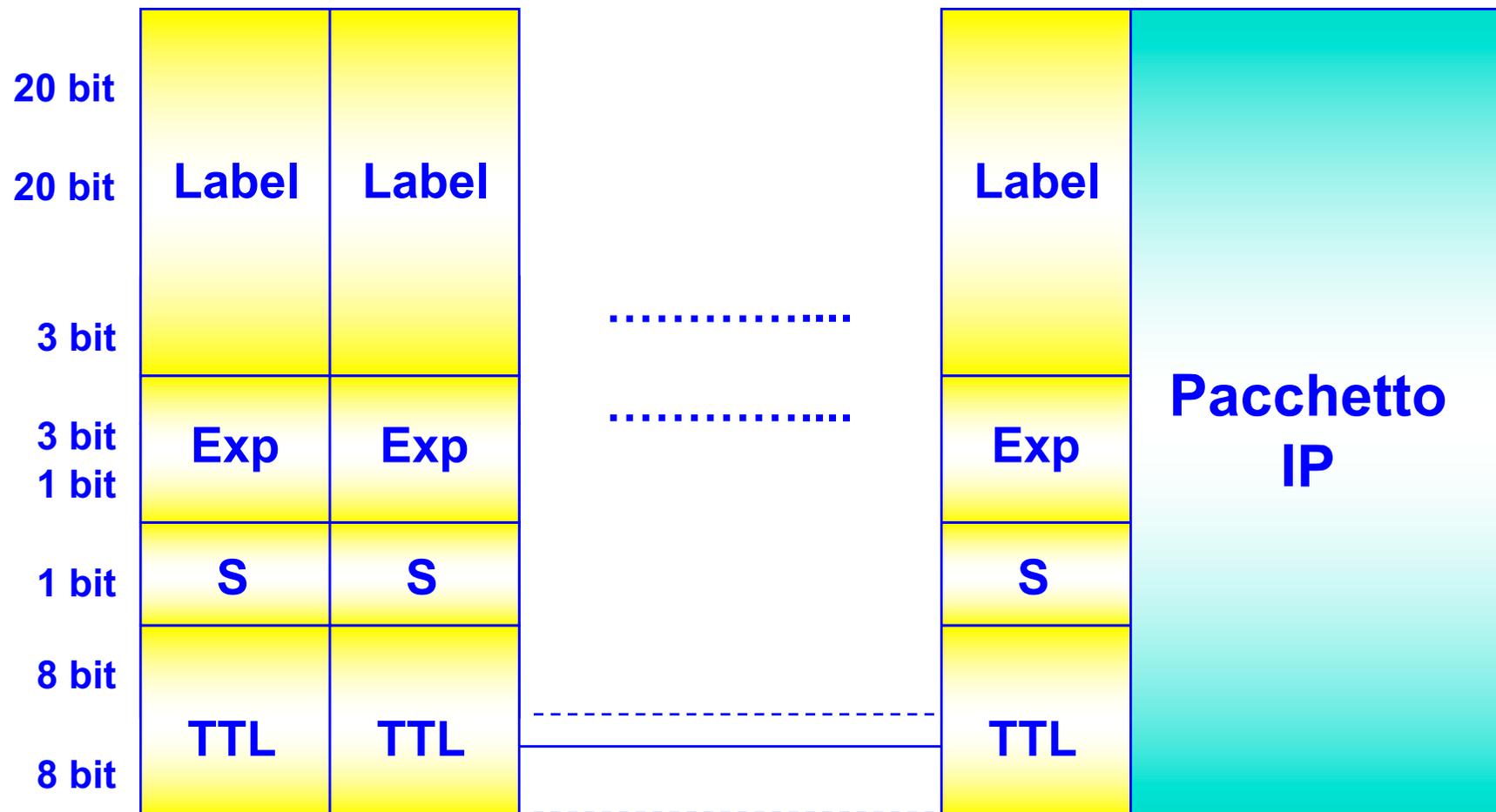


---

# Multi Protocol Label Switching (QoS & Traffic Engineering)

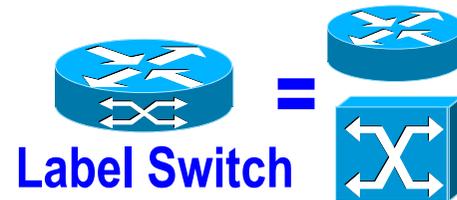
# MPLS: architettura

- + L'idea alla base di tale architettura è quella di associare a tutti i pacchetti un breve identificativo di lunghezza fissa, detto Label, con cui gli apparati di internetworking possono effettuare un instradamento veloce basato sulla commutazione di etichetta (label swapping)
- + MPLS è indipendente sia dalla sottorete di trasporto (Frame Relay, ATM, etc.) sia dai protocolli di rete adottati



# Nodo MPLS

## Il nodo di rete MPLS



Control component (router + LDP)

+

Forwarding component (L2 switch)

### + Control Component

- Insieme di moduli demandati all'allocazione delle Label ed al binding delle Label tra nodi adiacenti
- 'Intelligenza' di livello 3 (IP addressing, IP routing)

### + Forwarding Component

- Inoltro secondo il paradigma label swapping

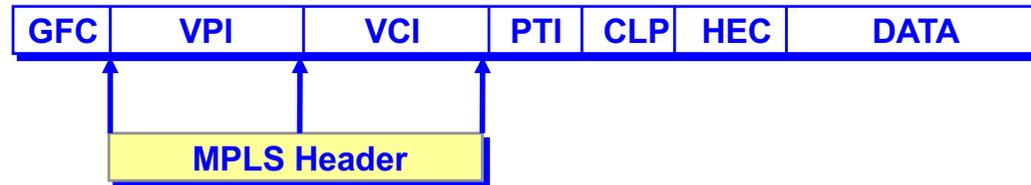
+ Le due componenti devono essere indipendenti: posso sviluppare differenti protocolli su qualsiasi mezzo

+ La Control Component viene talvolta realizzata come parte integrata (SW o HW) del nodo di rete, talvolta come controllore esterno

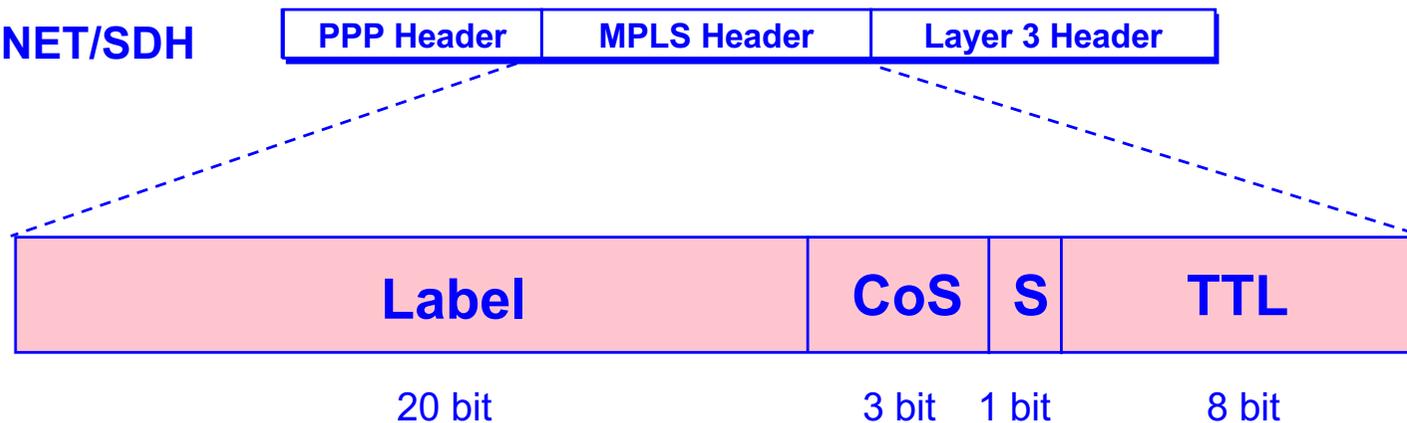
# Label encoding

- + Se il livello data-link supporta nativamente un campo per la label (ATM con VPI/VCI, Frame Relay con DLCI) in questo campo viene inserita la label MPLS
- + Se il livello data-link non supporta un tale campo, la label MPLS viene incapsulata in un MPLS header che viene inserito tra l'header di livello 2 e quello di livello 3

## Header cella ATM



## Packet Over SONET/SDH



# Terminologia

---

- + **Label Edge Router (LER)**: router di frontiera per una rete MPLS; svolgono funzionalità di instradamento da e verso l'esterno, applicando e rimuovendo le label ai pacchetti in ingresso ed uscita dalla rete
- + **Label Switching Router (LSR)**: switch che operano la commutazione di etichetta (label) all'interno della rete e prevedono il supporto di funzionalità di instradamento
- + **Label Distribution Protocol (LDP)**: in congiunzione con i protocolli di routing tradizionali, LDP è utilizzato per distribuire le label tra i dispositivi della rete
- + **Forwarding Equivalence Class (FEC)**: un insieme di pacchetti IP che vengono instradati allo stesso modo (ad es., lungo lo stesso percorso, con lo stesso trattamento)
- + **Label Switched Path (LSP)**: Il percorso attraverso uno o più LSRs seguito da un pacchetto appartenente ad un certa FEC

# Funzionamento

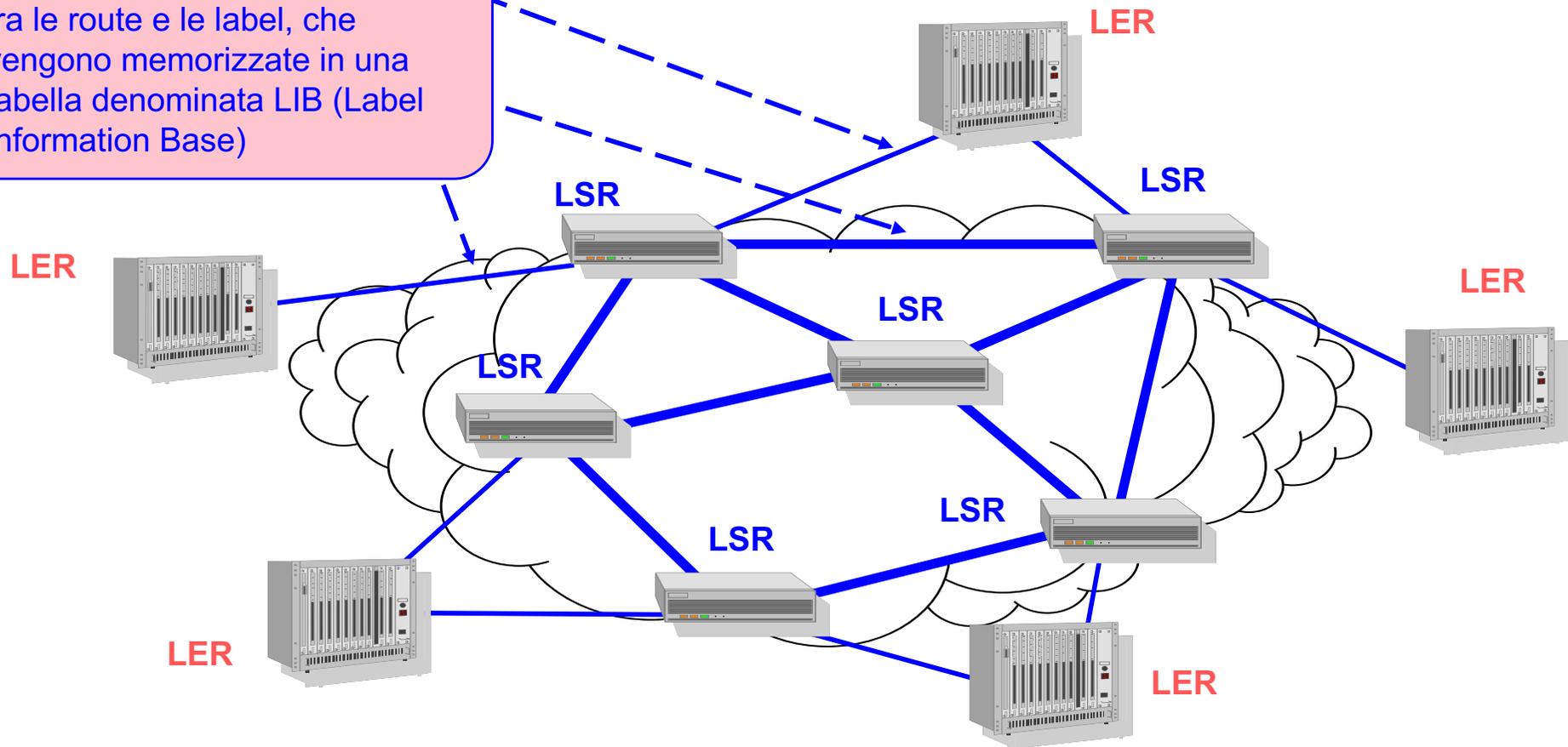
---

- + **Il LER di ingresso al backbone MPLS analizza l'intestazione IP del pacchetto, classifica il pacchetto, aggiunge la label e lo trasmette al next hop LSR**
- + **Nella nuvola di LSRs il pacchetto viene instradato lungo il LSP in accordo alla label**
- + **Il LER di uscita rimuove la label ed il pacchetto viene instradato in base all'indirizzo IP di destinazione**

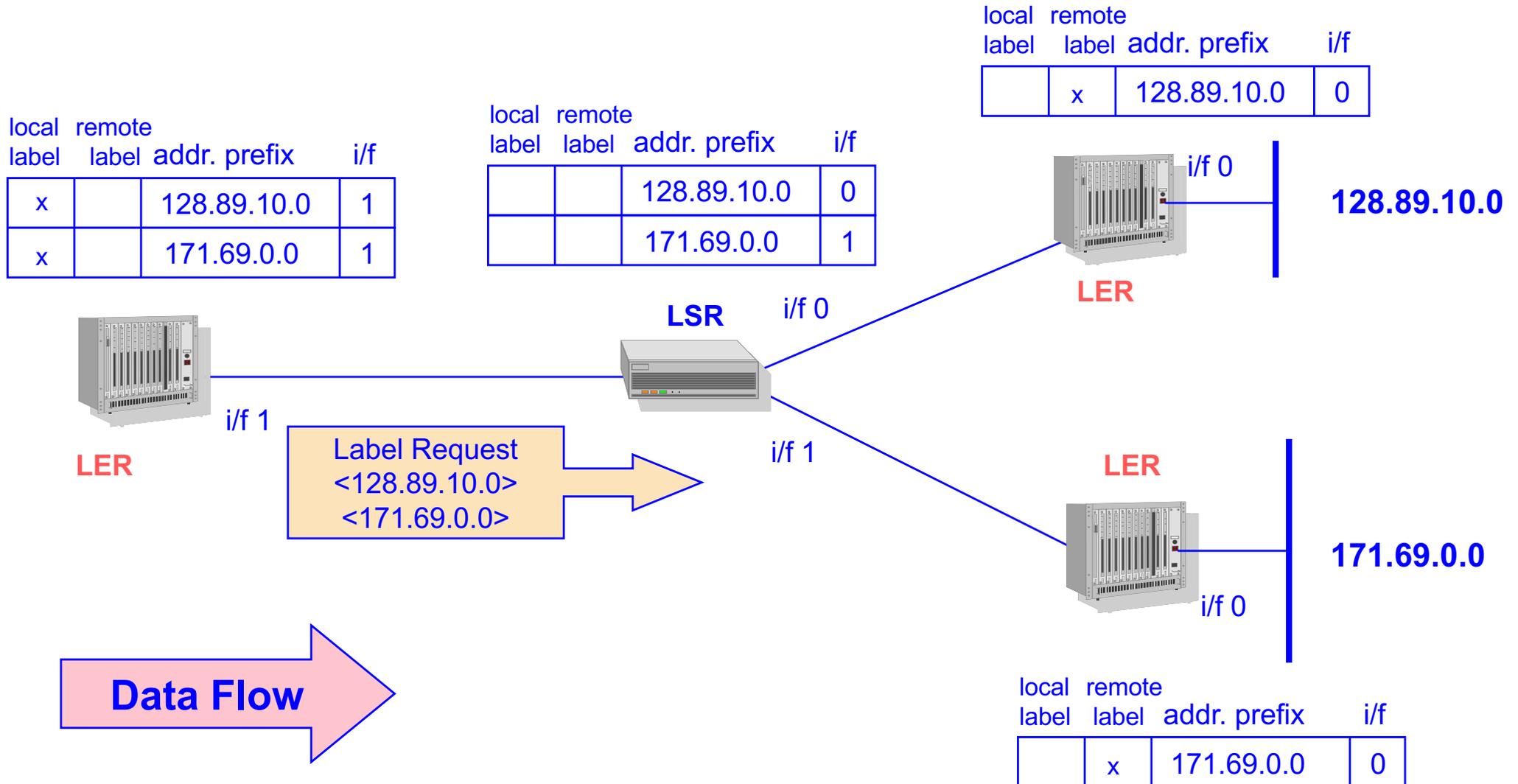
# Label Switching Operation: Control

- + LDP è utilizzato per distribuire le associazioni <label, prefisso>

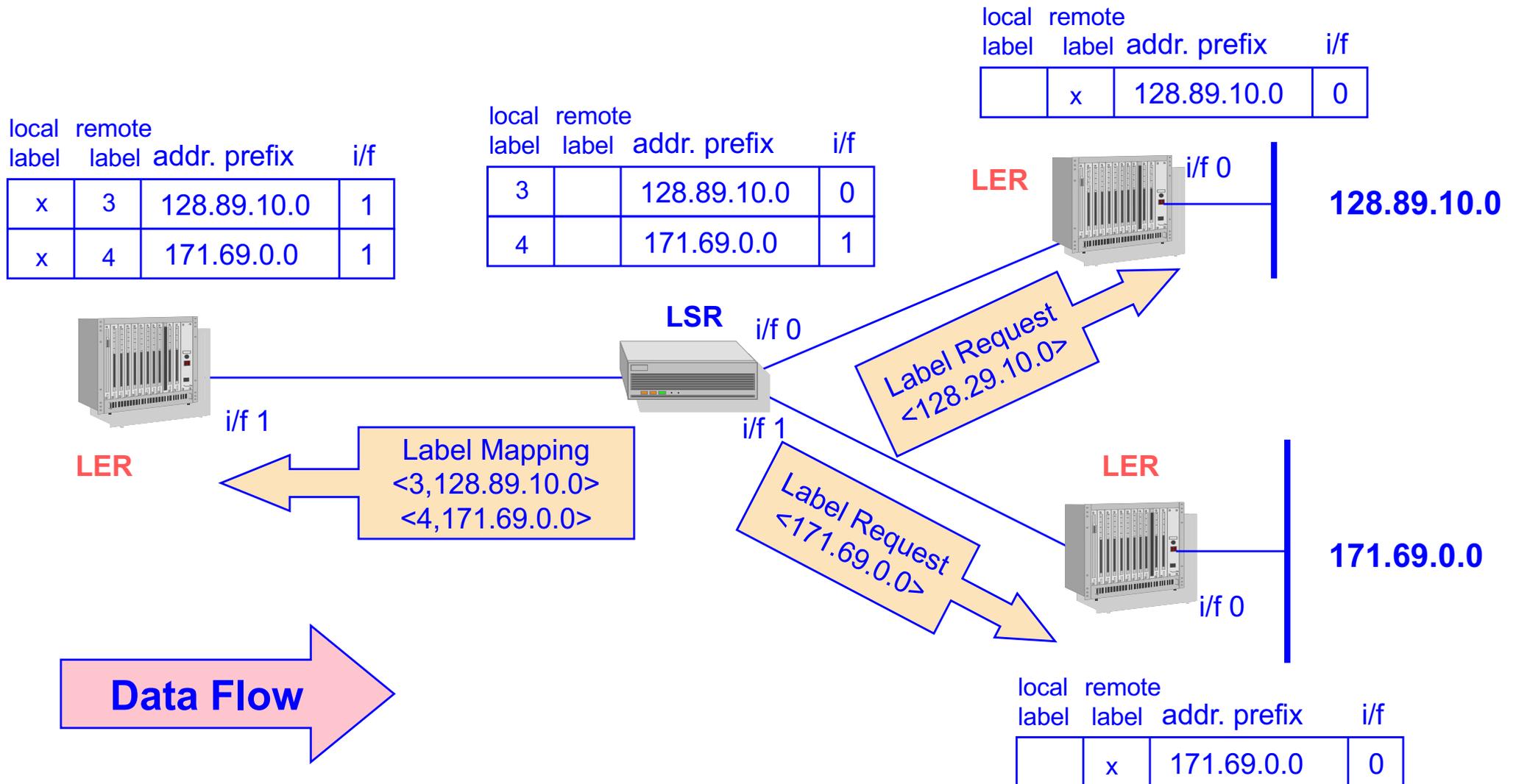
• LDP stabilisce le associazioni tra le route e le label, che vengono memorizzate in una tabella denominata LIB (Label Information Base)



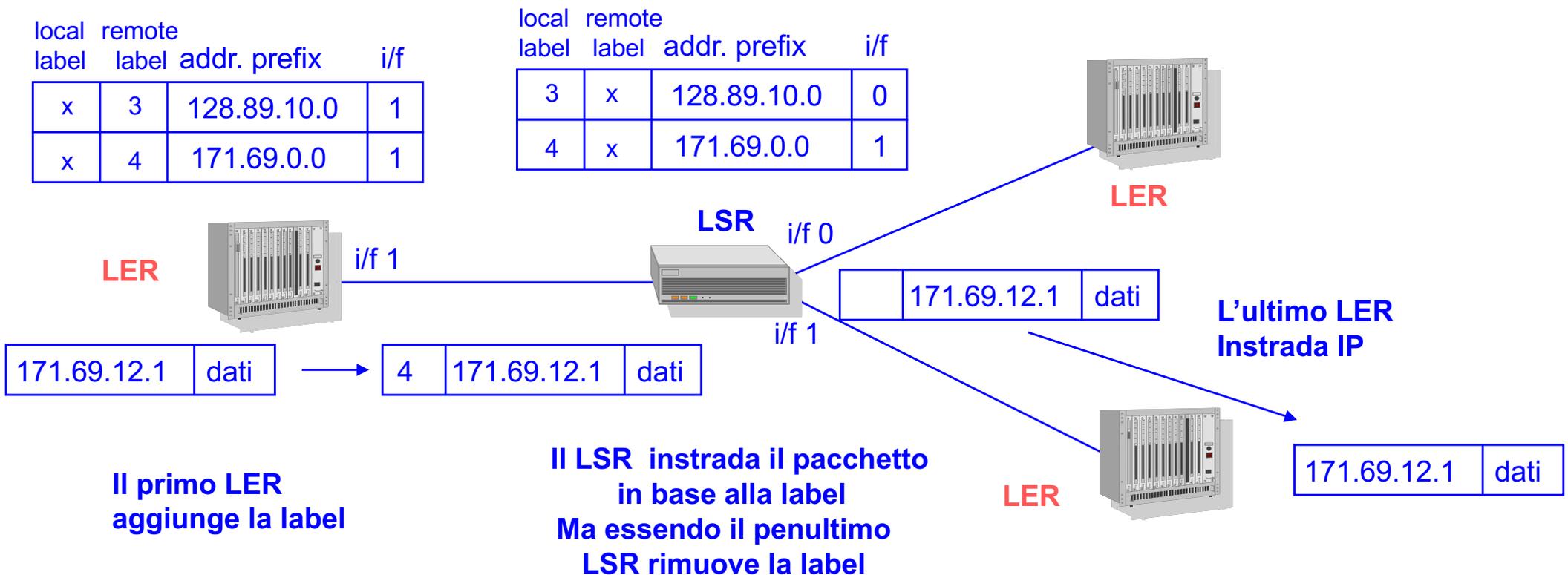
# LDP: Downstream on Demand



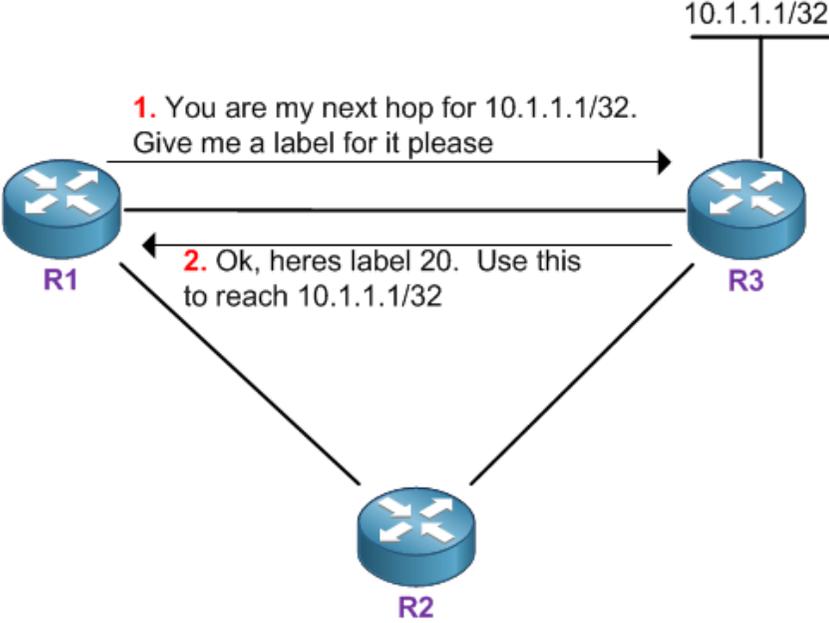
# LDP: Downstream on Demand



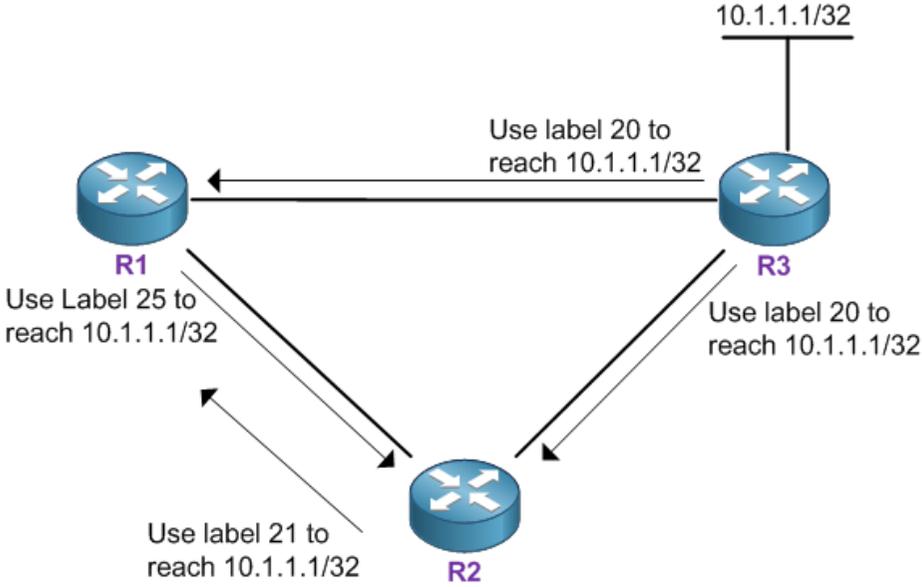
# Label Switching Operation: Forwarding



# LDP: Downstream Unsolicited vs OnDemand



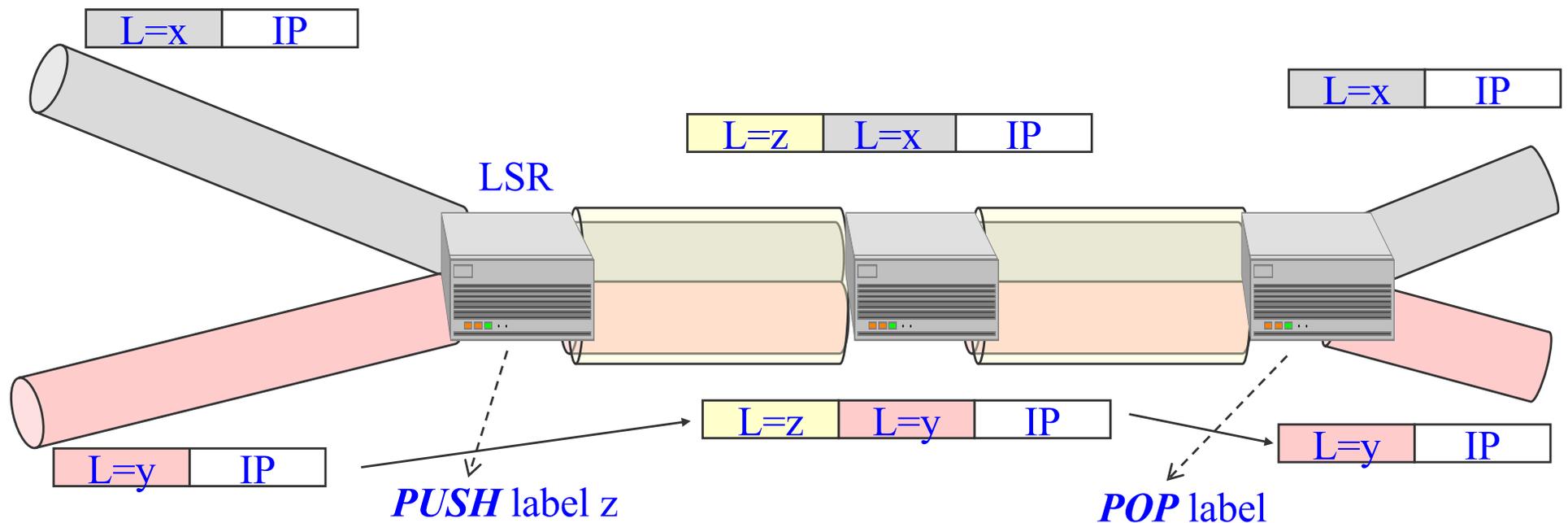
OnDemand



Unsolicited  
(Cisco default)

# Label Stacking

- + Le label MPLS possono essere messe in stack per aggregare, su un tratto di rete, due o più LSP in un unico LSP di ordine gerarchico superiore
- + L'inserimento di una label prende il nome di *label push*
- + La rimozione di una label prende il nome di *label pop*
- + L'instauramento è fatto sempre in base alla label di ordine superiore; se la label non c'è si instrada a livello IP



# MPLS: realtà

---

## + **Perchè un ISP dovrebbe impiegare MPLS?**

- **Il vantaggio principale di MPLS è che consente ad un ISP di offrire nuovi servizi che non possono essere supportati semplicemente attraverso le tecniche di routing convenzionale**

## + **Attualmente possono essere individuate tre applicazioni di MPLS nel core degli ISP:**

- **Traffic Engineering (MPLS-TE)**
- **Traffic Engineering with QoS (MPLS DS-TE)**
- **Virtual Private Networks (VPN)**

# MPLS-TE

- + **Traffic Engineering consente ad un ISP di instradare un certo flusso di traffico lungo un percorso differente da quello calcolato dal protocollo di routing in modo da utilizzare (qualora sia necessario) un percorso fisico meno congestionato**
- + **Ciò consente agli ISP per bilanciare il carico sui vari link e nodi della rete in modo che nessuno di questi componenti sia sovrautilizzato o sottoutilizzato**
- + **MPLS-TE estende il funzionamento di base di MPLS introducendo:**
  - **dei meccanismi per il recupero delle informazioni di occupazione dei link**
  - **Dei meccanismi di segnalazione RSVP-TE, CR-LSP per il setup di LSP con un routing forzato**

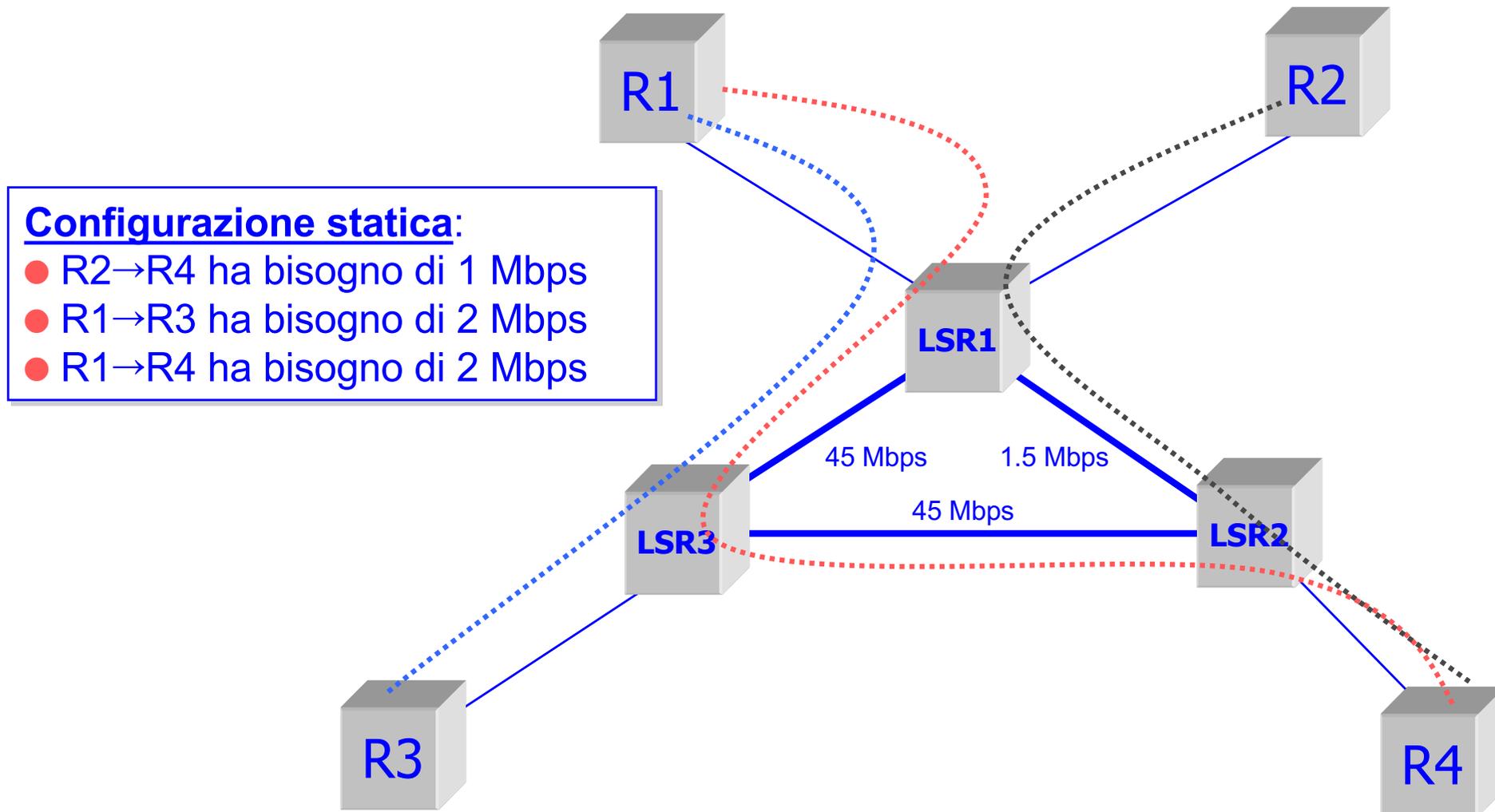
# Traffic Engineering: come?

- + **Normalmente un LSP viene instaurato in base al calcolo del percorso a costo minore effettuato dal protocollo di routing utilizzato sul backbone**
- + **Questa modalità non offre nessun valore aggiunto in termini di traffic engineering**
- + **Possono essere utilizzati diversi meccanismi per instaurare un LSP differente da quello determinato attraverso il protocollo di routing:**
  - **configurazione statica di tutti i LSR nel LSP (in modo analogo a come viene configurato un backbone IP/ATM tradizionale)**
  - **configurazione del LER con il percorso completo. Il LER poi utilizza una versione modificata del protocollo RSVP per installare le LIB in ciascuno dei LSR lungo il percorso (LSP)**

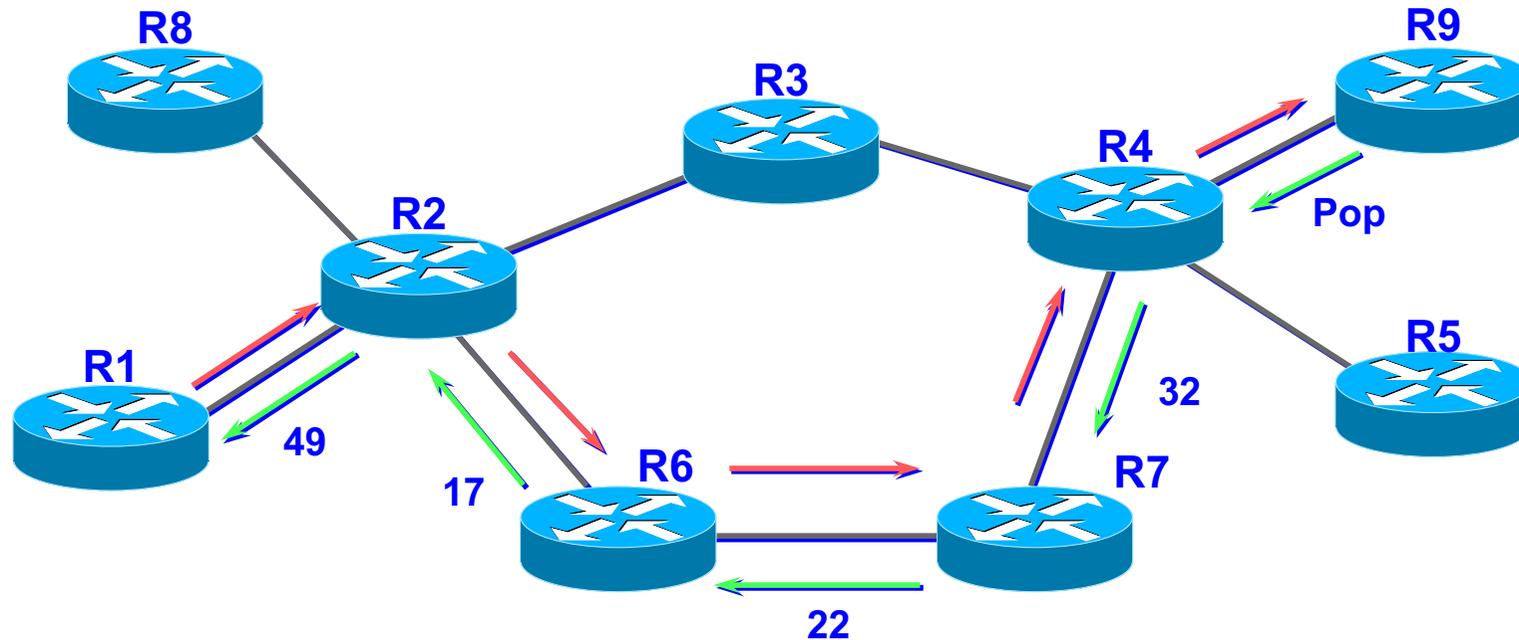
# Routing di un LSP

- + Il percorso che un LSP deve seguire affinché attraversi link con {opportuna capacità riservabile, scarichi} è solitamente pre-calcolato da un tool off-line
- + E' necessaria la conoscenza dello stato di occupazione delle varie interfacce di uscita dei LER
  - A) Soluzioni proprietarie che sfruttano query dello LSR MIB
  - B) Estensione dei protocolli di routing link-state (flooding delle informazioni delle interfacce), IGP di tipo OSPF o IS-IS, in modo che trasportino anche lo stato di occupazione delle risorse. Conseguentemente i LER (o una entità centralizzata di management) conoscono topologia e occupazione della rete
- + Calcolo del percorso attraverso constraint-based, shortest path first (CSPF)
  - Algoritmo shortest-path calcolato sulla topologia di rete con esclusione dei link che non possono supportare la banda B dello LSP di cui si sta facendo il setup
- + Setup manuale o attraverso **RSVP-TE** / CR-LDP

# LSP: configurazione statica



# LSP: configurazione con RSVP-TE



**Setup: Path (R1->R2->R6->R7->R4->R9)**

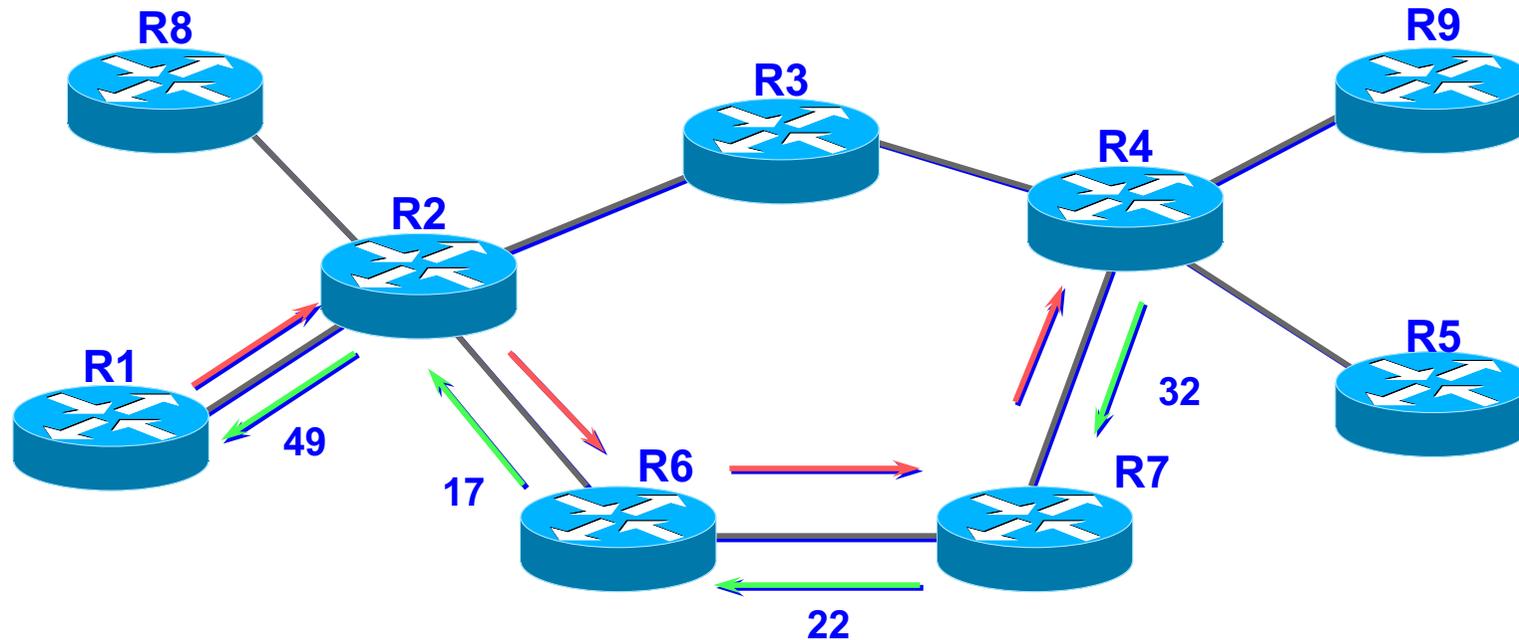


**Reply: RESV (Comunica le label)**

# RSVP-TE

RSVP Object	RSVP Message	Description
LABEL_REQUEST	Path	Label request to downstream neighbor
LABEL	Resv	MPLS label allocated by downstream neighbor
EXPLICIT_ROUTE	Path	Hop list defining the course of the TE LSP
RECORD_ROUTE	Path, Resv	Hop/label list recorded during TE LSP setup
SESSION_ATTRIBUTE	Path	Requested LSP attributes (priority, protection, affinities)

# LSP: configurazione con CR-LDP



➡  
**Setup: Label Request (R1->R2->R6->R7->R4->R9)**

←  
**Reply: Label mapping**

# MPLS & QoS

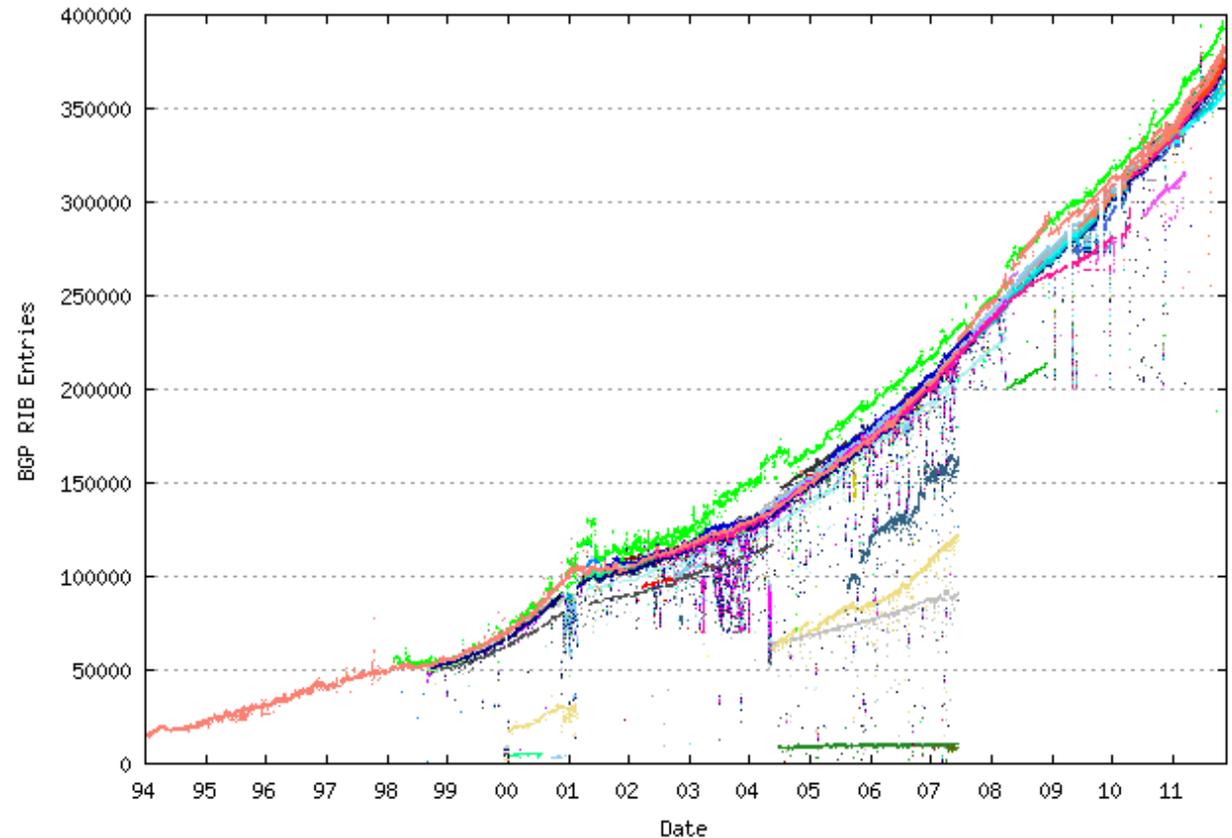
- + L'ingegnerizzazione del traffico implica una pianificazione dell'uso delle risorse al fine di permettere un efficace trasferimento dei dati appartenenti agli LSP
- + Pertanto il traffic engineering tenta di rendere i link scarichi
- + Come ci si comporta in caso di classi di servizio diverse ?
- + **Regola Empirica:** Nel caso in cui la somma delle capacità richieste da tutti gli LSP su una interfaccia di uscita di un LSR sia inferiore o uguale alla metà della capacità della linea, tutti i LSP subiranno un ritardo molto basso e non c'è necessità di impiego di meccanismi di scheduling (e.g., WDRR)
- + Quando però, post traffic engineering, le capacità delle interfacce in gioco cominciano a lavorare con carichi  $\gg 0.5$ , allora è necessario differenziare il trattamento del traffico
- + MPLS si integra con DiffServ

# MPLS e DiffServ

- + **Quale è il criterio di classificazione che un LSR adotta per determinare la coda dello scheduler (ovvero il forwarding-behaviour DiffServ) ?**
  
- + **Due soluzioni:**
  - **Exp inferred LSP (E-LSP)**
    - La classificazione per lo scheduler avviene attraverso il campo **Exp** (3 bit) dell'intestazione MPLS
    - Forwarding behaviour e drop precedence dedotto dalla codifica del campo Exp
    - Pacchetti di LSP diversi con lo stesso campo Exp sono trattati ugualmente
    - Richiede al massimo 8 code dello scheduler, tante quanto le possibili combinazioni di campo EXP
  
  - **Label inferred LSP (L-LSP)**
    - Forwarding behaviour dedotto dalla label, drop precedence dedotto dal campo **Exp**
    - Ogni LSP può essere gestito con un differente forwarding behaviour indipendentemente dal campo EXP
    - Richiede un numero variabile di code dello scheduler
    - Più complesso, ma più versatile
    - L'associazione < Forwarding behaviour , label > deve essere segnalata esplicitamente durante la fase di setup dello LSP

# MPLS & BGP

- + BGP è il protocollo di routing utilizzato tra AS
- + BGP è eseguito dai nodi di bordo di un AS
- + Le tabelle BGP contengono le rotte verso tutte le net
  - 2011 circa 400k rotte (<http://bgp.potaroo.net/>)

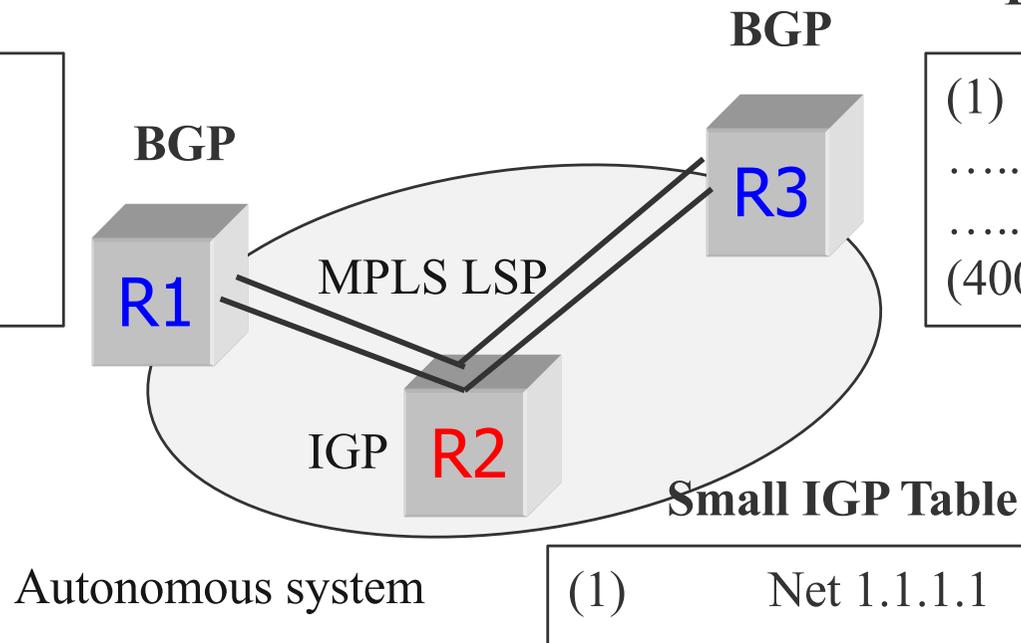


# MPLS & BGP

- + **Problema: come fanno i router interni (e.s. R2) a instradare i pacchetti in transito, i.e. diretti ad una delle 400k reti esterne ?**
  - Replico le tabelle BGP anche nei router interni (router interni costosi)
  - Faccio dei Label Switched Path MPLS tra router di bordo (full mesh) su cui instrado solo il traffico di transito
    - I router interni hanno solo tabelle di routing che riguardano gli instradamenti interni

## Large BGP Table

(1)	Net a.a.a.a
.....	
.....	
(400 k)	Net z.z.z.z



## Large BGP Table

(1)	Net a.a.a.a
.....	
.....	
(400 k)	Net z.z.z.z

(1)	Net 1.1.1.1
-----	-------------

---

# Cisco MPLS tools

# MPLS/LDP basic

- + ***Setup a MPLS LSP for each network prefix following the IP OSPF routes***
- + **Configure IGP (e.g. OSPF)**
- + **Configure label range**
  - **R2(config)# mpls label range 32 200 static 16 31**
- + **Enable MPLS general engine**
  - **R2(config)#mpls ip**
- + **Enable MPLS on single interface**
  - **R2(config-if)#mpls ip**
- + **Now LDP is active by default and ...**

# MPLS/LDP basic

- + LDP default behavior is *unsolicited downstream* mode
- + Allocate and announce to downstream LSRs a local label for
  - all non-BGP prefixes, which includes IGP learned prefixes and connected interfaces with LDP on
- + The downstream LSR inserts in its MPLS forwarding table only FEC/LABEL mappings coming from the Next-hop IP LSR for the related FEC
- + Debug commands
  - R2#show mpls forwarding-table
  - R2#show mpls ldp bindings
    - All association FEC label
  - show mpls ldp neighbor

# MPLS Traffic Engineering with tunnels

## + Global conf

- ip cef [distributed] (default)
- mpls traffic-eng tunnels

## + Link bandwidth information distribution

- router ospf 1
  - mpls traffic-eng router-id loopback0
  - mpls traffic-eng area *ospf-area*

## + On each physical interface

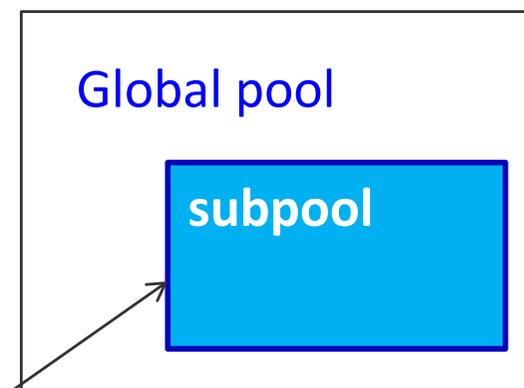
- interface f0/0
- mpls traffic-eng tunnels
- ip rsvp bandwidth *kbps* subpool *kbps*

# MPLS Traffic Engineering with tunnels

## + Build the tunnel

- interface Tunnel0
  - ip unnumbered loopback0
  - tunnel destination *RID-of-tail*
  - tunnel mode mpls traffic-eng

RSVP bandwidth pools (Russian Dolls)



## + Tunnel attributes

- interface Tunnel0
  - tunnel mpls traffic-eng bandwidth [sub-pool] *Kbps*
    - tunnel mpls traffic-eng bandwidth 1000
  - tunnel mpls traffic-eng priority *pri* [hold-pri] *pri* [setup-priority]
    - tunnel mpls traffic-eng priority 7 7
    - Lower is better
    - Hold>=Setup to avoid instability
  - tunnel mpls traffic-eng exp *value*
    - tunnel mpls traffic-eng exp 5

# MPLS Traffic Engineering with tunnels

## + **Dynamic path calculation**

- **int Tunnel0**
  - tunnel mpls traffic-eng path-option dynamic

## + **Explicit path calculation**

- **int Tunnel0**
  - tunnel mpls traffic path-opt explicit name foo
- **ip explicit-path name foo**
  - next-address 1.2.3.4 [loose]
  - next-address 1.2.3.8 [loose]

# MPLS Traffic Engineering with tunnels

## + Static routing to inject traffic on tunnel

- `ip route prefix mask Tunnel0`

## + Policy Routing

- `access-list 101 permit ip any any dscp 20`
- `access-list 102 permit ip any any dscp 0`
- `interface Serial0`
  - `ip policy route-map foo`
- `route-map foo permit 10`
  - `match ip address 101`
  - `set interface Tunnel0`
- `route-map foo permit 20`
  - `match ip address 102`
  - `set interface Tunnel1`
- **Be careful: MUST be possible to route the packet via the plain routing-table, also if a destination is forced with route-map**
  - in case routes for tunnels was not present, the routing-table can use a static route: `ip route 0.0.0.0 0.0.0.0 null 0`

## + Debug

- `show mpls traffic-eng topology`
- `show mpls traffic-eng tunnels`

# MPLS Traffic Engineering with tunnels

## + Mark exp field

- **class-map match-all voice**
  - match access-group 101
- **policy-map set-exp5t**
  - class voice
    - set mpls experimental imposition 5
- **interface FastEthernet2/0**
  - service-policy input set-exp5t

+ **E.g. mark with exp=5 every MPLS packet whose tunnel gathers bandwidth from the subpool**

+ **A scheduler based on exp field classification MUST be deployed in every network node**

- **MPLS DS-TE IS ONLY CONTROL PLANE**
- **No scheduler is actually deployed. Thus must be deployed manually node-by-node**
- **E.g. Priority bandwidth equal to the subpool bandwidth and CBQ bandwidth for the remaining part of the global pool**