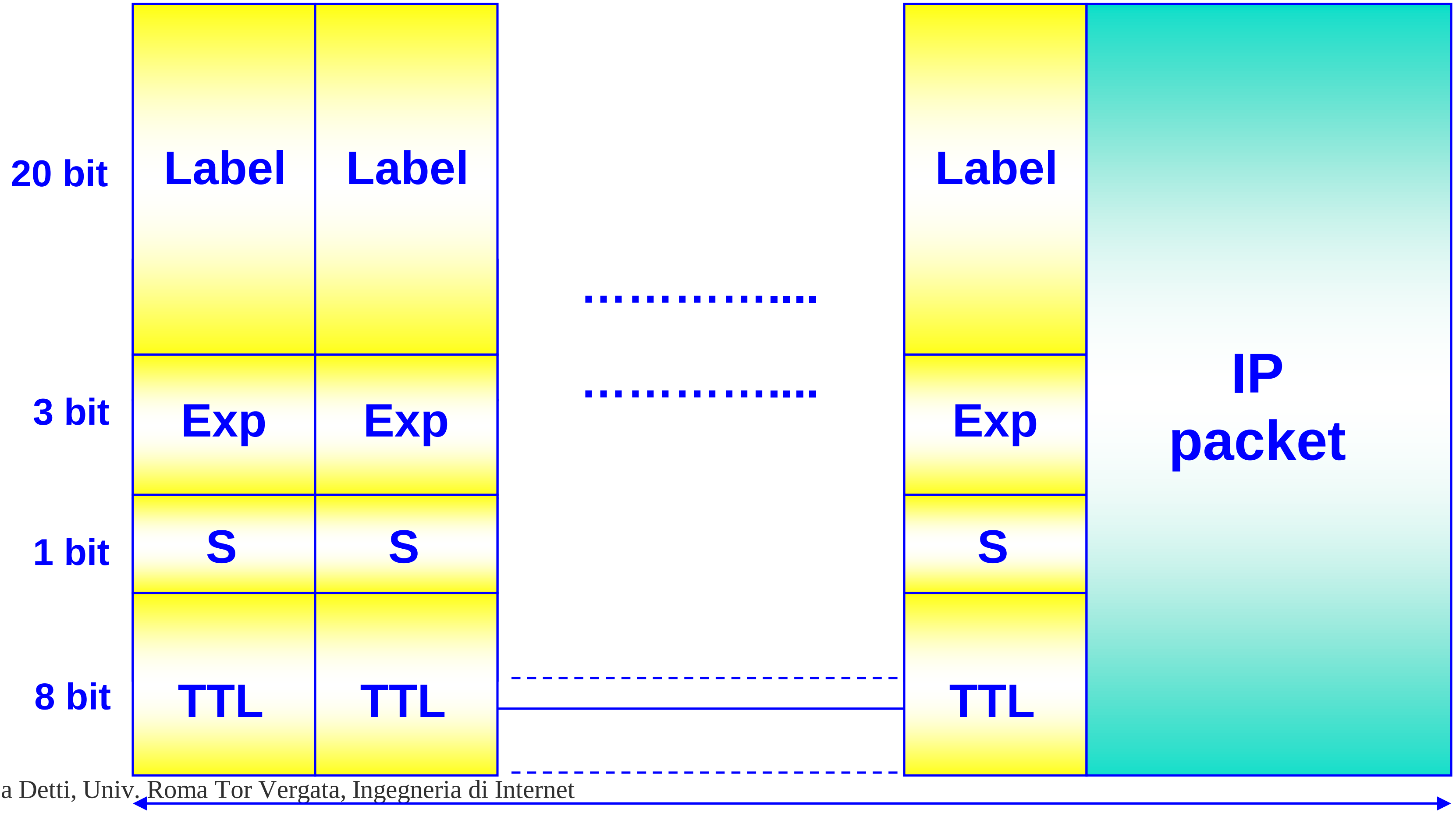
Multi Protocol Label Switching (QoS & Traffic Enginnering)

MPLS: architecture

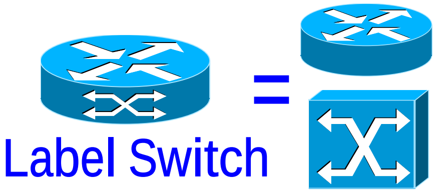
- The key idea of the MPLS architecture is to associate a brief identifier, namely *Label,* to every packet. Internetworking nodes can then apply fast forwarding mechanisms based on label switching / label swapping

- MPLS is independent both from the transport subnet (Frame Relay, ATM, etc.) both from adopted network protocols



MPLS node

MPLS network node



Label Switch =

Control component (router + LDP)

+

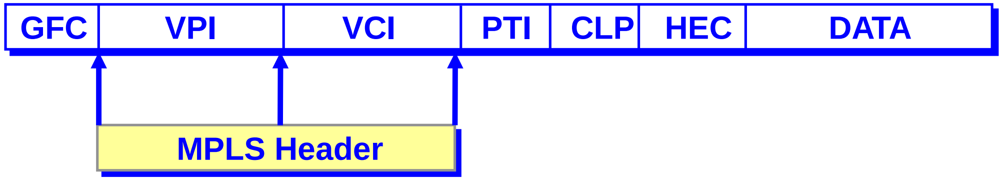
Forwarding component (L2 switch)

- Control Component
  - A set of modules dealing with Label allocation and binding Labels between adjacent nodes
  - Layer 3 «intelligence» (IP addressing, IP routing)
- Forwarding Component
  - Forwarding based on the *label swapping* paradigm
- The two components must be independent: they can employ different protocols within every medium
- The Control Component is sometimes realized as a part (SW or HW) of the network node, other times as external controller

Label encoding

- If data-link layer natively supports a field for the label (ATM does it with VPI/VCI, Frame Relay with DLCI), this can be used to insert the MPLS label
- If data-link layer doesn't support that field, the MPLS label is embedded in an MPLS header, inserted between layer 2 and layer 3 headers (e.g. Ethernet/MPLS/IP)

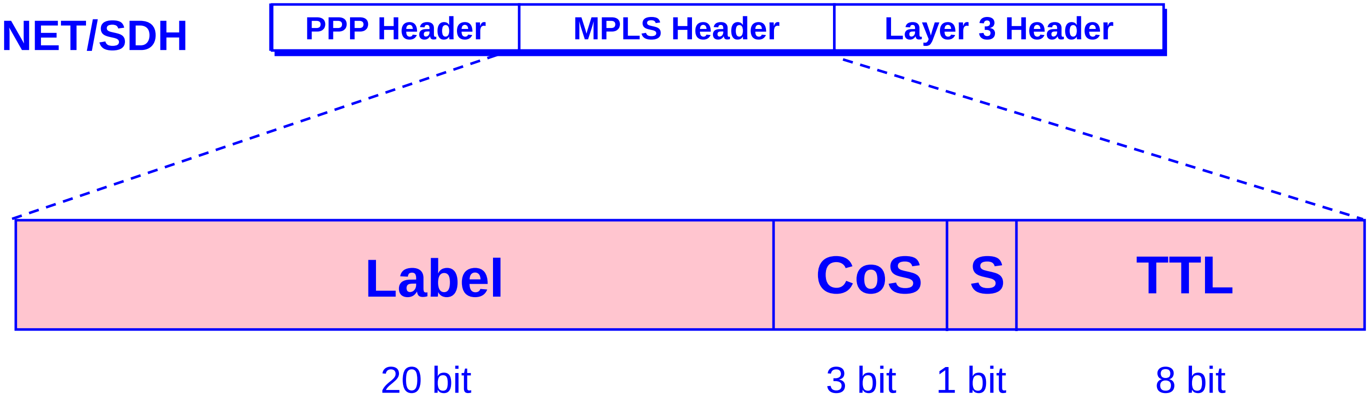
Header ATM cell



GFC	VPI	VCI	PTI	CLP	HEC	DATA

MPLS Header

Packet Over SONET/SDH



PPP Header	MPLS Header	Layer 3 Header

Label	CoS	S	TTL
20 bit	3 bit	1 bit	8 bit

Terminology

- **Label Edge Router (LER)**: edge routers for an MPLS network: they have forwarding functionalities from and to the outer networks, applying and removing the labels to ingress and egress packets

- **Label Switching Router (LSR)**: switches operating label swapping inside the MPLS network and supporting forwarding functionalities

- **Label Distribution Protocol (LDP)**: in conjunction with traditional routing protocols, LDP is used for distributing labels between network devices

- **Forwarding Equivalence Class (FEC)**: a set of IP packets that are forwarded in the same way (for instance along the same path, with the same treatment)

- **Label Switched Path (LSP)**: the path through one or more LSRs followed by a packet belonging to a certain FEC

Operation

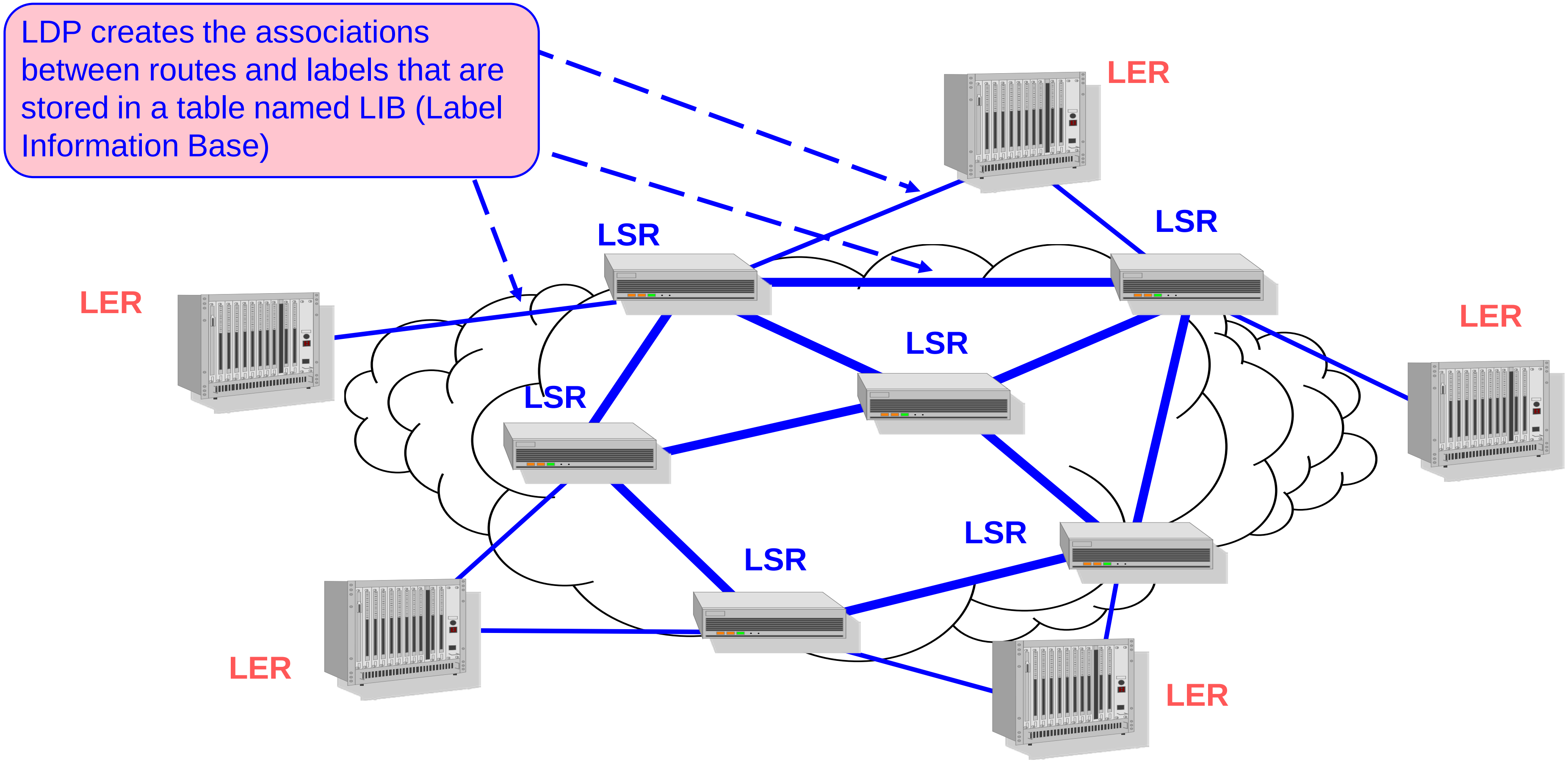
- The ingress LER of the MPLS backbone analyzes the packet’s IP header, classifies the packet, adds the label and forwards it to the next hop LSR

- In the LSRs cloud the packet is forwarded along the LSP according to the label

- The egress LER removes the label and the packet is forwarded based on IP destination address

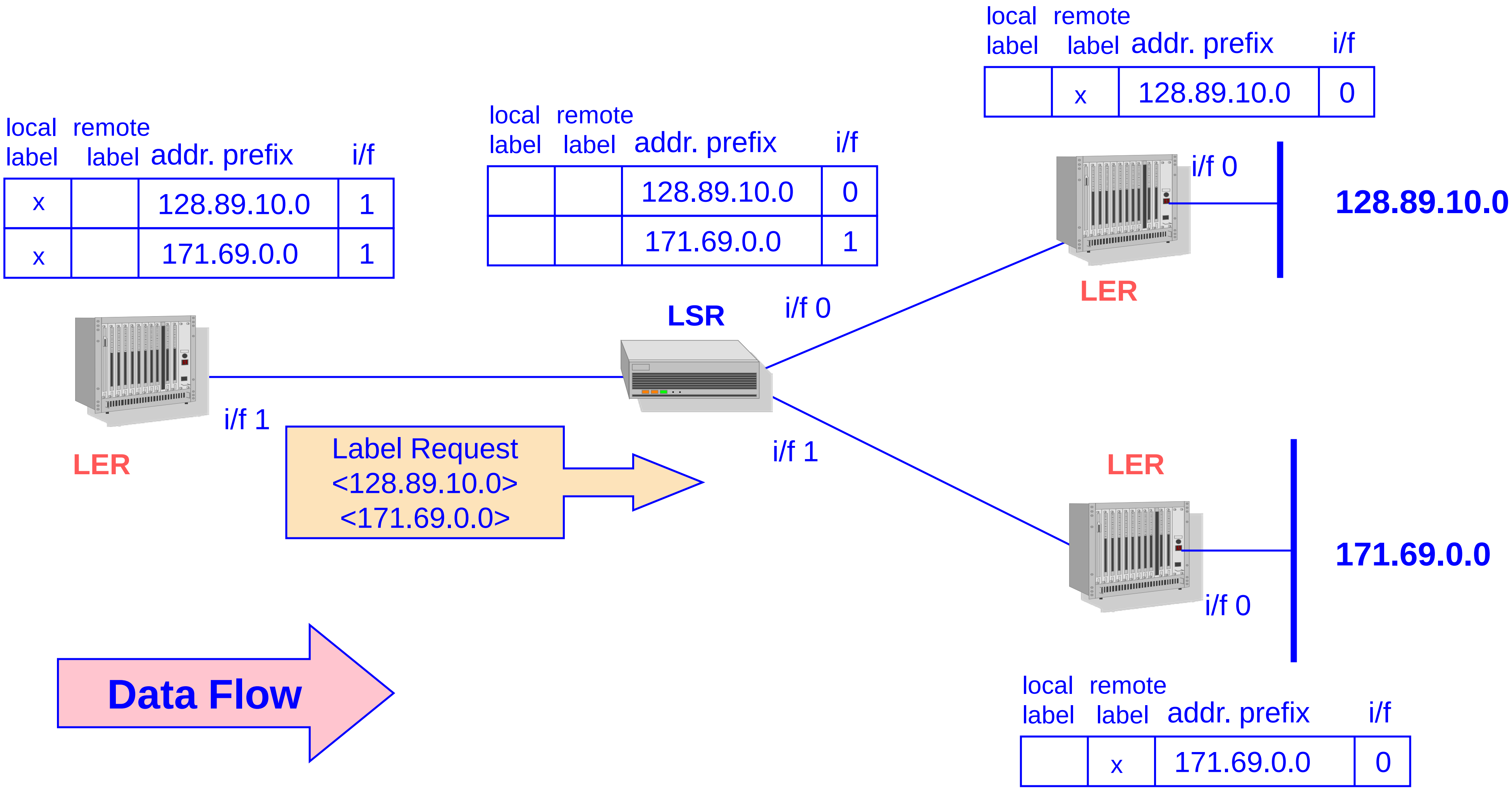
Label Switching Operation: Control

- **LDP is used for distributing the <label, prefix> associations between MPLS nodes**



LDP creates the associations between routes and labels that are stored in a table named LIB (Label Information Base)

LDP: Downstream on Demand



local label	remote label	addr. prefix	i/f
x		128.89.10.0	1
x		171.69.0.0	1

local label	remote label	addr. prefix	i/f
		128.89.10.0	0
		171.69.0.0	1

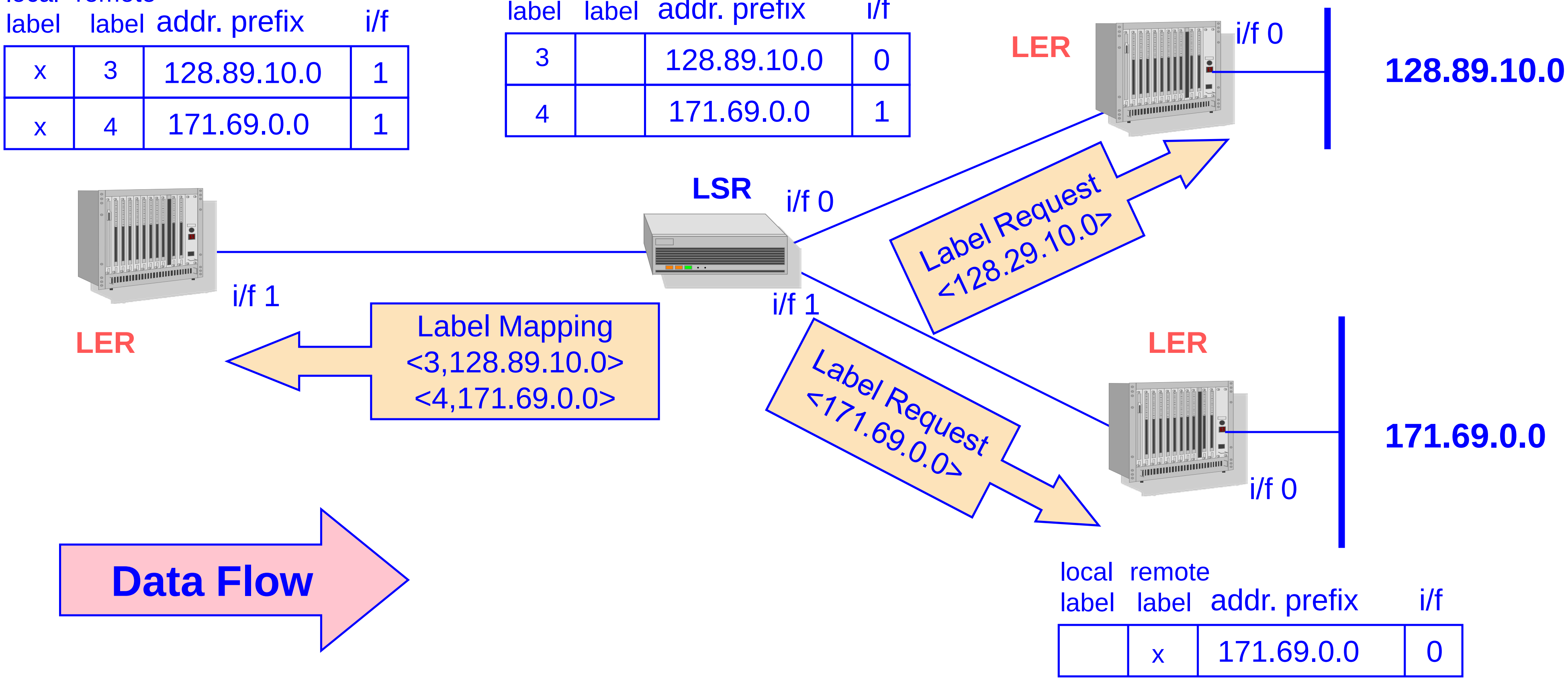
local label	remote label	addr. prefix	i/f
	x	128.89.10.0	0

local label	remote label	addr. prefix	i/f
	x	171.69.0.0	0

Label Request
<128.89.10.0>
<171.69.0.0>

Data Flow

LDP: Downstream on Demand



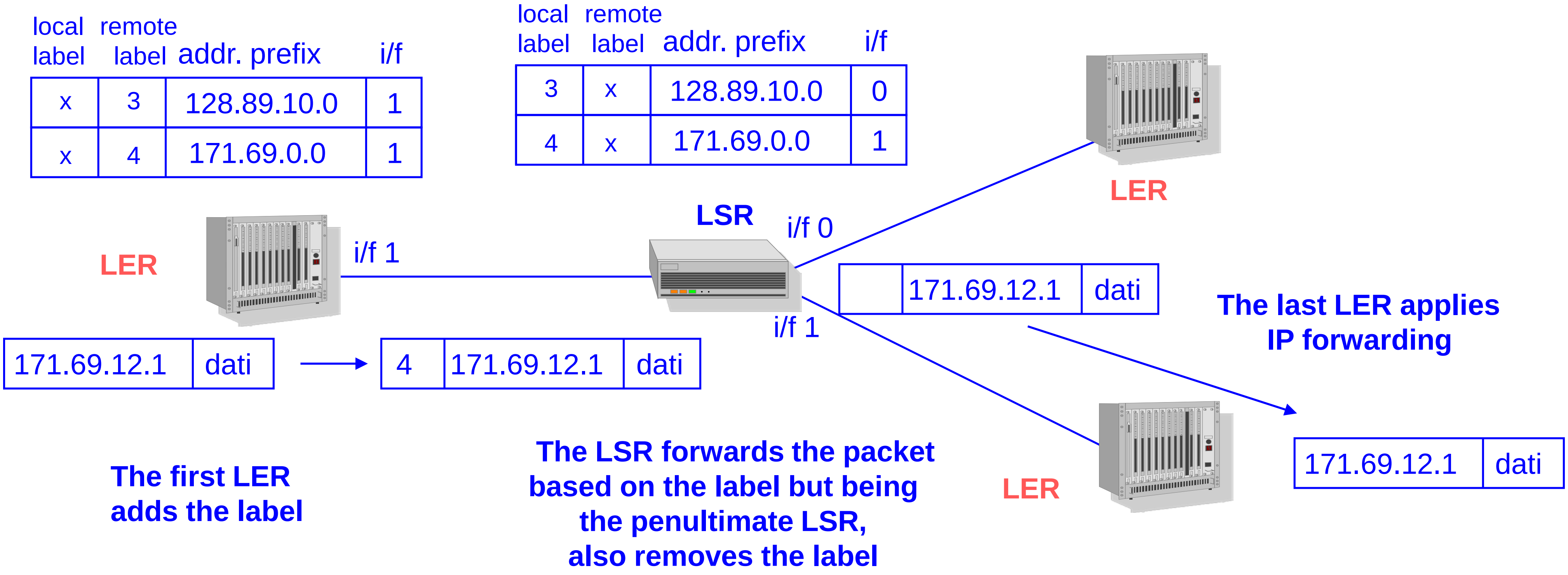
local label	remote label	addr. prefix	i/f
x	3	128.89.10.0	1
x	4	171.69.0.0	1

local label	remote label	addr. prefix	i/f
3		128.89.10.0	0
4		171.69.0.0	1

local label	remote label	addr. prefix	i/f
	x	128.89.10.0	0

local label	remote label	addr. prefix	i/f
	x	171.69.0.0	0

Label Switching Operation: Forwarding



local label	remote label	addr. prefix	i/f
x	3	128.89.10.0	1
x	4	171.69.0.0	1

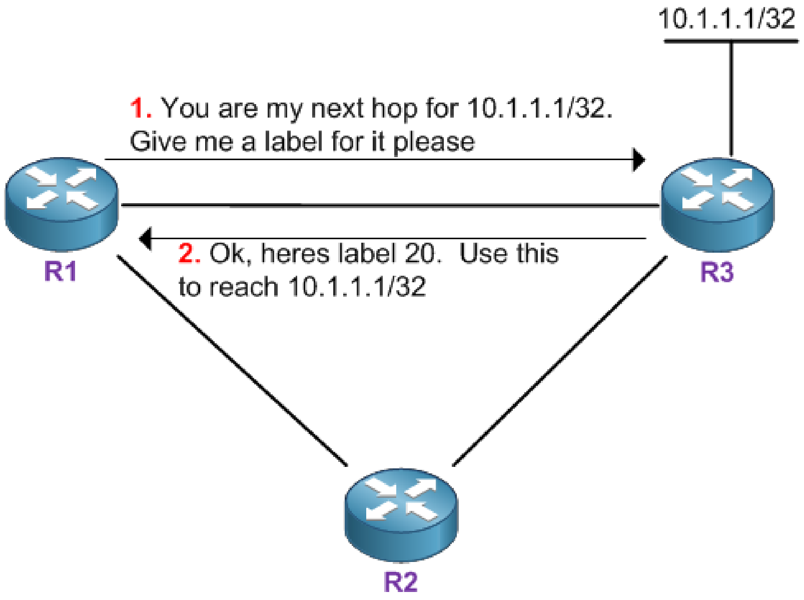
local label	remote label	addr. prefix	i/f
3	x	128.89.10.0	0
4	x	171.69.0.0	1

The first LER adds the label

The LSR forwards the packet based on the label but being the penultimate LSR, also removes the label

The last LER applies IP forwarding

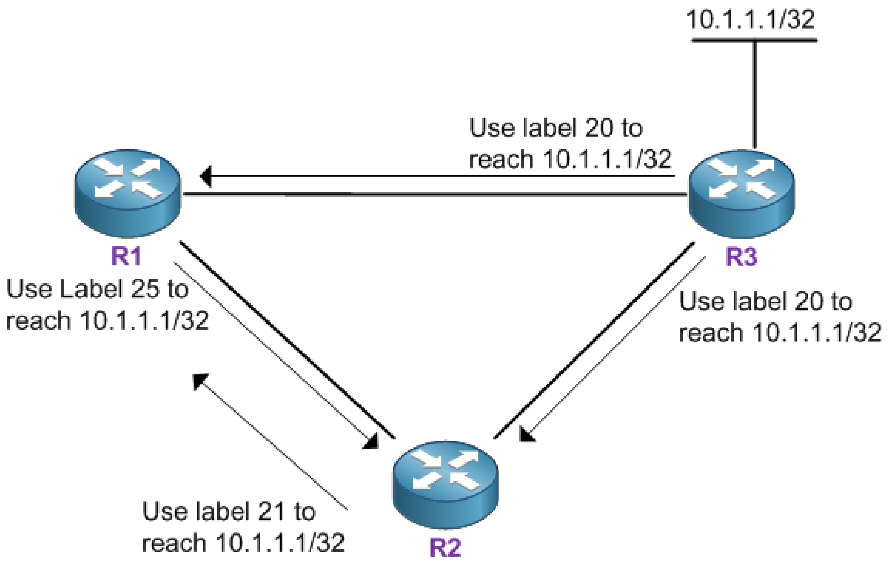
LDP: Downstream Unsolicited vs OnDemand



1. You are my next hop for 10.1.1.1/32. Give me a label for it please

2. Ok, heres label 20. Use this to reach 10.1.1.1/32

OnDemand



Use label 20 to reach 10.1.1.1/32

Use Label 25 to reach 10.1.1.1/32

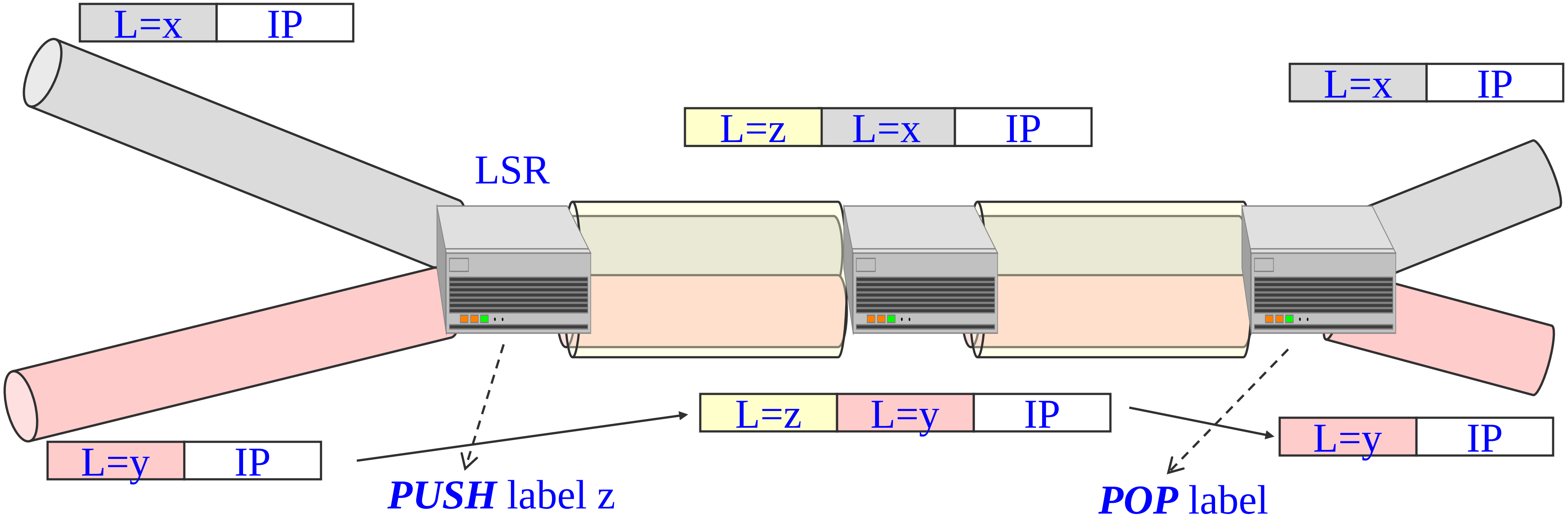
Use label 20 to reach 10.1.1.1/32

Use label 21 to reach 10.1.1.1/32

Unsolicited
(Cisco default)

Label Stacking

- **MPLS label can be stacked to aggregate, in a network section, two or more LSP in a single LSP with higher pecking order**
- **Label insertion is named after *label push***
- **Label removal is names after *label pop***
- **Forwarding is always made according to the highest order label; if there isn't a label, IP level forwarding is applied**



MPLS: reality

- Why do ISPs employ MPLS?
  - The key advantage is that MPLS enables an ISP the offering of new services that cannot be supported simply through conventional routing technology
- By now, there are three main MPLS use cases in ISP cores
  - Traffic Engineering (MPLS-TE)
  - Traffic Engineering with QoS (MPLS DS-TE)
  - Virtual Private Networks (VPN)

MPLS-TE

- Traffic Engineering enables the forwarding of a certain traffic flow along a path possibly different from the one calculated by the routing protocol. In this way it can use a less congested path (if necessary)

- This allows to ISPs the load balancing on the various links and network nodes so that none of them is under or over utilized

- MPLS-TE extends the base MPLS functionalities including:
  - Mechanisms for network monitoring of link utilization
  - Mechanism for RSVP-TE/CR-LSP signalling to setup LSPs with forced routing

Traffic Engineering: how?

- **Normally, an LSP is setup according to the computation made by the backbone routing protocol of the path with the lower cost**

- **Questa modalità non offre nessun valore aggiunto in termini di traffic engineering**

- **This mode doesn’t offer anything in terms of Traffic Engineering**

- **For the different setup of an LSP with respect to the one determined by the routing protocol, various mechanisms can be used:**
  - **Static configuration of all LRS in the LSP (in the same way an IP/ATM tranditional backbone is configured)**
  - **LER configuration with the whole path. Then the LER uses a modified version of RSVP protocol to install the LIBs for each LSR along the path (LSP)**

Routing of an LSP

- **The path an LSP has to follow in order to cross the links with appropriate capacity is usually pre-computed by an offline tool**

- **Knowing the output interfaces utilization of LERs is mandatory:**
  - **A) Proprietary solutions exploiting queries to LSR's MIB**
  - **B) Extension of link-state routing protocols (flooding of interfaces' information), ICP like OSPF or IS-IS, in a way that they are bringing also utilization state of resources. Then LERs (or a centralized management entity) can know about both topology and network utilization**

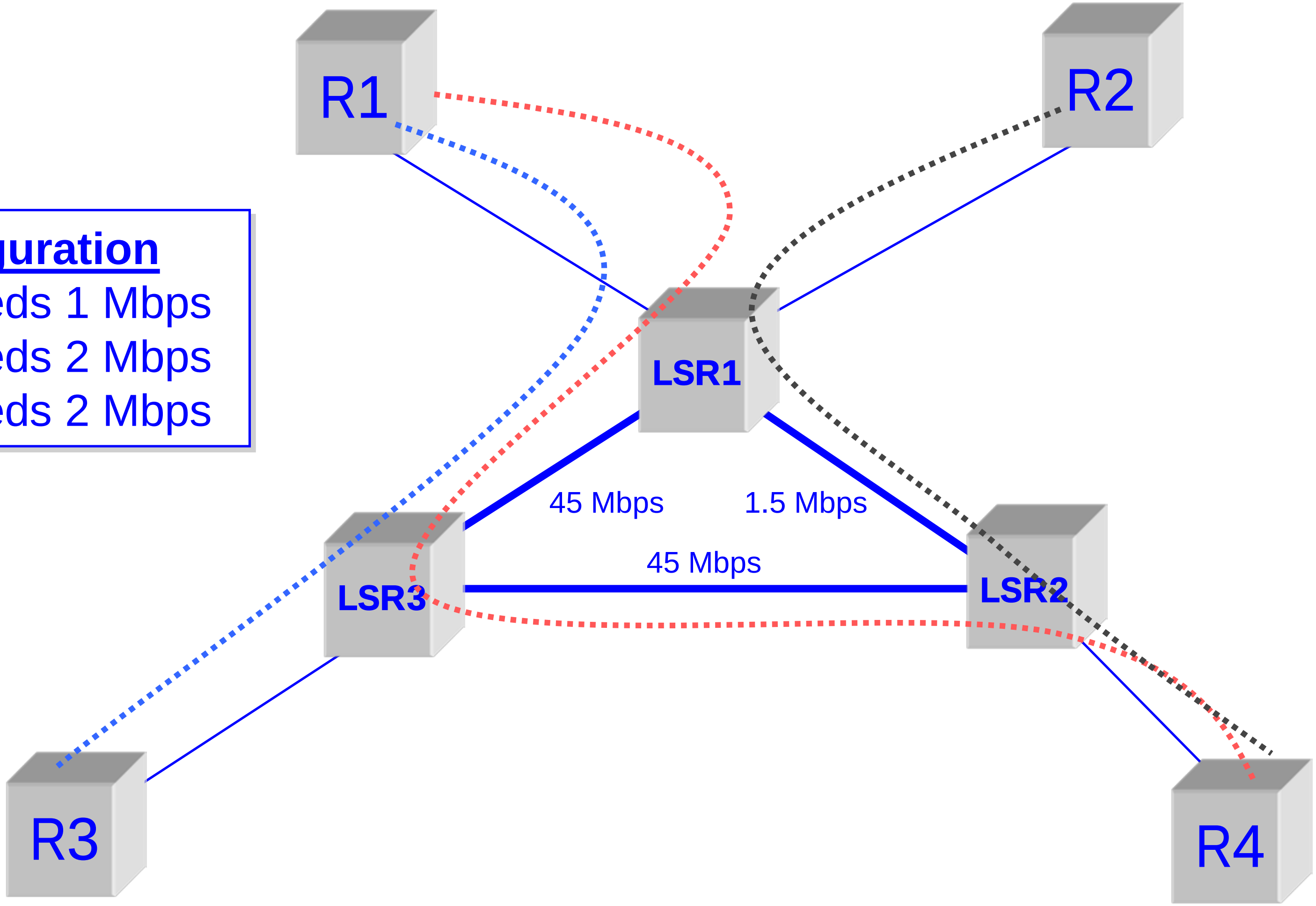
- **Path calculation through Constraint-based, Shortest Path First (CSPF)**
  - **Shortest path algorithm calculated upon the network topology except for the links that can't support the bandwidth of the LSP on which the setup is being made**

- **Manual setup or with RSVP-TE / CR-LDP**

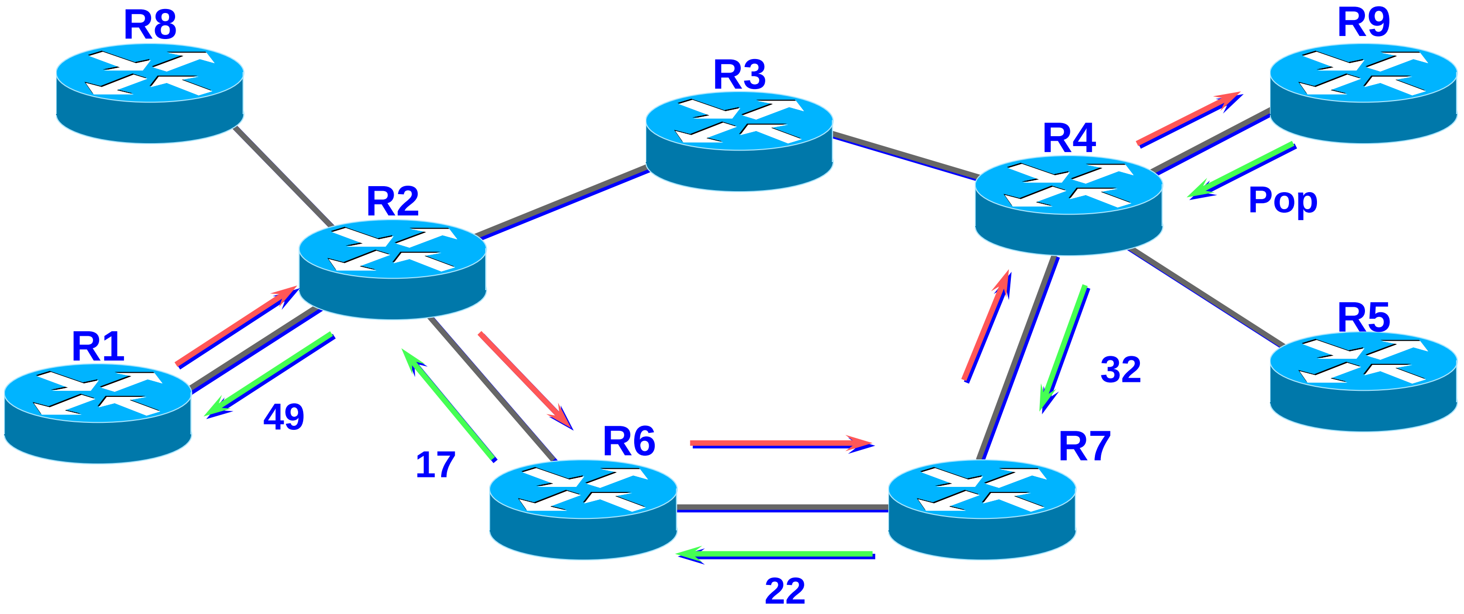
LSP: static configuration

Static configuration

- R2→R4 needs 1 Mbps
- R1→R3 needs 2 Mbps
- R1→R4 needs 2 Mbps



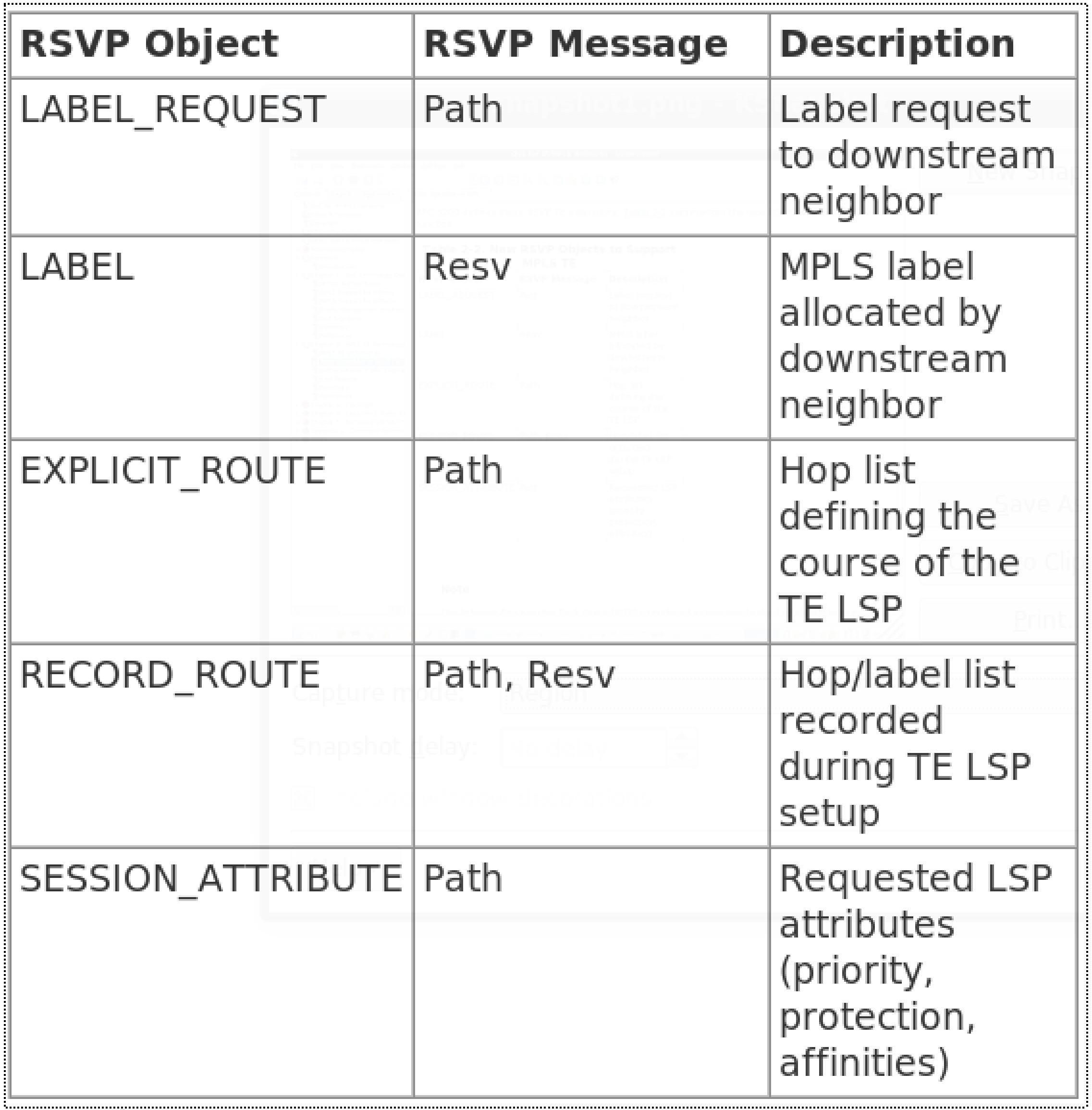
LSP: configuration with RSVP-TE



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

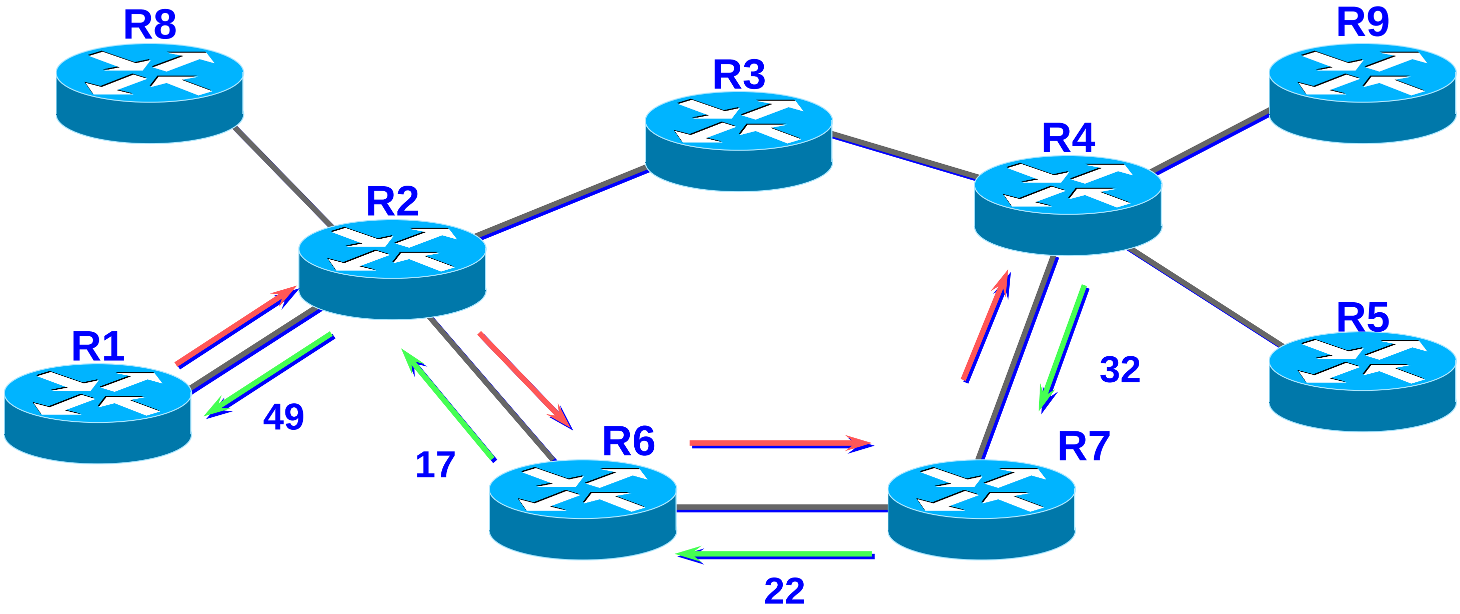
Reply: RESV (Notify the labels)

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: configuration with CR-LDP



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

Reply: Label mapping

MPLS & QoS

- The Engineering of traffic implies a planning of the resources usage in order to permit an effective transfer of data across the LSPs
- So traffic engineering tries to make links minimally loaded
- How do we handle different service classes?
- **Rule of thumb**: In the case the overall requested capacity from all LSPs on an output interface of an LSR is less or equal to half of the link capacity, all LSPs will experience a low delay, so scheduling mechanisms (e.g. WDRR) aren’t required
- When, in post traffic engineering, the interfaces capacity start to work with loads >> 0.5, a differenciation on how the traffic is handled is necessary
- MPLS can cooperate with DiffServ

MPLS and DiffServ

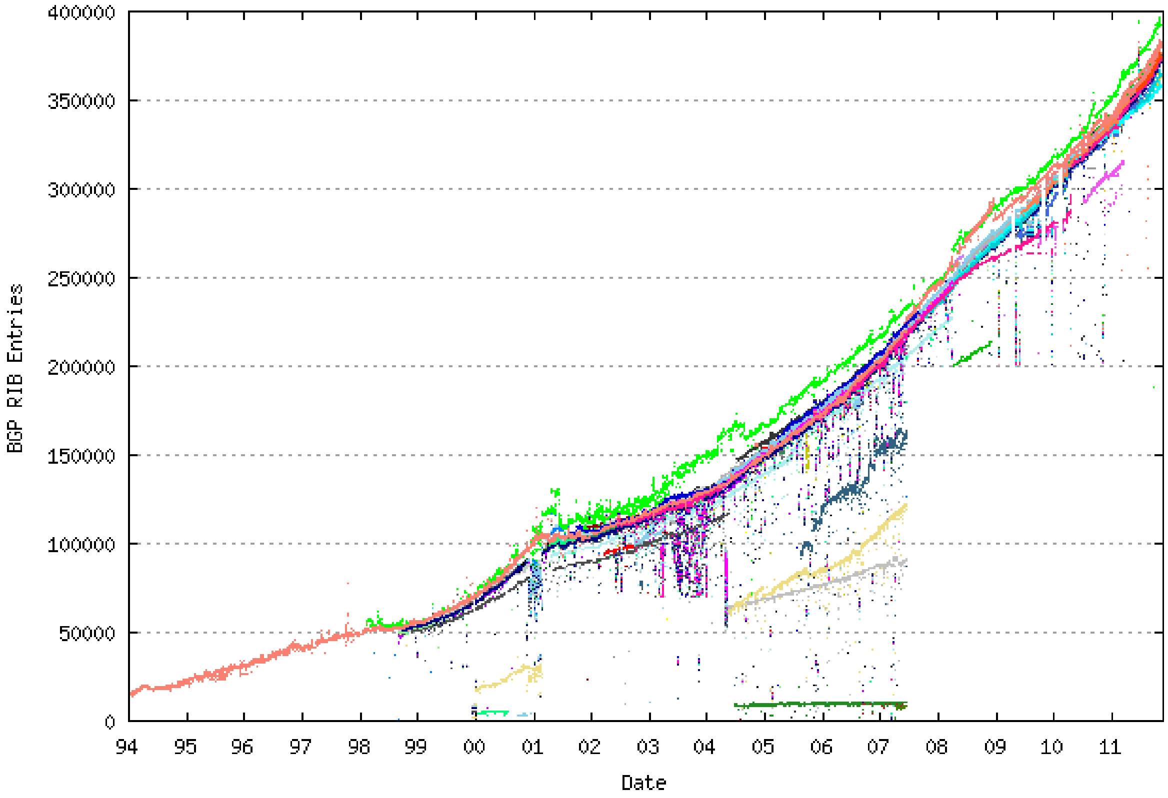
- **What is the classification criteryon an LSR adopts to determine the scheduler queue occupation (i.e. the DiffServ forwarding-behavior)?**

- **Two solutions:**
  - **Exp inferred LSP (E-LSP)**
    - Scheduler classification is made through **Exp** (3 bit) field of the MPLS header
    - Forwarding behavior and drop precedence inferred by the Exp field codification
    - Pachetti di LSP diversi con lo stesso campo Exp sono trattati ugualmente
    - Different LSP packets with the same Exp field are treated equally
    - Requires a maximum of 8 scheduler queues, the number of the possible values of the Exp field

  - **Label inferred LSP (L-LSP)**
    - Forwarding behavior is label inferred, drop precedence is **Exp** inferred
    - Each LSP can be handled with a different forwarding behavior regardless of Exp field
    - Requires a variable number of scheduler queues
    - More complex but more versatile
    - The <forwarding-behavior, label> association must be explicitly signalled during the LSP setup

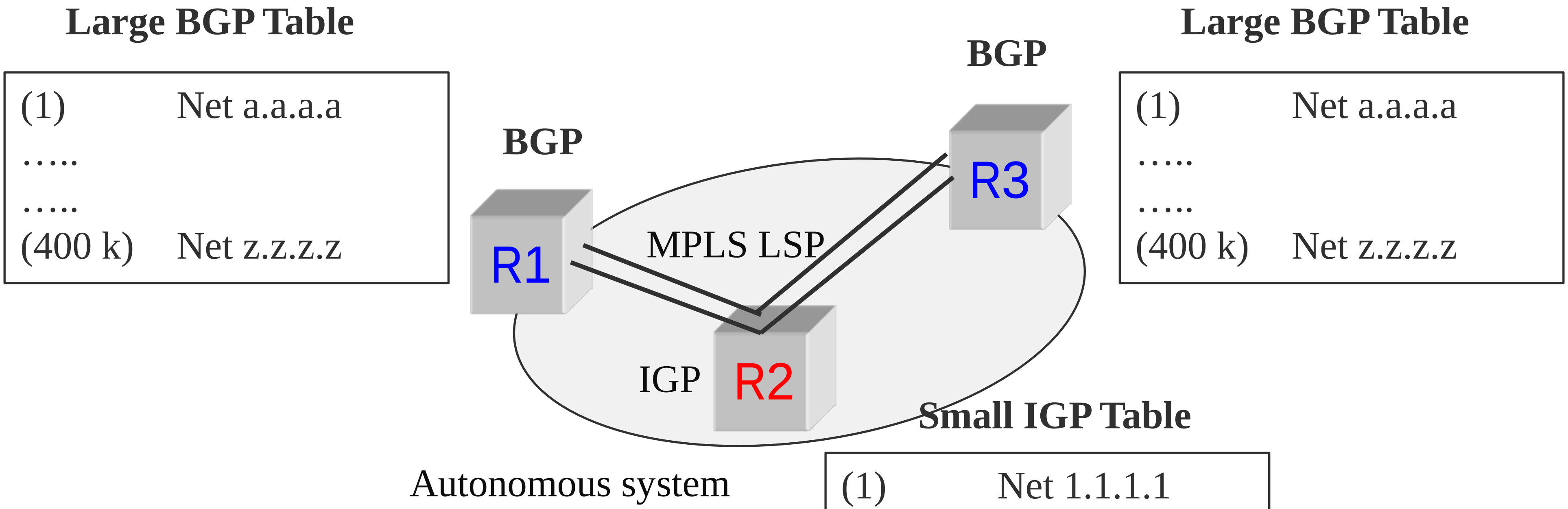
MPLS & BGP

- BGP is the routing protocol used between ASs
- BGP is executed by AS border gateways
- BGP tables contain all the routes to the Internet
  - 2018 almost 800k routes (http://bgp.potaroo.net/)



MPLS & BGP

- **Problem: how can internal routers (e.g. R2) forward transit packets, i.e. intended to one of the 800k external routes?**
  - **Replicate BGP tables also in core routers (costly)**
  - **<u>Full mesh</u> LSPs between border routers through which only transit traffic is forwarded**
    - Internal routers only matters about routing tables to reach internal network nodes



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

(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

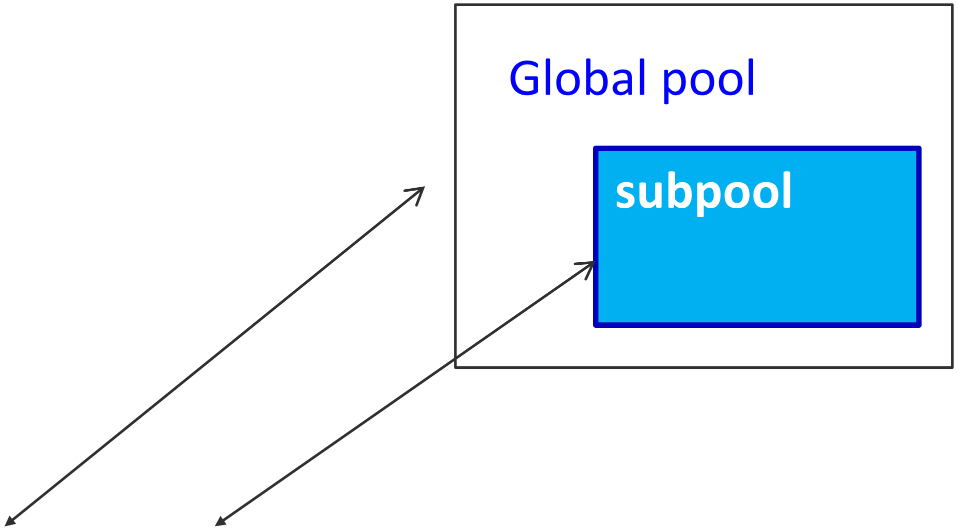
- **Global conf**
  - **ip cef [distributed] (default)**
  - **mpls traffic-eng tunnels**
- **Link bandwidth information distribution**
  - **router ospf 1**
    - mpls traffic-eng router-id loopback0
    - mpls traffic-eng area *ospf-area*
- **On each physical interface**
  - **interface f0/0**
  - **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**