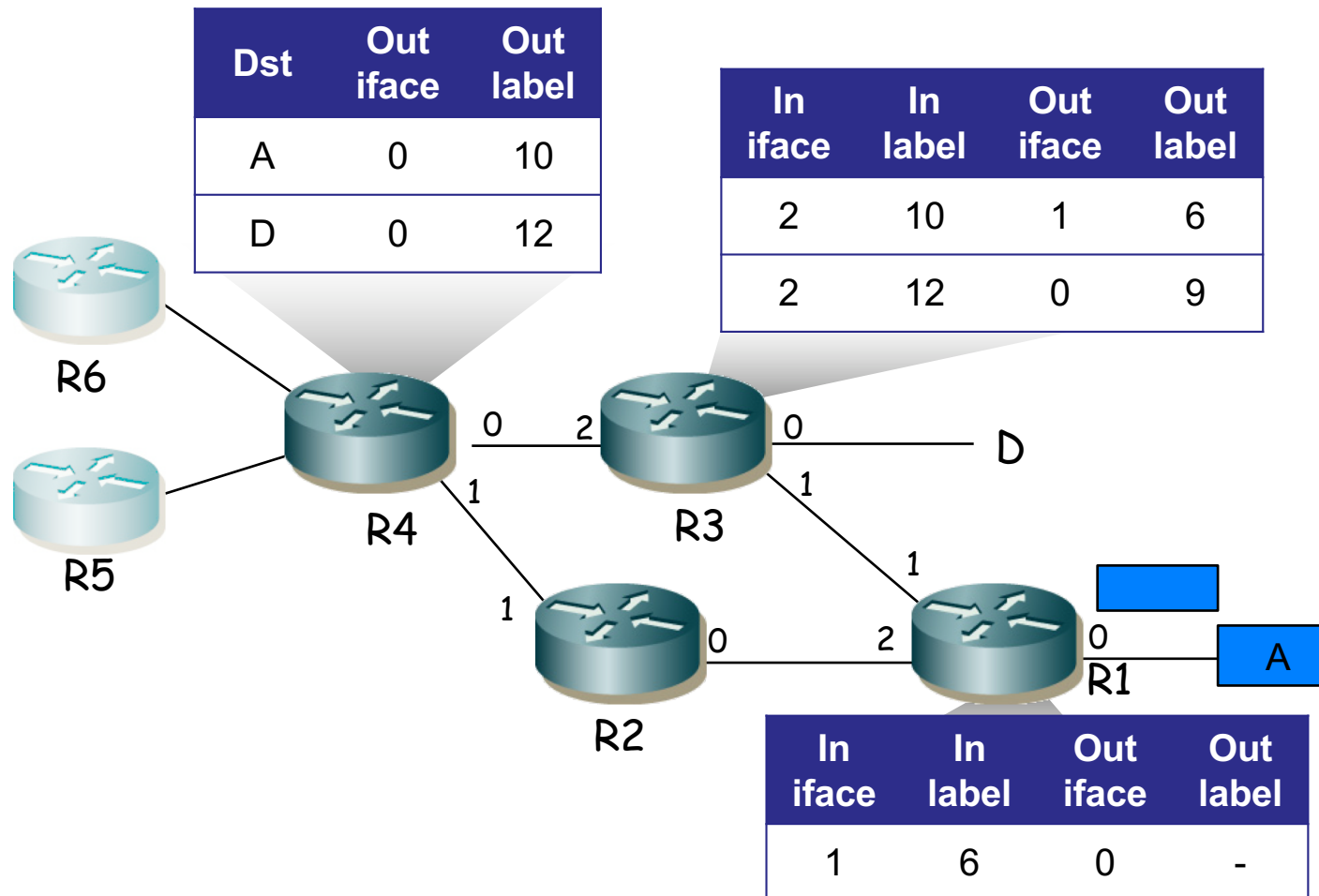
Redes de Nueva Generación
Área de Ingeniería Telemática

MPLS

MPLS

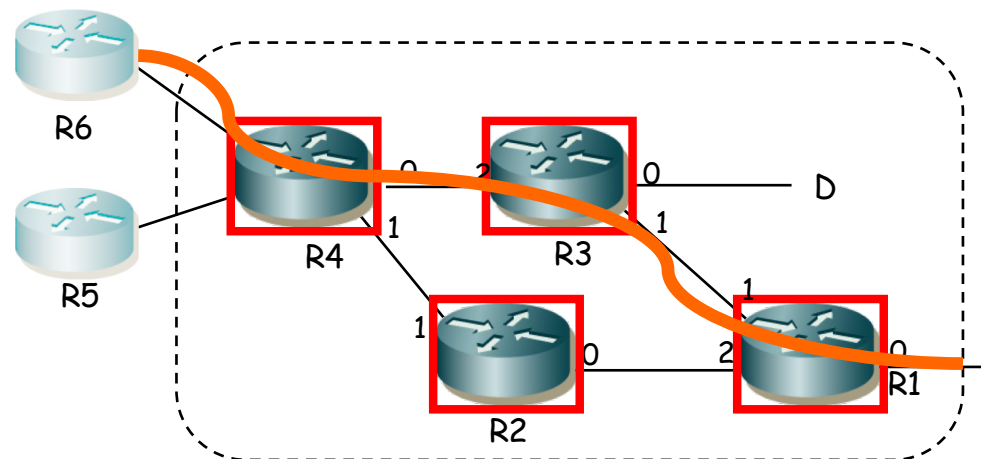
- Inicialmente para ahorrarse el cálculo del *Longest-prefix-match* en los equipos de core
- Hoy en día para hacer *Traffic Engineering*
- Conmutación de paquetes, pero circuitos virtuales
- Heredero de ATM pero con paquetes de tamaño variable
- Inicialmente sin QoS

MPLS "forwarding"



Terminología

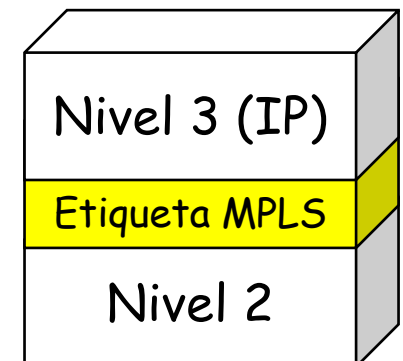
- “MPLS domain”: conjunto contiguo de nodos MPLS bajo una misma administración
- “MPLS ingress node”: nodo frontera de un dominio en su tarea como entrada de tráfico al mismo
- “MPLS egress node”: nodo frontera de un dominio en su tarea como salida de tráfico del mismo
- “Label”: etiqueta numérica, corta,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

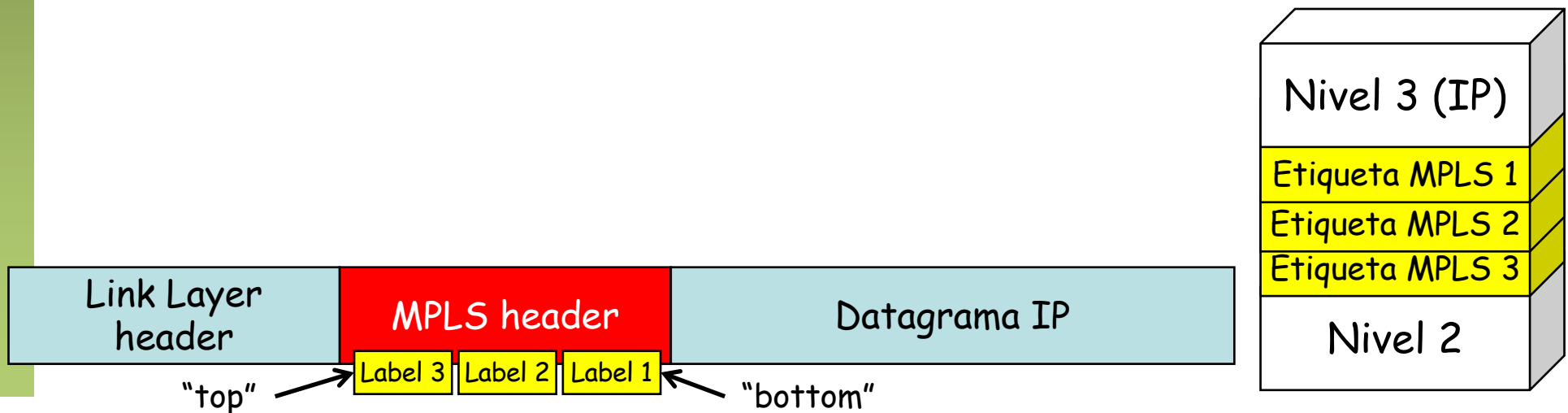
Label Stack

- 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*) (...)



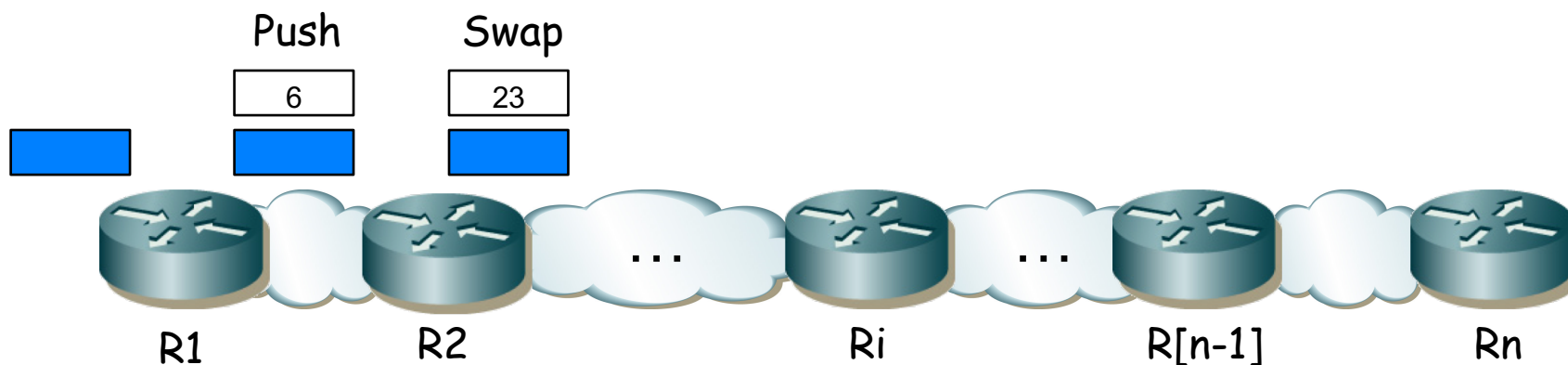
Label Stack

- 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 procesamiento 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



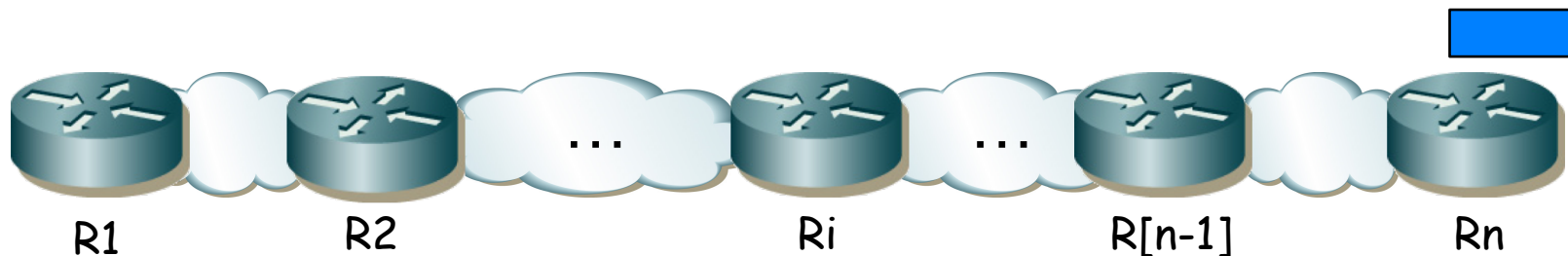
LSP de nivel m

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



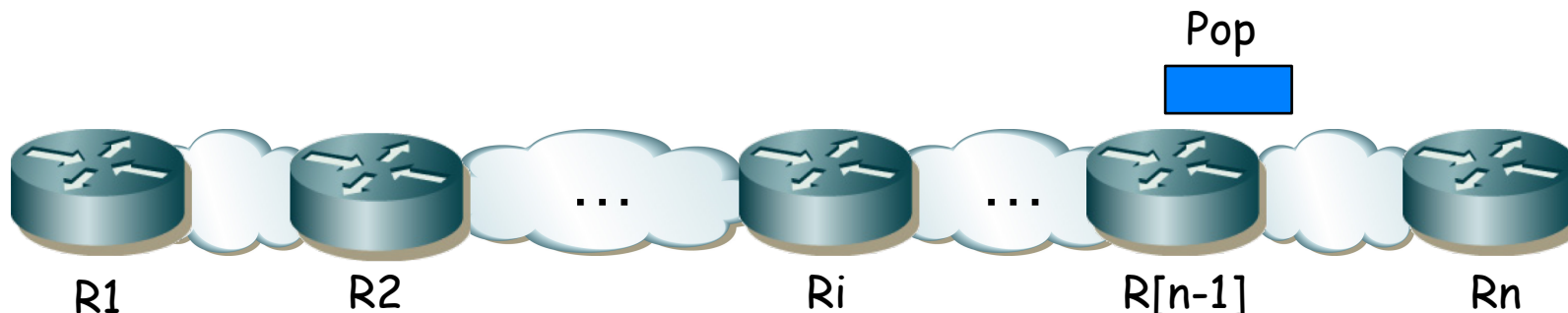
LSP de nivel m

- Secuencia de routers, paquete P con pila de profundidad m-1
- R1: LSP ingress, añade (*push*) una etiqueta a la pila del paquete
- $1 < i < n$ R_i 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
- R_i transmite P a $R[i+1]$ empleando MPLS, es decir, usando la etiqueta superior de la pila
- Equipos entre R_i 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 R_n , 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 R_n el paquete llevara una pila de profundidad $m-1$
- Sin PHP, R_n 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 R_n
 - R_n tendrá como superior la etiqueta de nivel $m-1$ o si $m=1$ la cabecera original para tomar la decisión de reenvío
 - R_n no necesita ser un LSR

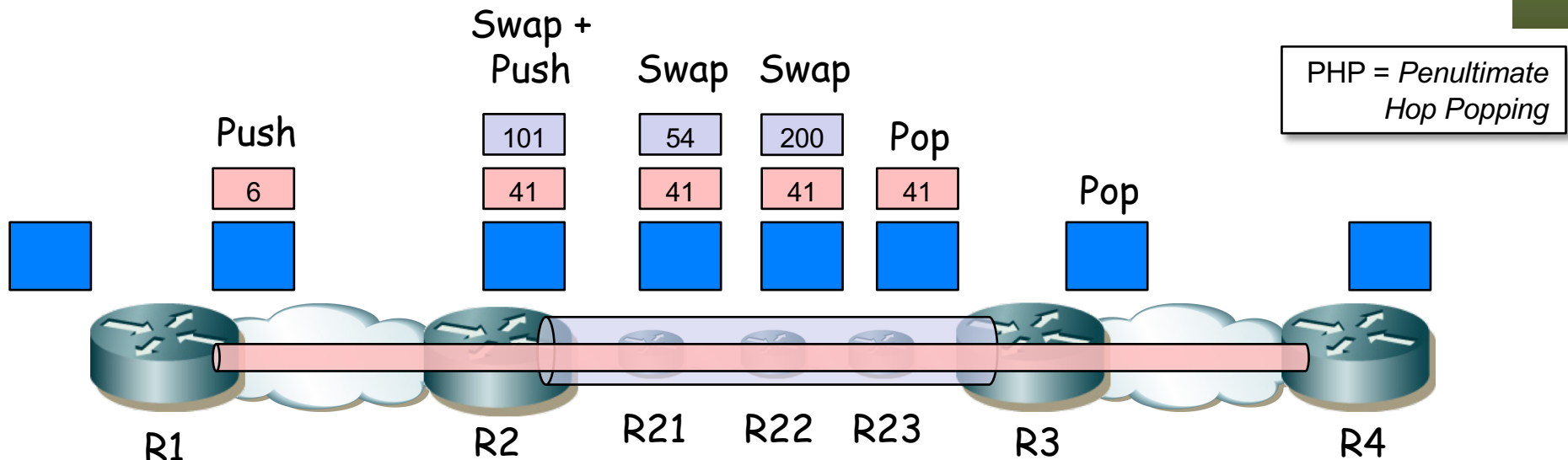


Label distribution

- 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 Protocol – 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

LSP Tunnels dentro de LSPs

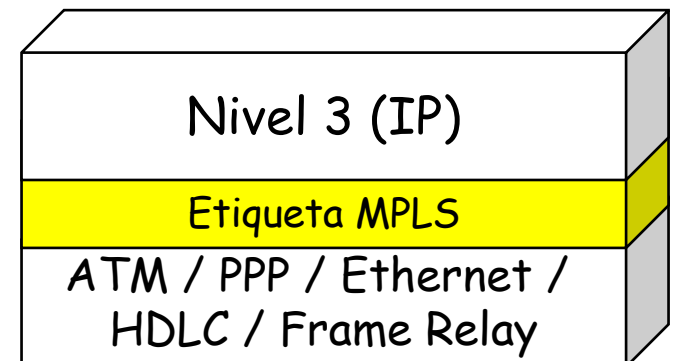
- Por ejemplo LSP <R1, R2, R3, R4>
- R1 recibe paquetes sin etiquetar y les añade una etiqueta
- R2 y R3 no están directamente conectados
- R2 y R3 son “vecinos” mediante un túnel LSP
- R2 no solo hace swap de etiqueta sino también push de una nueva para el túnel
- R21 conmuta en función de la etiqueta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



MPLS: Transporte

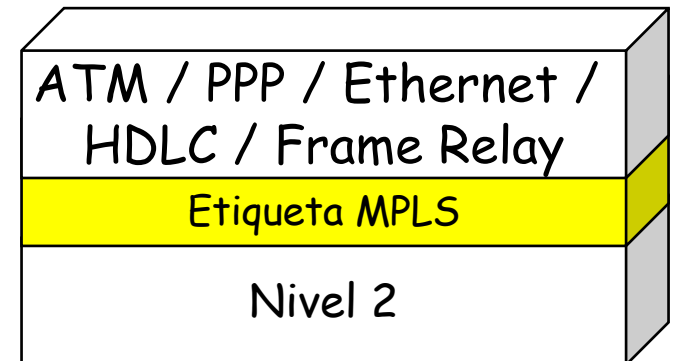
Transporte de MPLS

- 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 queremos...



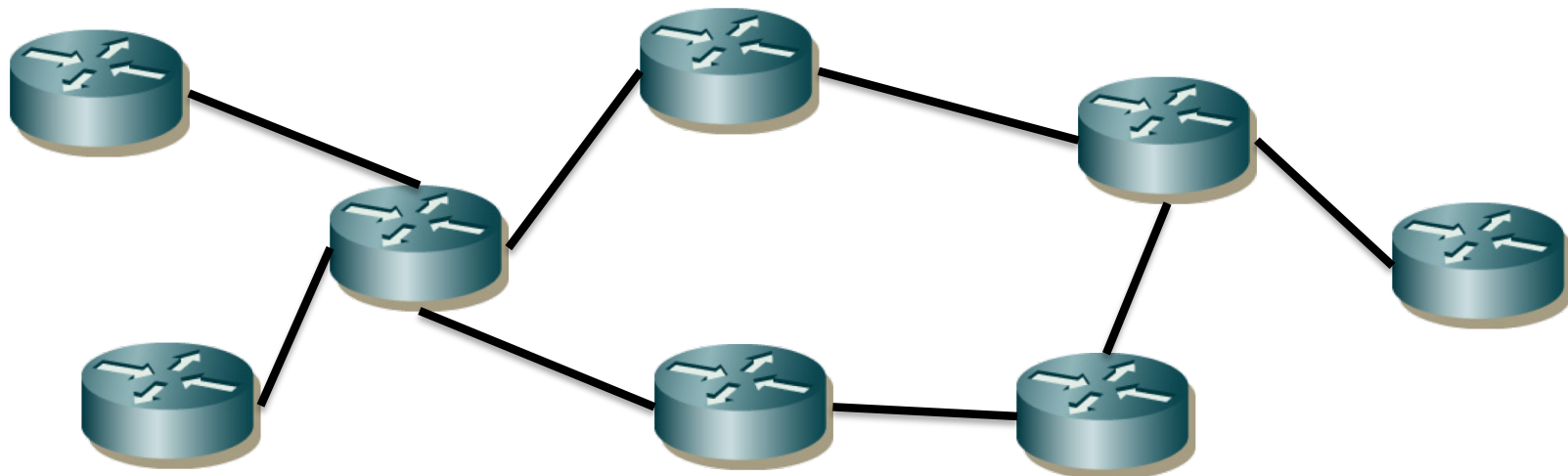
Traffic Engineering

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

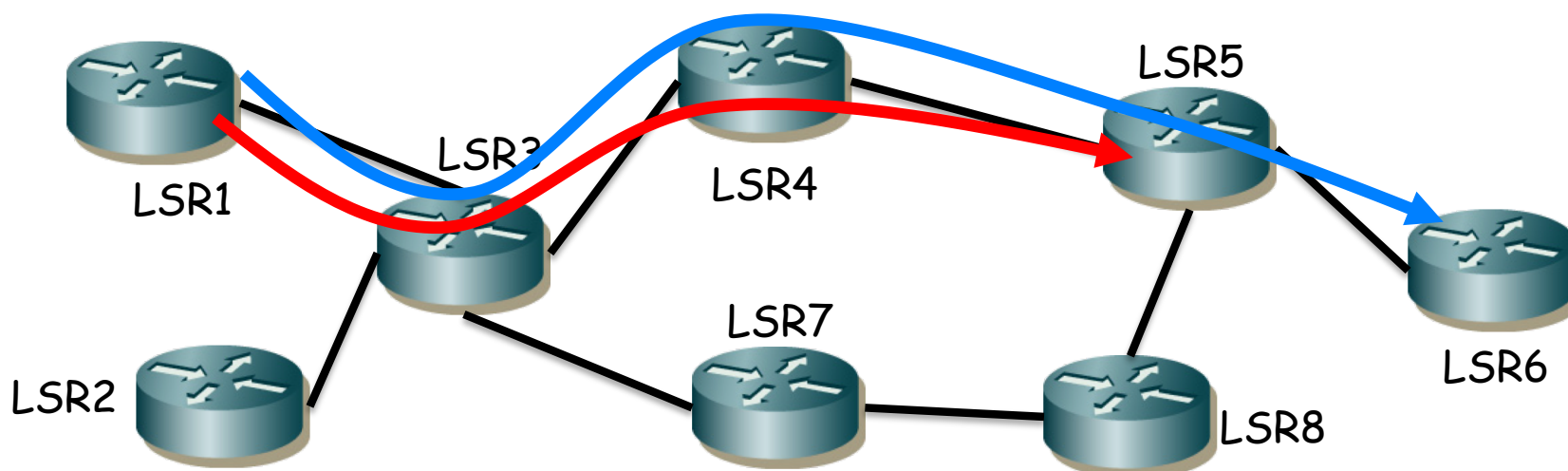
Traffic Engineering

- 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”* (...)



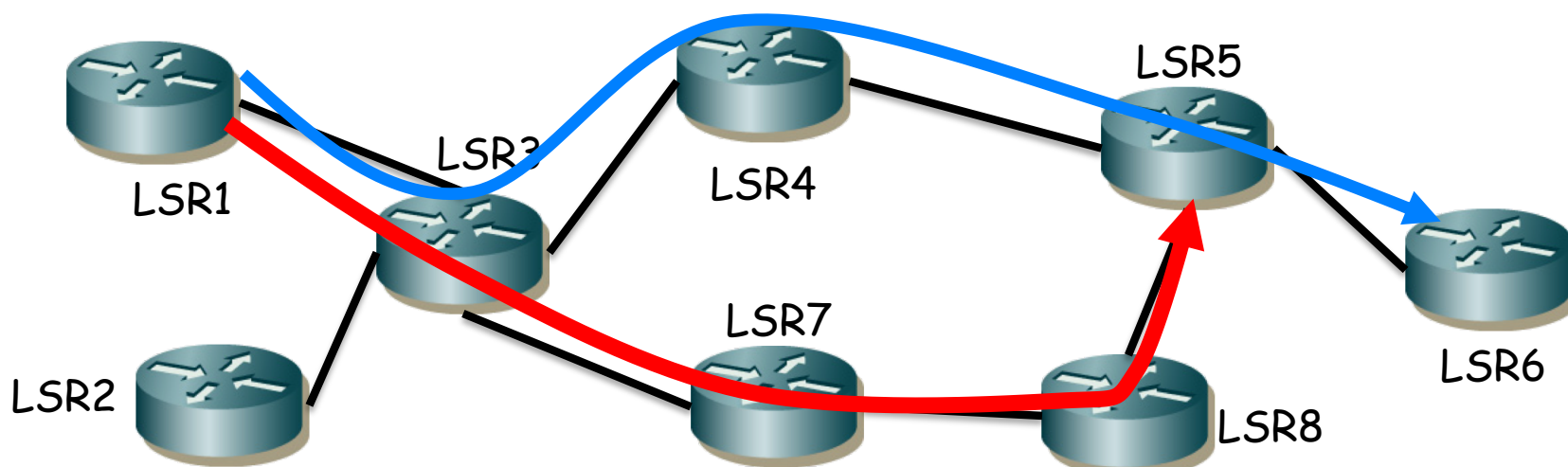
Ejemplo

- 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) (...)



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 (...)

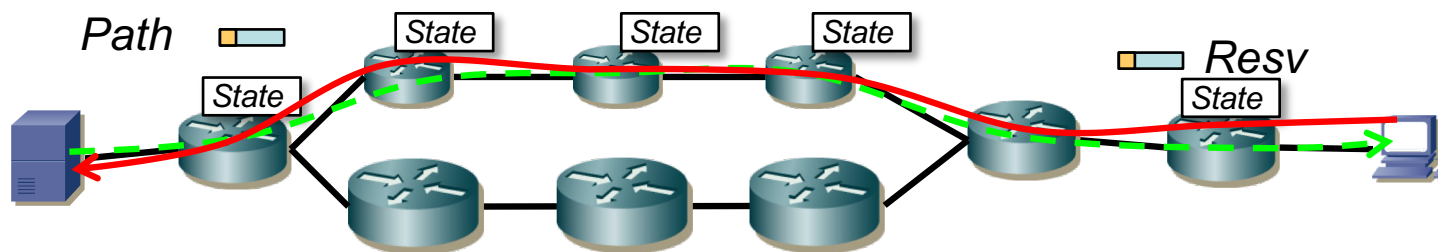


Explicit routing

- 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 (Committed Data Rate), CBS (Committed Burst Size), EBS (Excess Burst Size)
 - Parámetros para token buckets
 - RSVP-TE: RFC 3209 “Resource Reservation Protocol – Traffic Engineering”
 - (...)

RSVP-TE

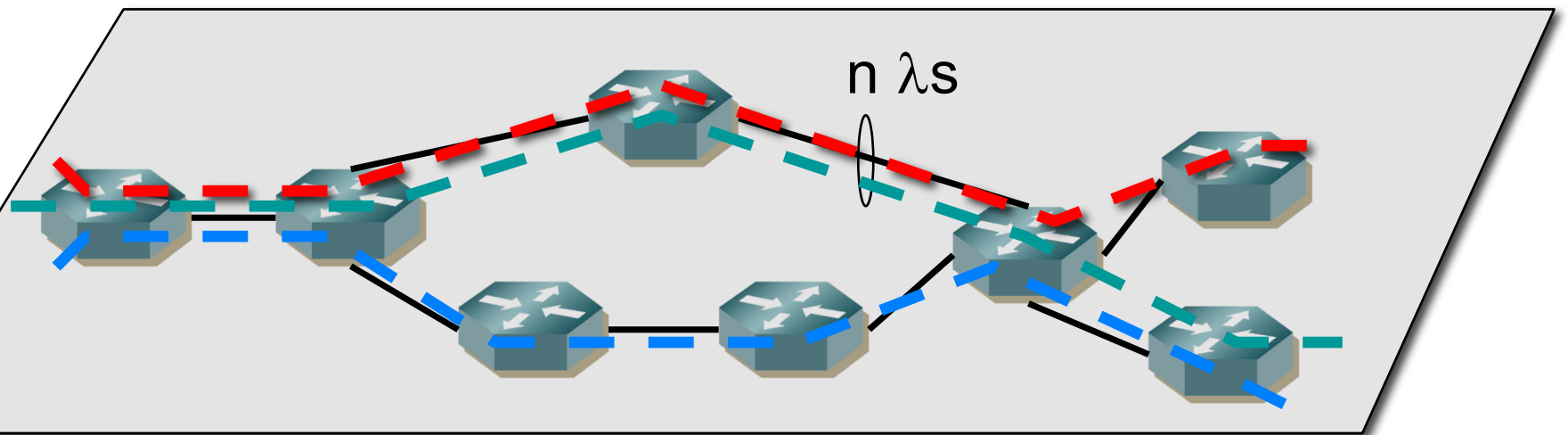
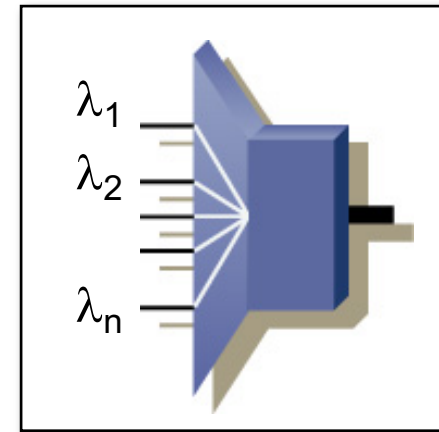
- RSVP
 - El mensaje *Path* lo envía la fuente y sigue la ruta calculada por los protocolos de encaminamiento
 - El mensaje *Resv* emplea ese estado para seguir el camino inverso
 - Si no hay recursos suficientes falla la reserva
 - Falla incluso si existe otro camino que sí disponga de recursos
- RSVP-TE 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



GMPLS

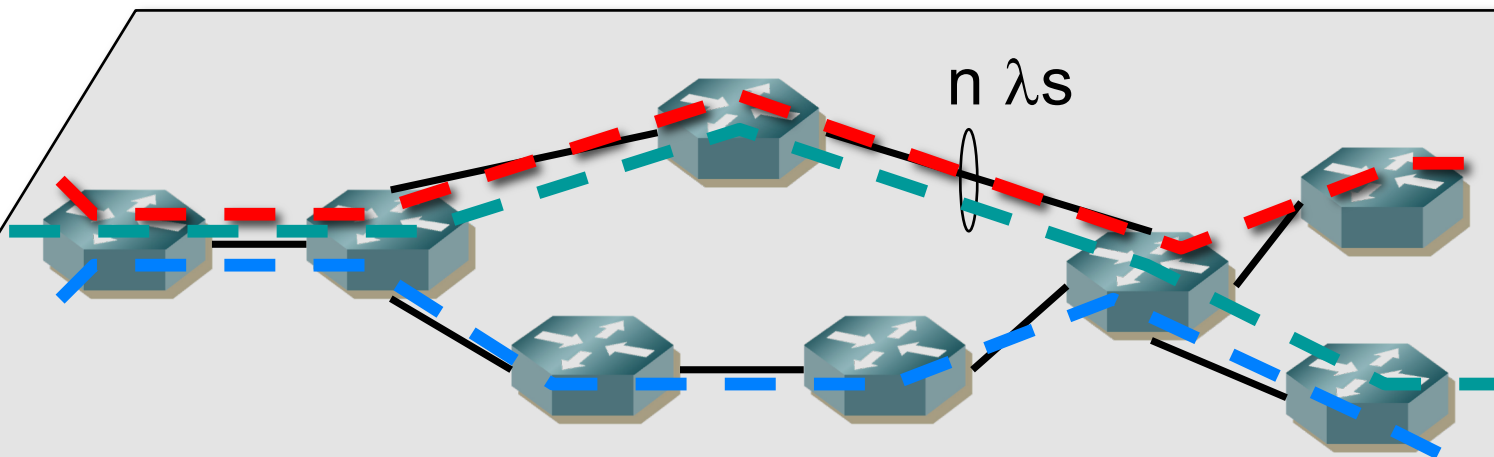
Lightpaths

- 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 MP λ S
- 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*

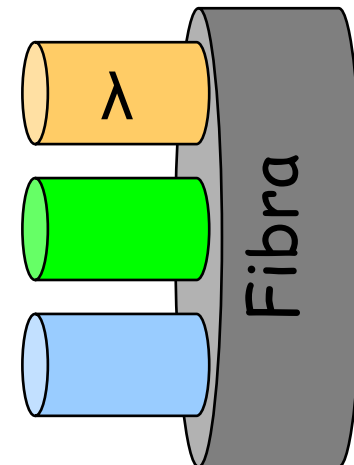


Switching types

- PSC
 - *Packet Switch Capable*
 - MPLS routers
 - Identifican paquetes y los conmutan independientemente
- LSC
 - *Lambda Switch Capable*
 - Un optical cross connect
 - Extrae wavelenghts 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 wavelenghts y soporta de 2.5, 10 y 40 Gbps, ¡ esa es toda la granularidad que soporta !



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

