MPLS Tutorial

SANOG VIII- Karachi

August 1, 2006

Mukhtiar A. Shaikh (mshaikh@cisco.com)

Yousuf Hasan (yhasan@cisco.com)

Mossadaq Turabi (mturabi@cisco.com)

Agenda

- **MPLS Basics**
- **LDP Fundamentals**
- **MPLS VPN Overview**
- **MPLS Traffic Engineering and Fast Reroute (FRR)**
- **L2VPN (Pseudowires)**

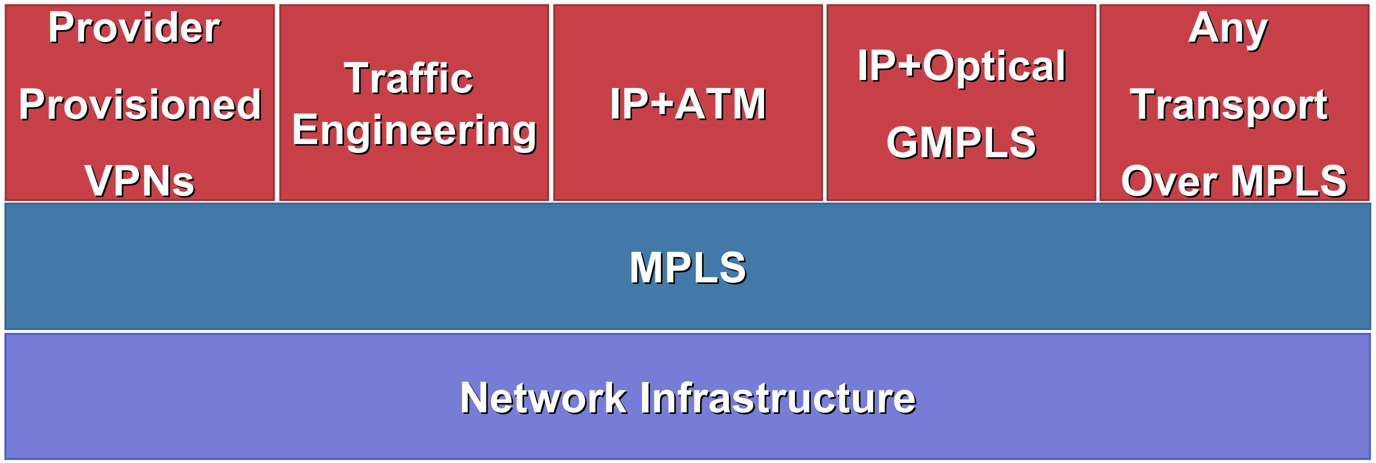
MPLS Basics

Agenda

- Introduction
- MPLS Concepts
- MPLS Applications
- MPLS Components
- MPLS Forwarding
- Basic MPLS Applications
  - Hierarchical Routing
  - IP+ATM Integration
- Summary and Benefits of MPLS

What Is MPLS?

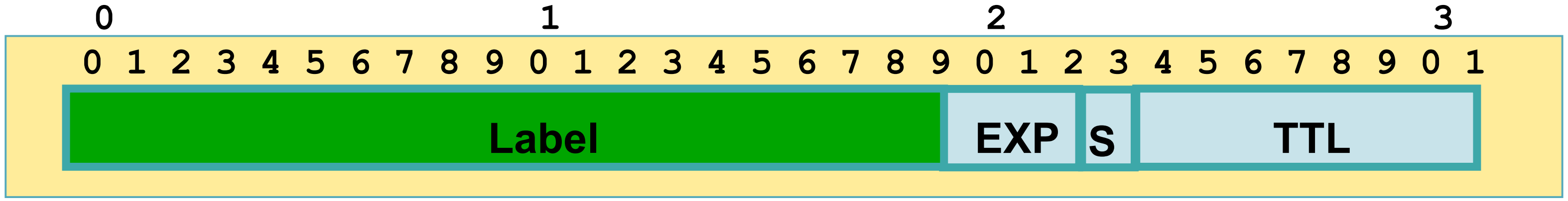
- Multi Protocol Label Switching
- Uses “Labels” appended to packets (IP packets, AAL5 frames) for transport of data
- MPLS packets can run on other layer 2 technologies such as ATM, FR, PPP, POS, Ethernet
- Other layer 2 technologies can be run over an MPLS network
- MPLS is a foundation technology for delivery of IP and other Value Added Services



MPLS concepts

- Packet forwarding is done based on labels
- Labels assigned when the packet enters the network
- Labels inserted between layer 2 and layer 3 headers
- MPLS nodes forward packets based on the label
- Separates ROUTING from FORWARDING
  - Routing uses IP addresses
  - Forwarding uses Labels
- Labels can be stacked

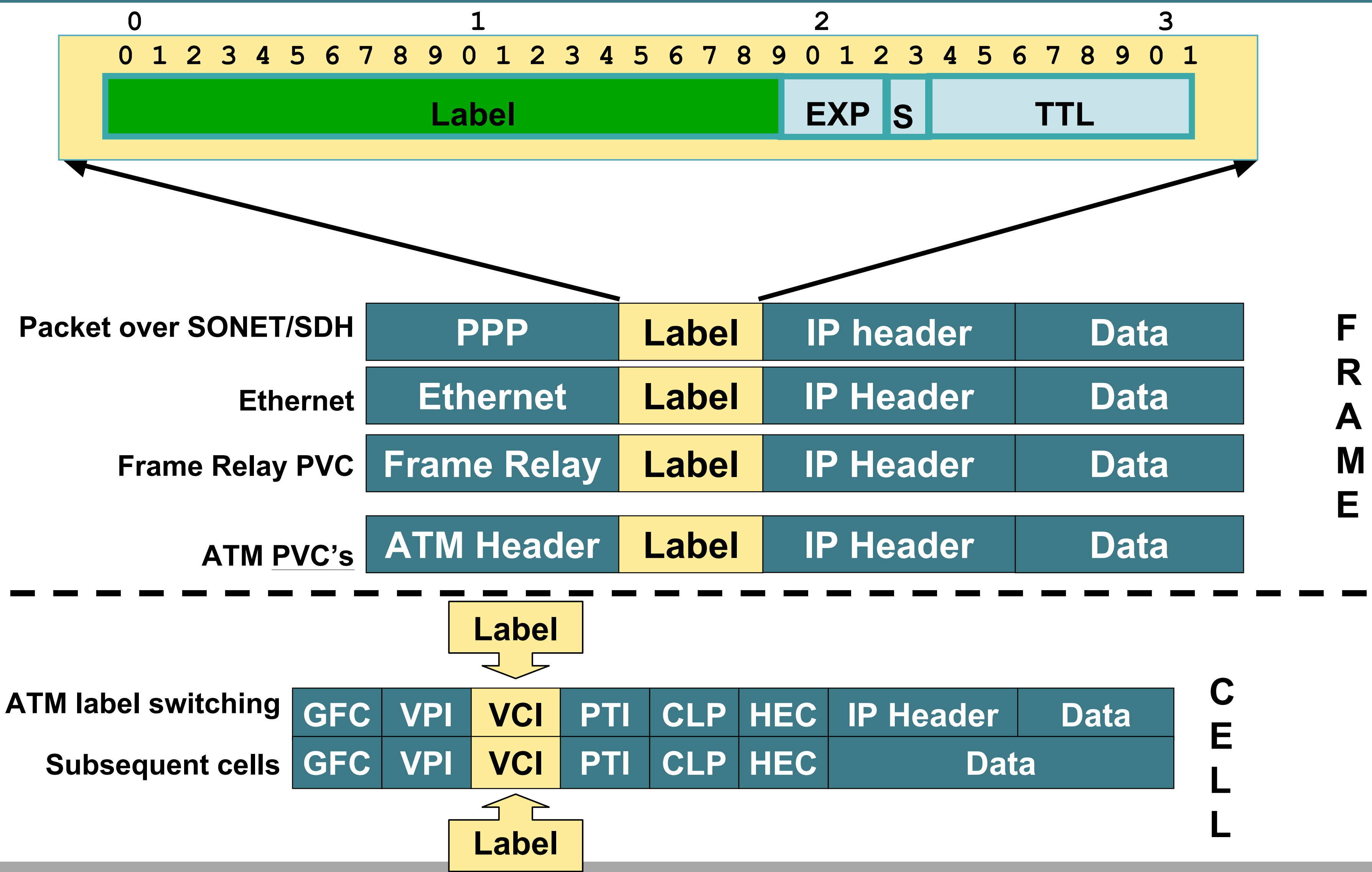
Label Format



Label = 20 Bits
COS/EXP = Class of Service, 3 Bits
S = Bottom of Stack, 1 Bit
TTL = Time to Live, 8 Bits

- Can be used over Ethernet, 802.3, or PPP links
- Ethertype 0x8847
- One for unicast, one for multicast
- Four octets per label in stack

Label Encapsulations



MPLS Applications

© 2003 Cisco Systems, Inc. All rights reserved.

9

Relevant MPLS Capabilities

- The ability to FORWARD on and STACK LABELS allows MPLS to provide some useful features including:
- IP+ATM Integration

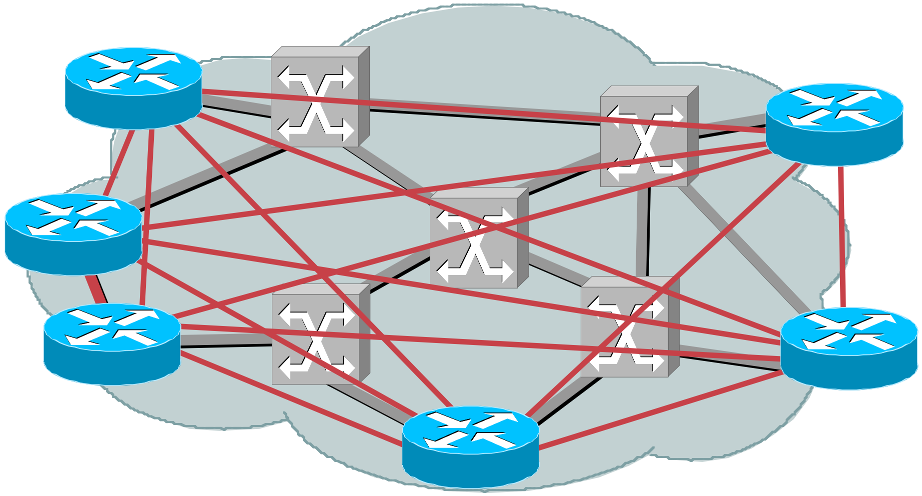
    Provides Layer 3 intelligence in ATM switches
- Virtual Private Networks

    Layer 3 – Provider has knowledge of customer routing

    Layer 2 – Provider has no knowledge of customer routing
- Traffic Engineering

    Force traffic along predetermined paths

Traditional IP over ATM



- **Put routers around the edge of an ATM network**
- **Connect routers using Permanent Virtual Circuits**
- **This does *not* provide optimal integration of IP and ATM**

IP+ATM Integration

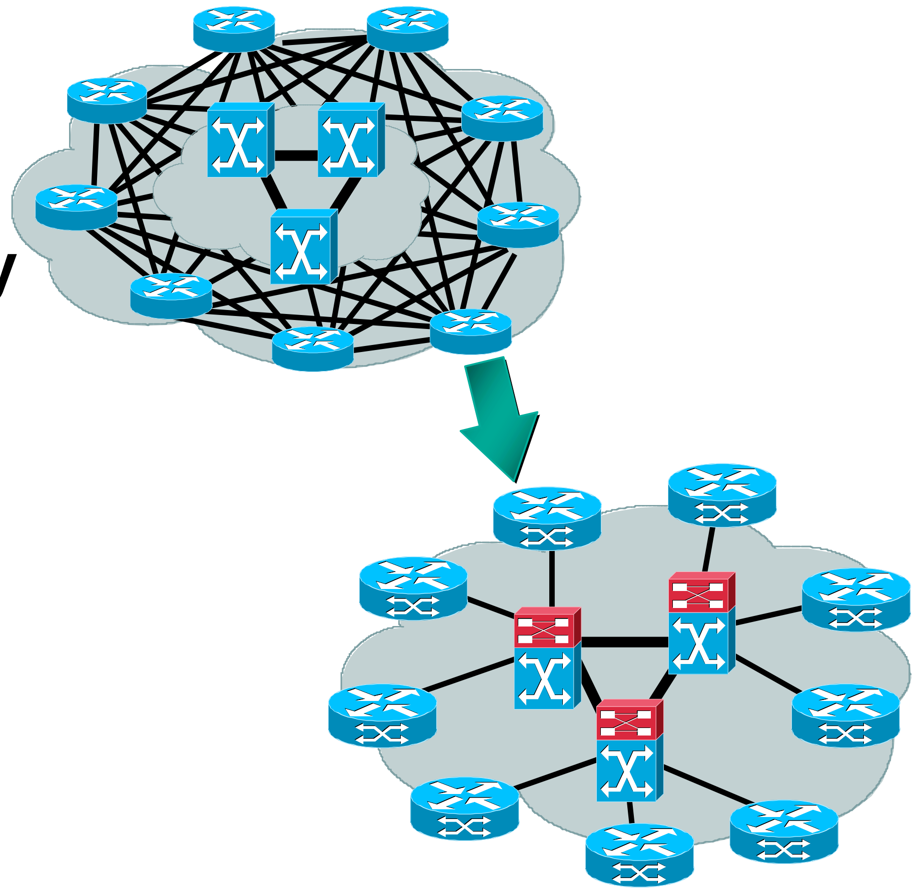
- **Internal routing scalability**

  Limited adjacencies

- **External routing scalability**

  Full BGP4 support, with all the extras

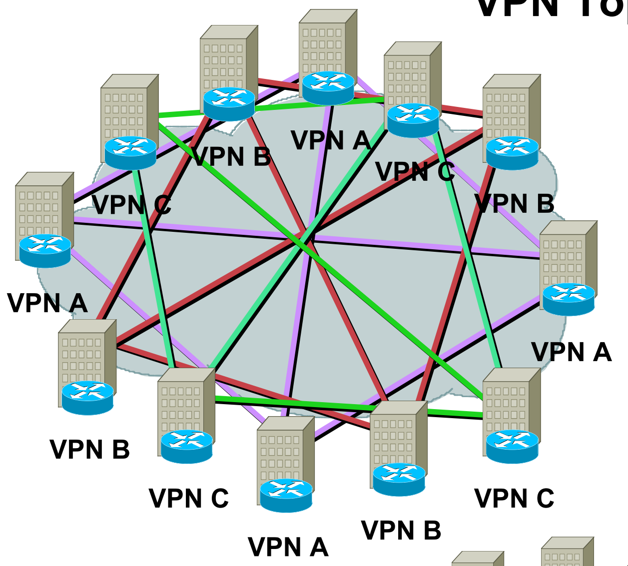
- **VC merge for very large networks**



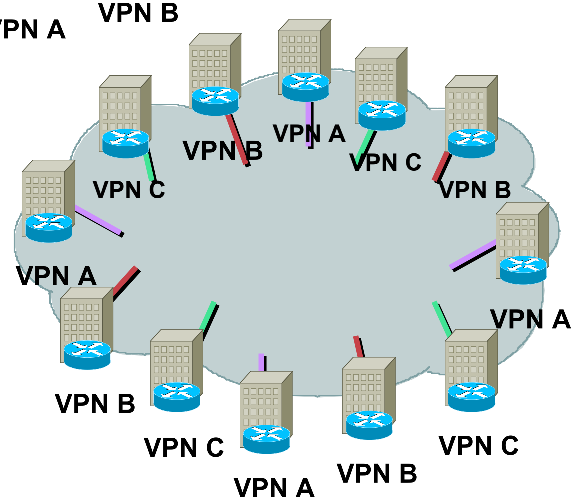
MPLS VPN – Layer 3

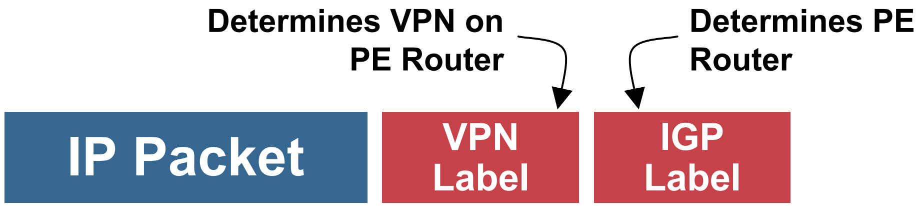
- Private, connectionless IP VPNs
- Outstanding scalability
- Customer IP addressing freedom
- Multiple QoS classes
- Secure support for intranets and extranets
- Easy to provide Intranet/Extranet/3rd Party ASP
- Support over any access or backbone technology

Connection-Oriented VPN Topology



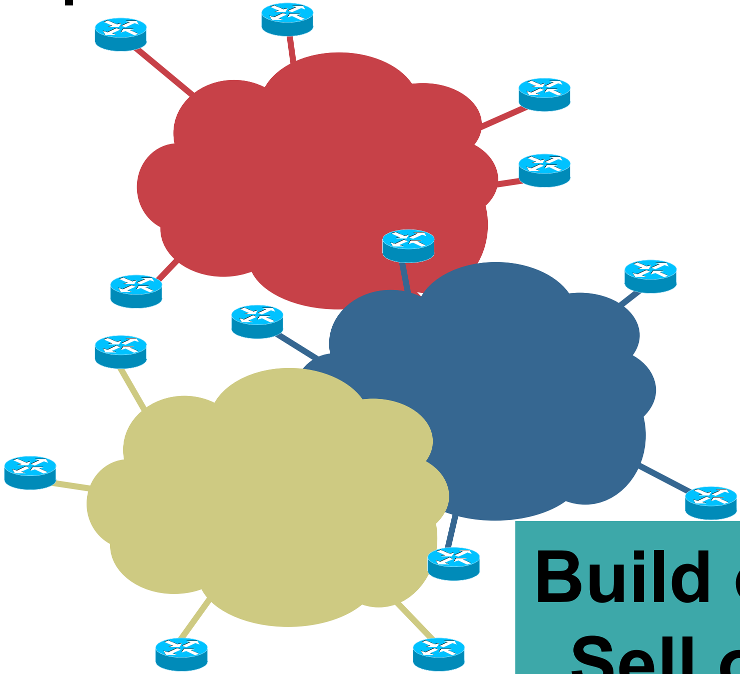
Connectionless VPN Topology



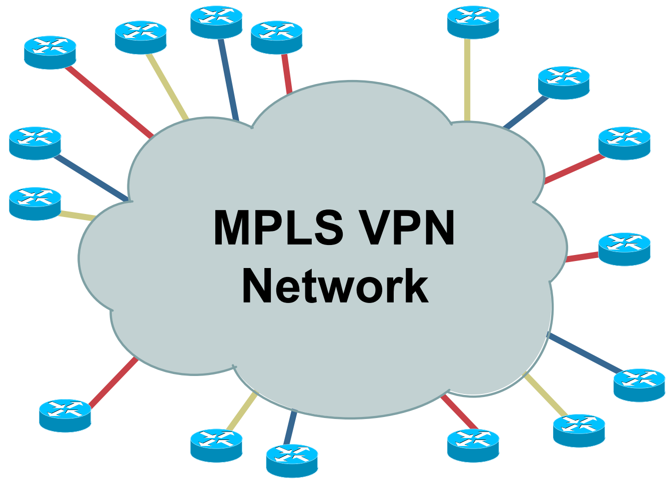


Why Providers like MPLS VPN…

Separately engineered private IP networks vs **Single network supporting multiple VPNs**



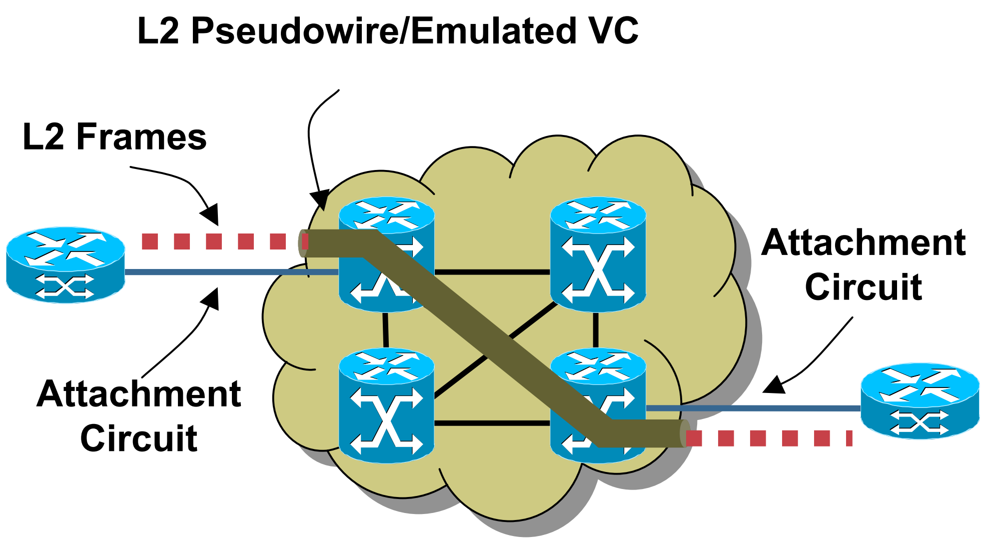
Build once, Sell once

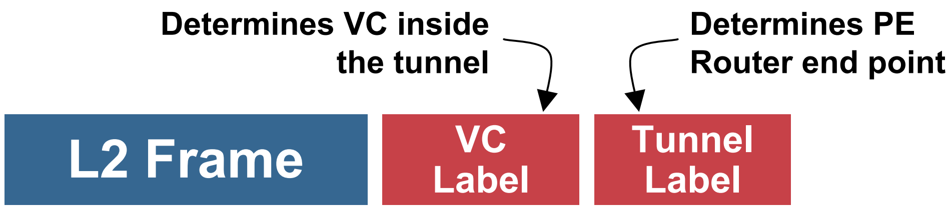


Build once, Sell many

MPLS VPN – Layer 2

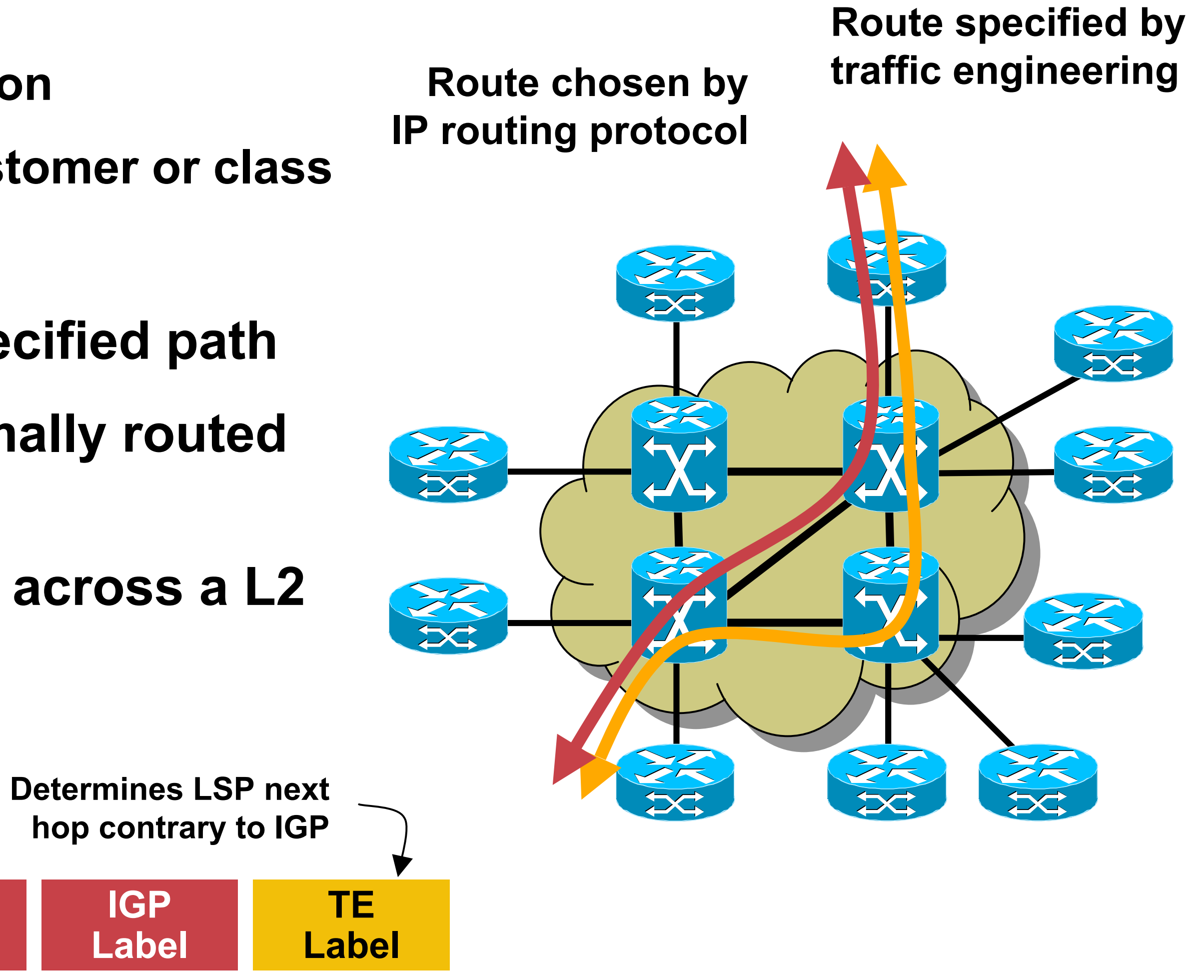
- **Additional Capabilities:**
  - Virtual leased line service
  - Offer "PVC-like" Layer 2-based service
- **Reduced cost—consolidate multiple core technologies into a single packet-based network infrastructure**
- **Simpler provisioning of L2 services**
- **Attractive to Enterprise that wish keep routing private**





Traffic Engineering

- **Why traffic engineer?**
  - **Optimise link utilization**
  - **Specific paths by customer or class**
  - **Balance traffic load**
- **Traffic follows pre-specified path**
- **Path differs from normally routed path**
- **Controls packet flows across a L2 or L3 network**



MPLS Components

© 2003 Cisco Systems, Inc. All rights reserved.

17

MPLS Components

- **Edge Label Switching Routers (ELSR or PE)**

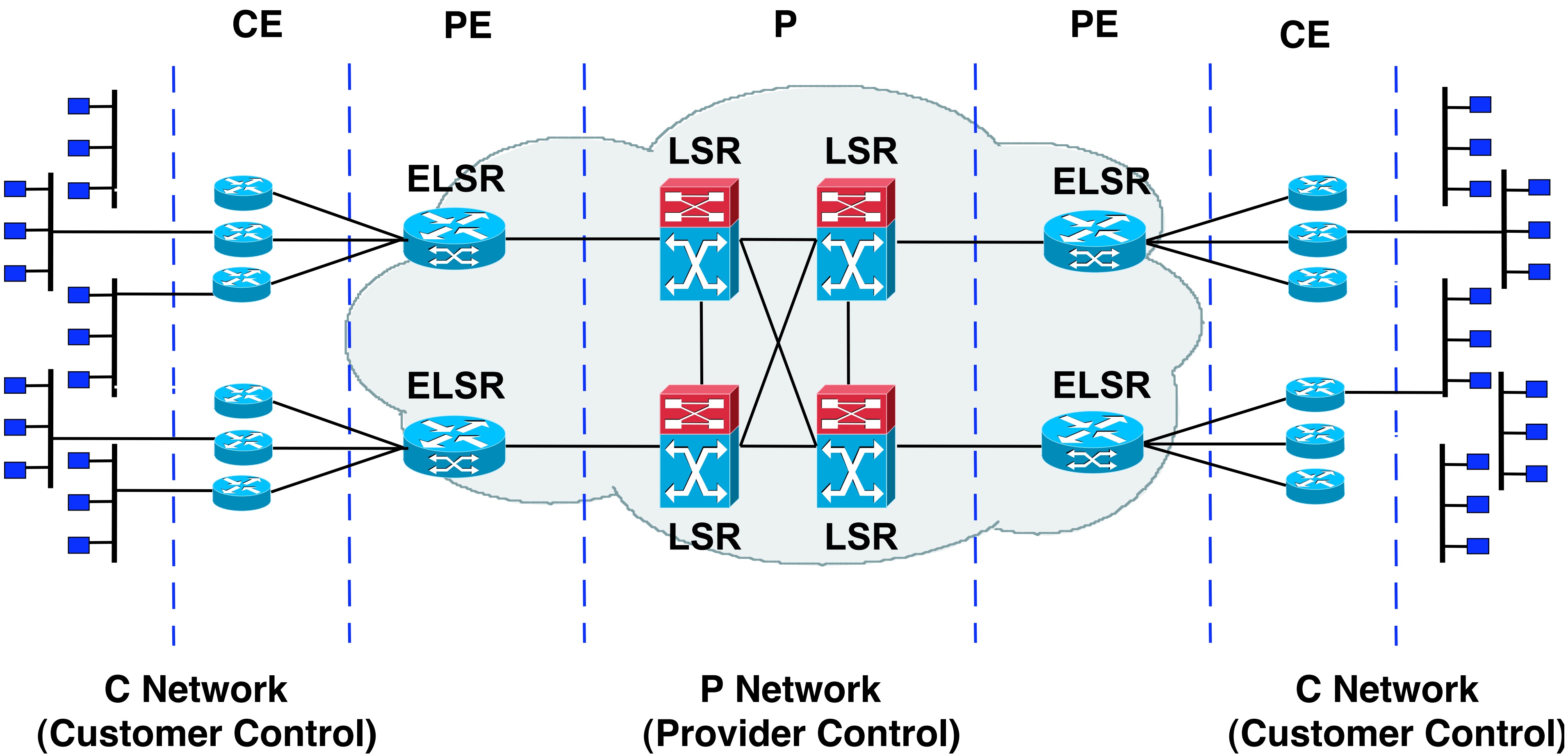
  Label previously unlabeled packets - at the beginning of a Label Switched Path (LSP)

  Strip labels from labeled packets - at the end of an LSP

- **Label Switching Routers (LSR or P)**

  Forward labeled packets based on the information carried by labels

MPLS Components



Functional Components

- **Forwarding component**

  Uses label information carried in a packet and label binding information maintained by a Label Switching Router to forward the packet

- **Control component**

  Responsible for maintaining correct label binding information among Label Switching Routers

Forwarding Component

- **Label Forwarding Information Base (LFIB)**
- **Each entry consists of:**

  incoming label
  outgoing label
  outgoing interface
  outgoing MAC address

- **LFIB is indexed by incoming label**
- **LFIB could be either per Label Switching Router or per interface**

Control Component

- **Labels can be distributed by several protocols**

  TDP/LDP – from IGP routes

  RSVP – for traffic engineering paths

  BGP – for VPN routes

- **Responsible for binding between labels and routes:**

  Create label binding (local)

  Distributing label binding information among Label Switching Routers

MPLS Forwarding Decisions

- Packets are forwarded based on the label value
- IP header and forwarding decision have been de-coupled for better flexibility
- No need to strictly follow unicast destination based routing
- Forwarding algorithm

  Extract label from a packet

  Find an entry in the LFIB with the INCOMING LABEL equal to the label in the packet

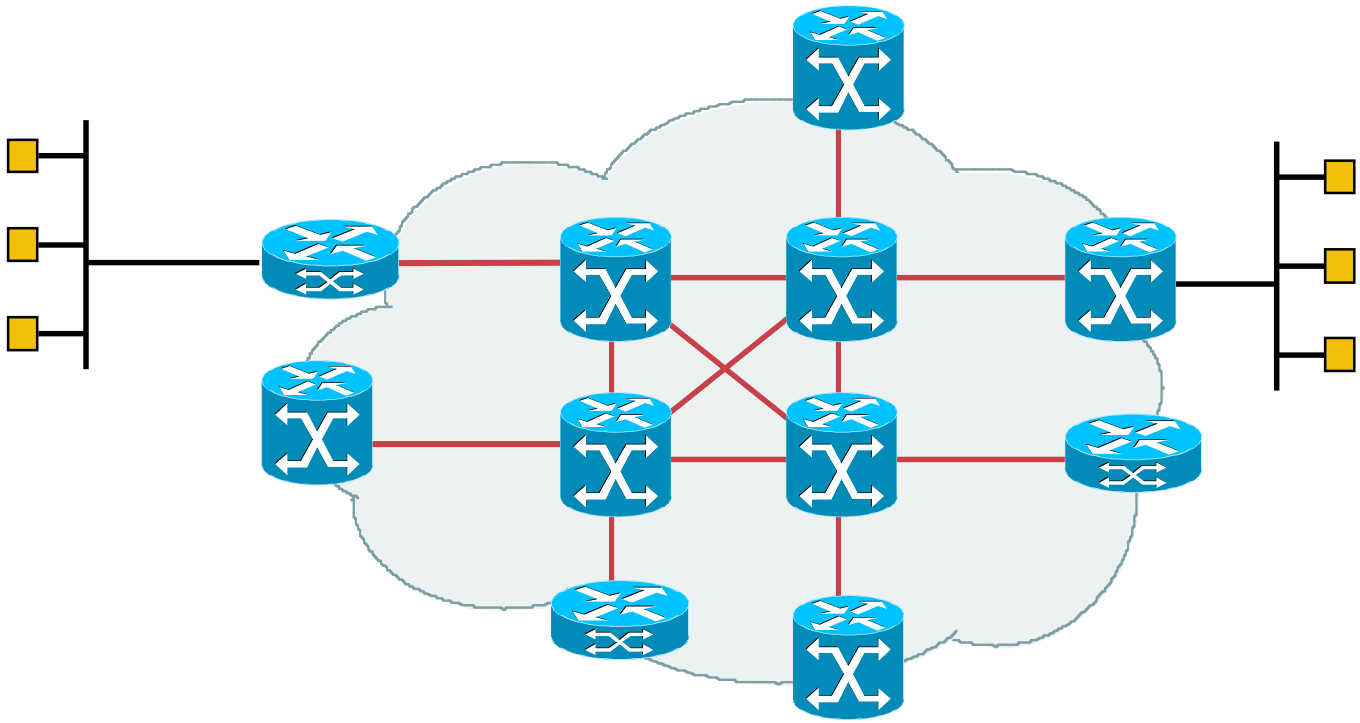
  Replace the label in the packet with the OUTGOING LABEL (from the found entry) and carry the label as part of the mac (layer2) header.

  Send the packet on the outgoing interface (from the found entry)

Basic MPLS Forwarding

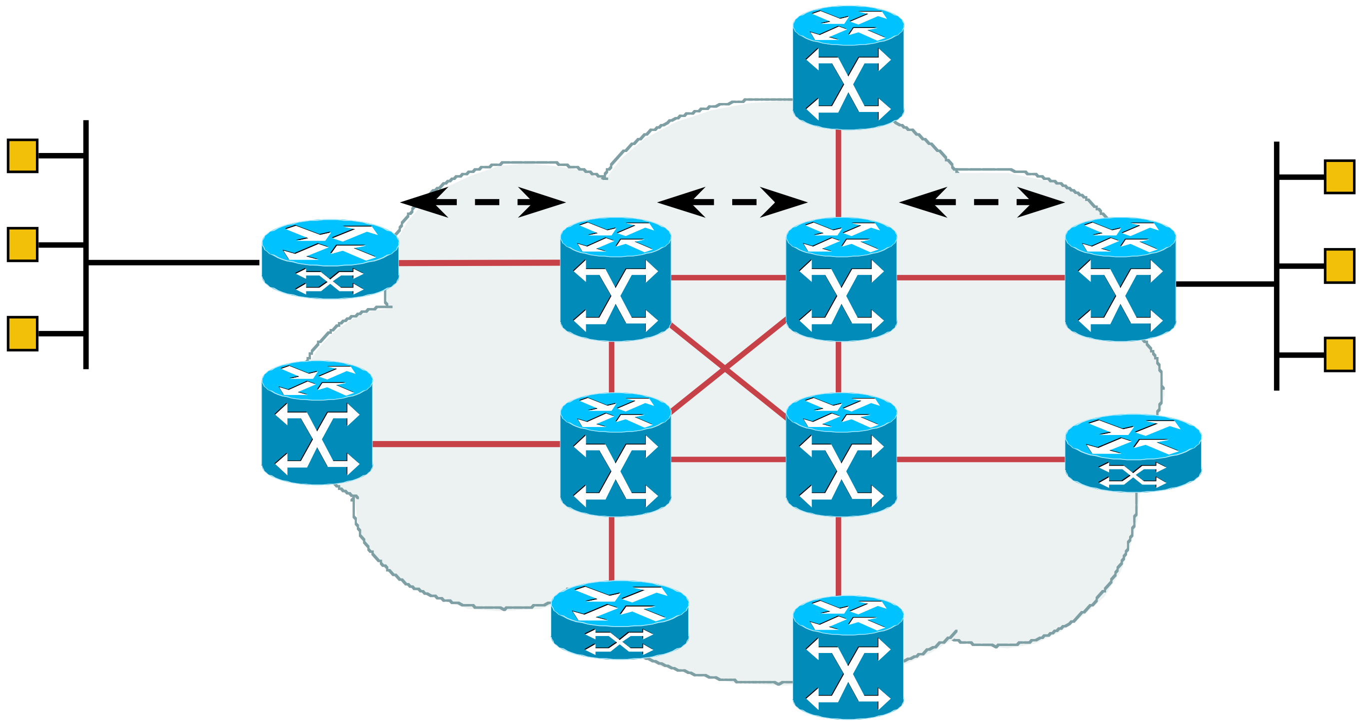
© 2003 Cisco Systems, Inc. All rights reserved.

MPLS: Forwarding



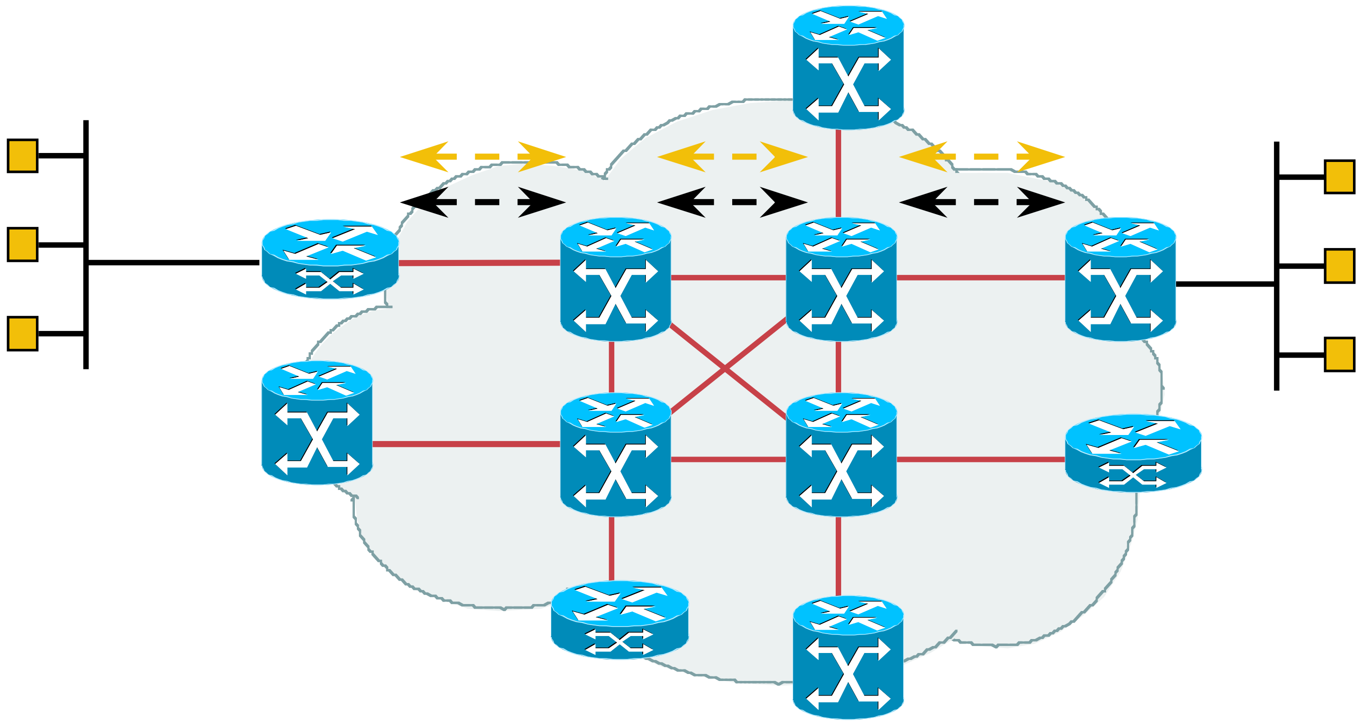
MPLS: Forwarding

Existing routing protocols (e.g. OSPF, IGRP) establish routes



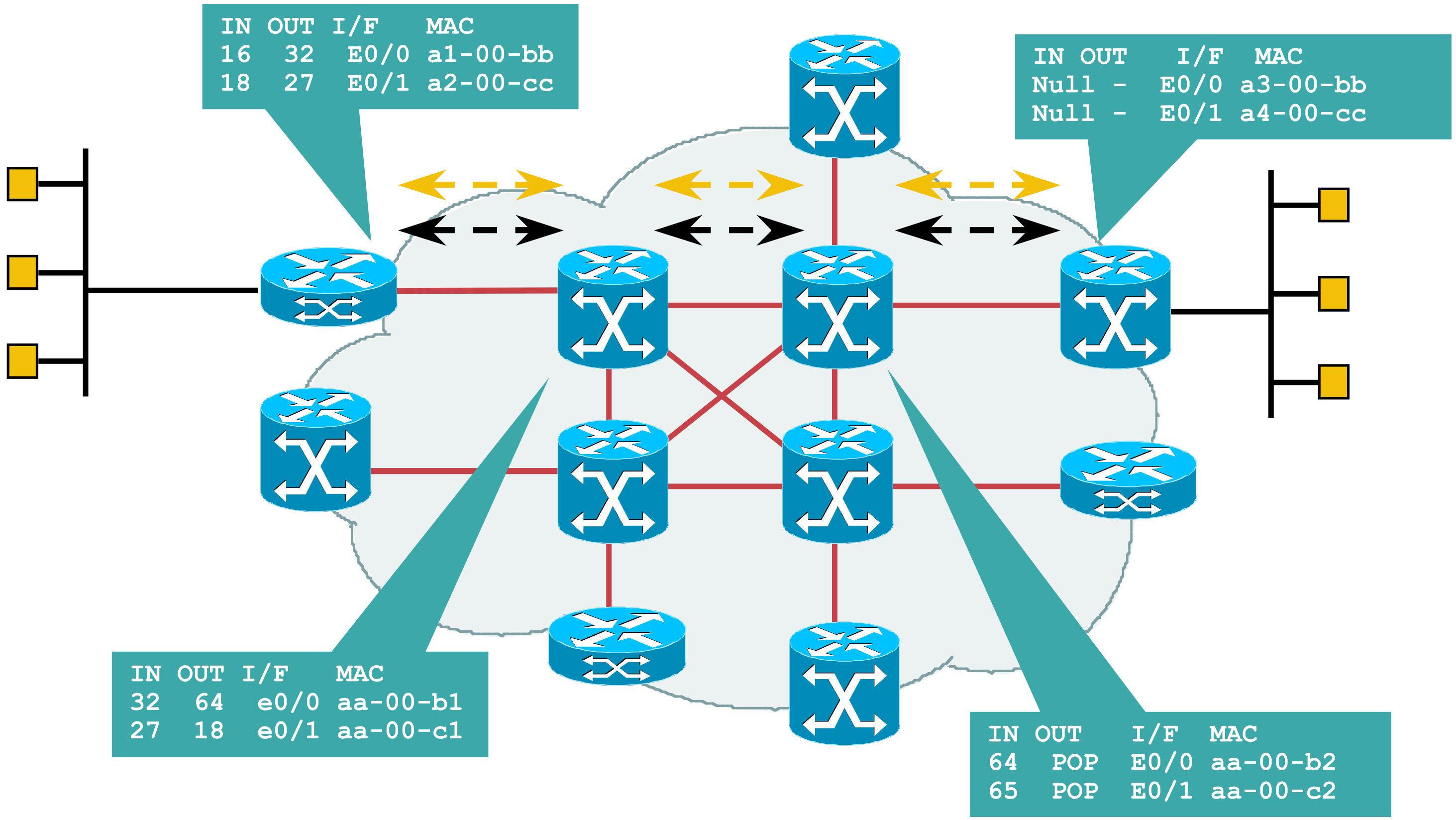
MPLS: Forwarding

Label Distribution Protocol (e.g., LDP) establishes label to routes mappings



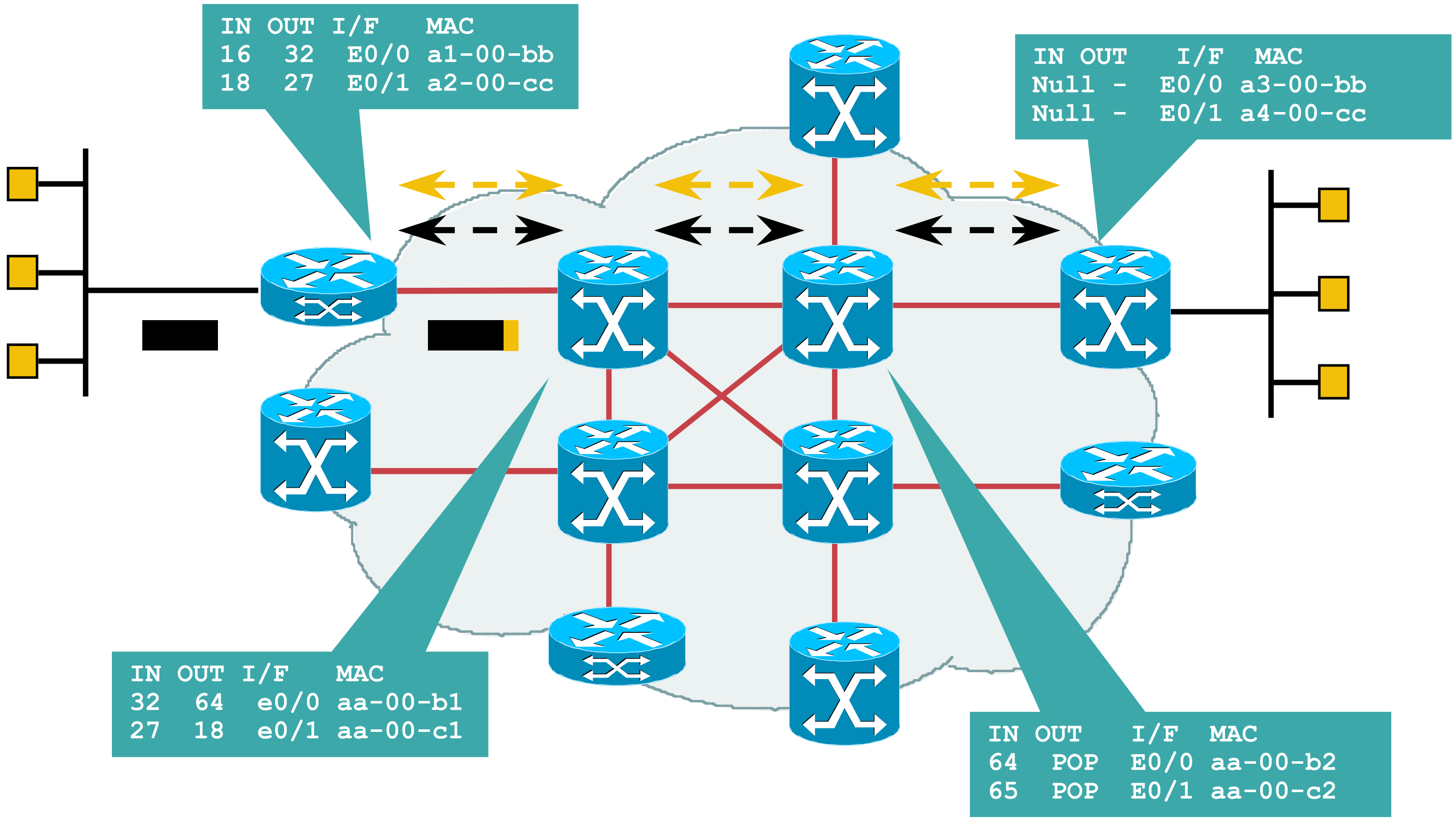
MPLS: Forwarding

Label Distribution Protocol (e.g., LDP) creates LFIB entries on LSRs



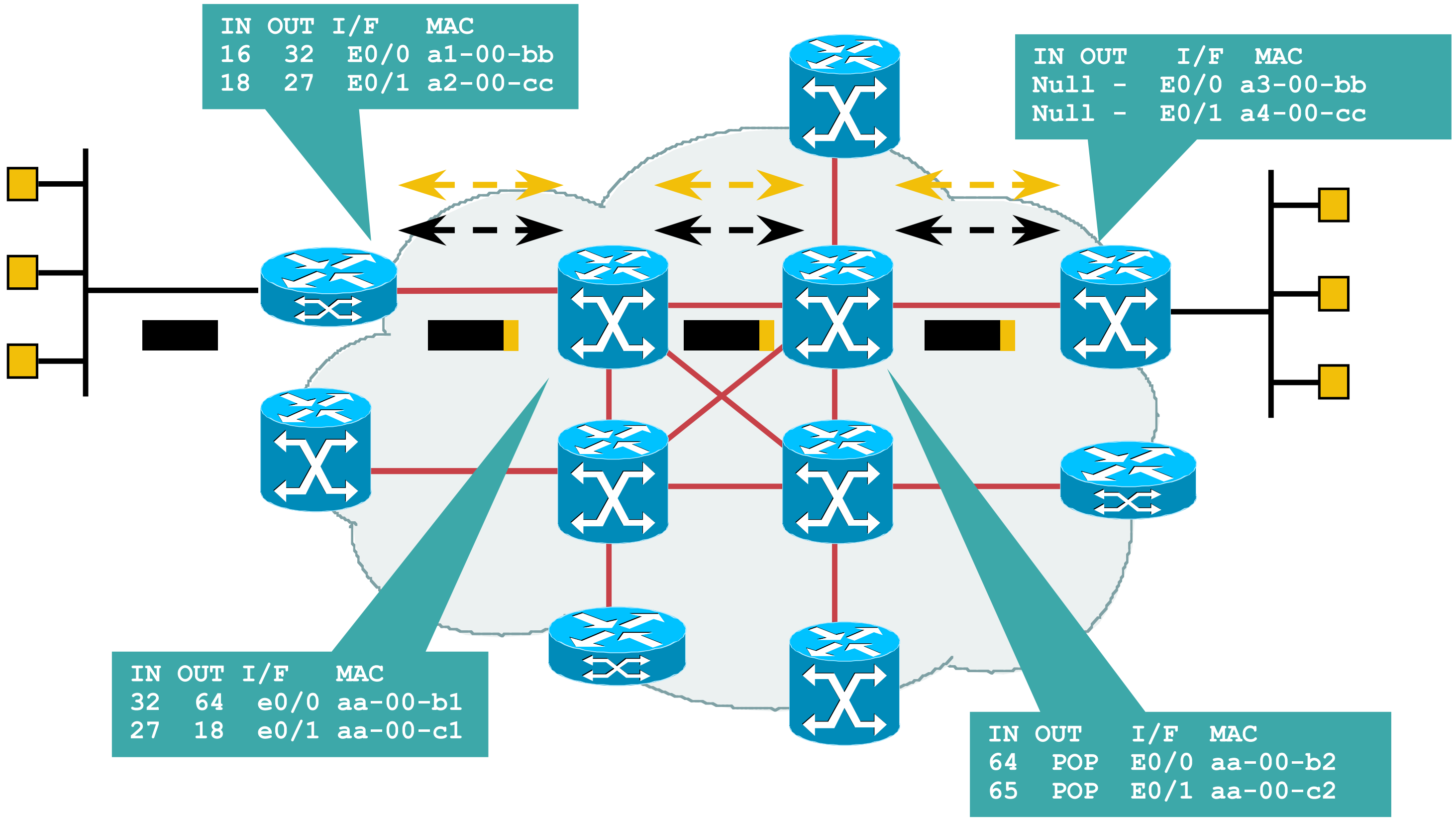
MPLS: Forwarding

Ingress edge LSR receives packet, performs Layer 3 value-added services, and “label” packets



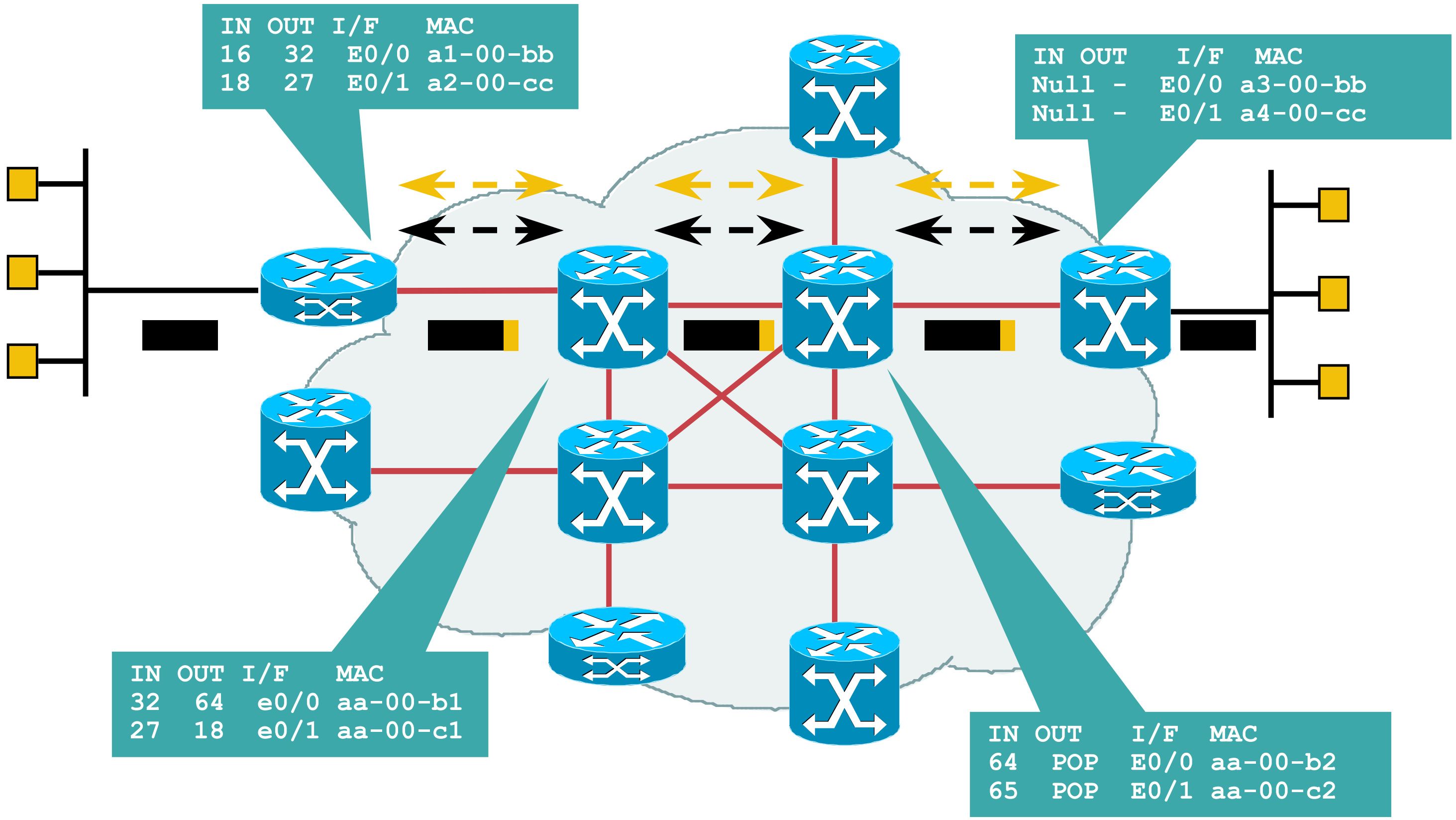
MPLS: Forwarding

LSRs forward labeled packets using label swapping



MPLS: Forwarding

Edge LSR at egress removes remaining label* and delivers packet



*** Pentulimate hop popping actually occurs. There may not necessarily be a label in the packet at the ultimate or egress LSR.**

Label Assignment and Label Distribution

Label Distribution Modes

- **Downstream unsolicited**

  Downstream node just advertises labels for prefixes/FEC reachable via that device

- **Downstream on-demand**

  Upstream node requests a label for a learnt prefix via the downstream node

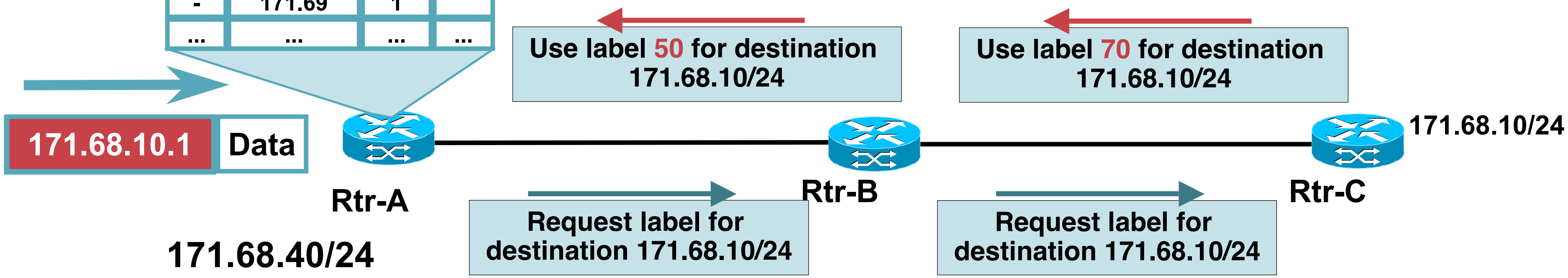
- **Several protocols for label Distribution**

  LDP - Maps unicast IP destinations into labels

  RSVP, CR-LDP - Used for traffic engineering and resource reservation

  BGP - External labels (VPN)

In label	Address Prefix	Out I'face	Out label
-	171.68.10.0	1	50
-	171.69	1	
...	...	...	...

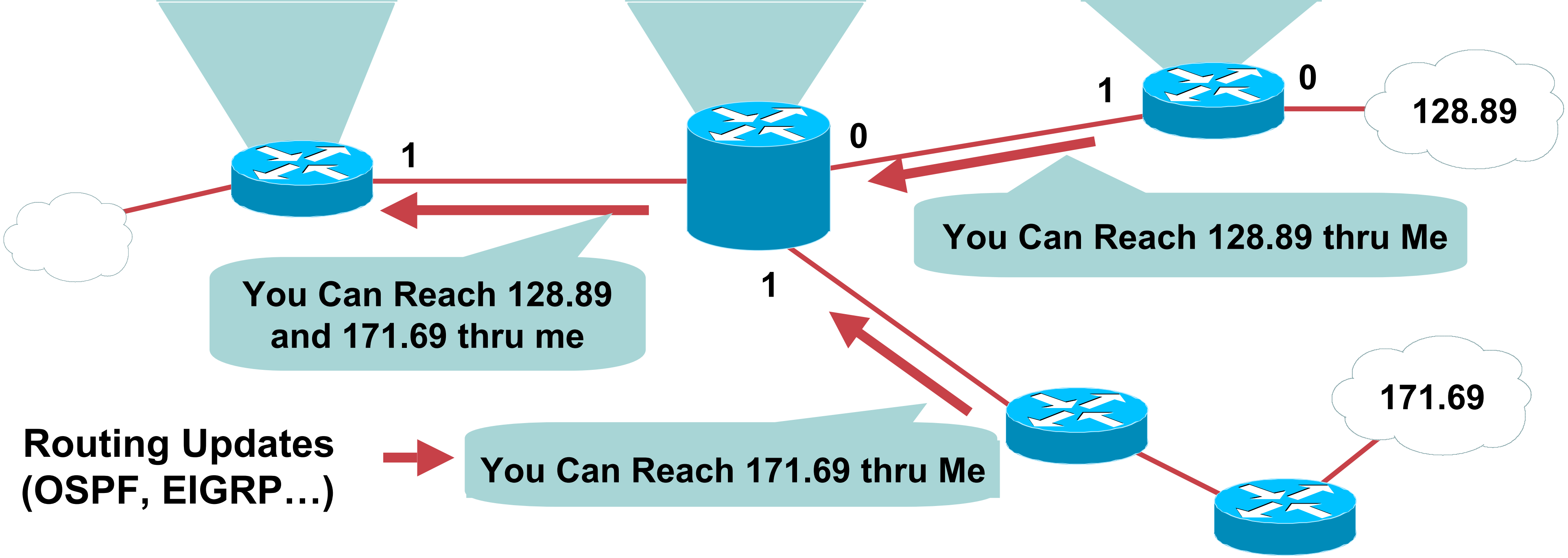


Traditional Routing
Route Distribution

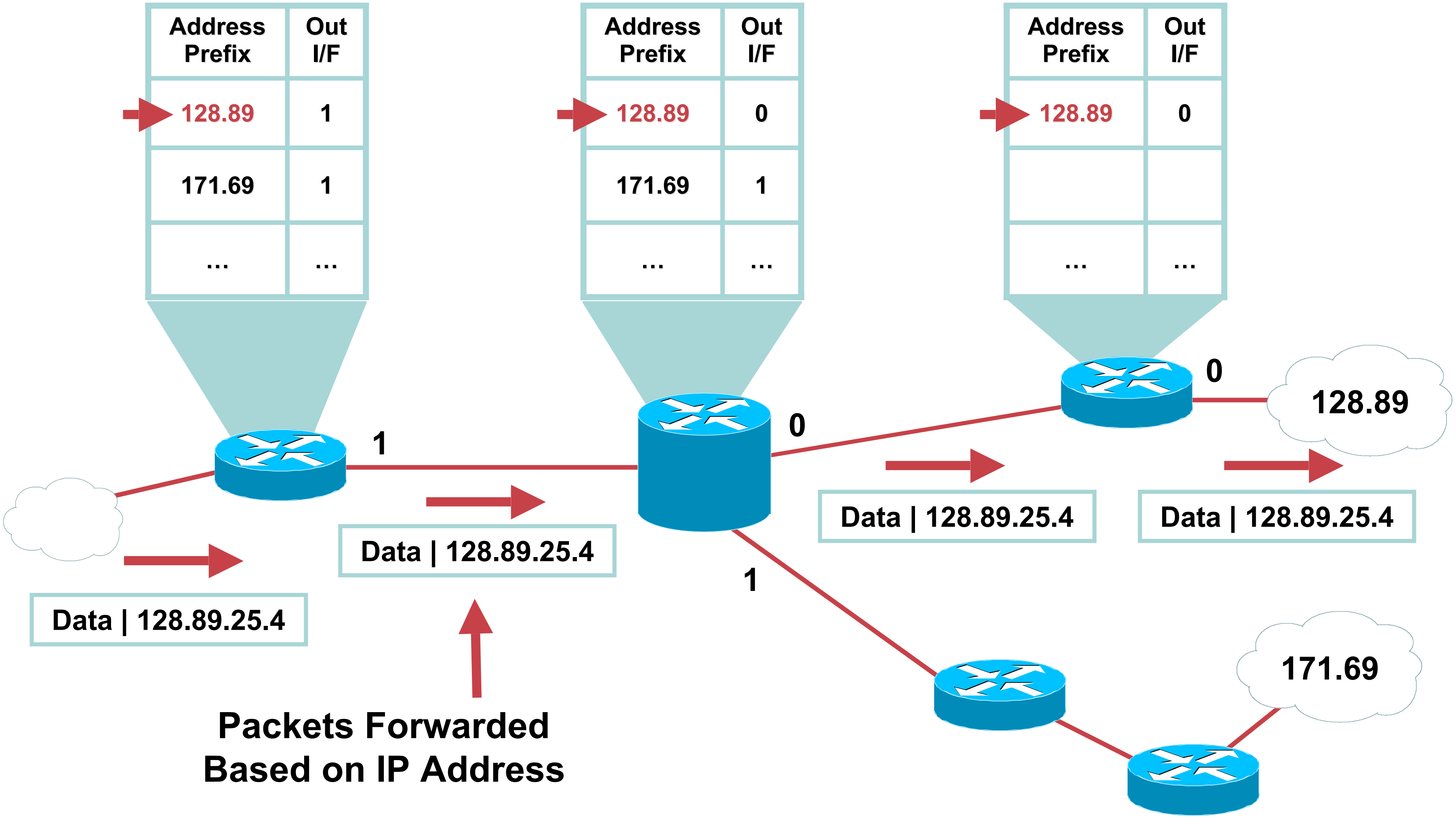
Address Prefix	Out I/F
128.89	1
171.69	1
…	…

Address Prefix	Out I/F
128.89	0
171.69	1
…	…

Address Prefix	Out I/F
128.89	0
…	…



Traditional Routing
Packet Routing

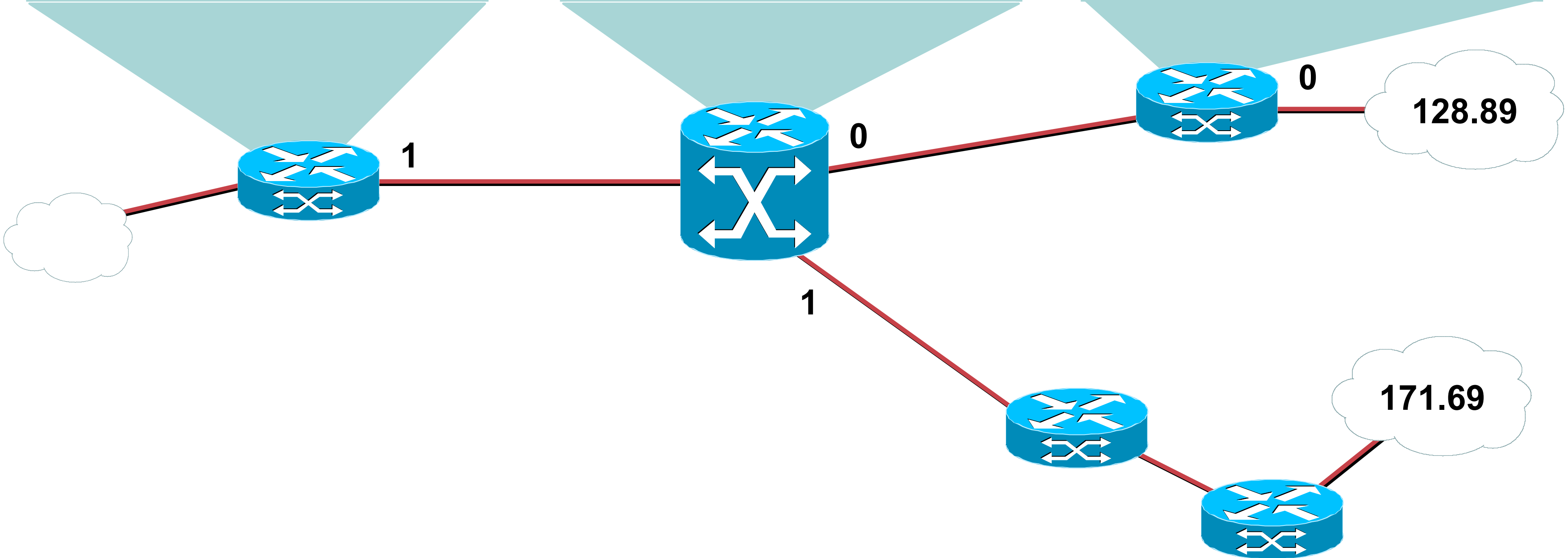


MPLS Forwarding
In/Out Label Fields

In Label	Address Prefix	Out I/F	Out Label
	128.89	1	
	171.69	1	
	...	...	

In Label	Address Prefix	Out I/F	Out Label
	128.89	0	
	171.69	1	
	...	...	

In Label	Address Prefix	Out I/F	Out Label
	128.89	0	
	...	...	

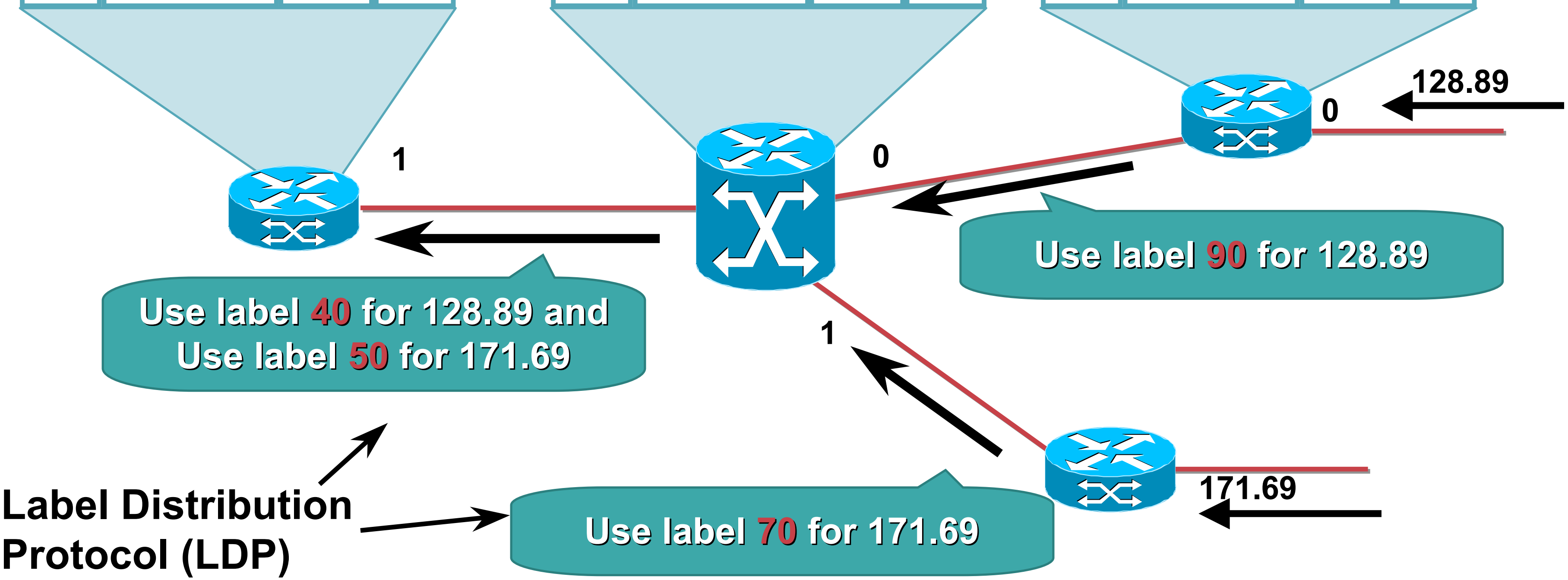


MPLS Example: Assigning and Distributing Labels

In label	Address Prefix	Out I'face	Out label
-	128.89	1	40
-	171.69	1	50
...	...	...	...

In label	Address Prefix	Out I'face	Out label
40	128.89	0	90
50	171.69	1	70
...	...	...	...

In label	Address Prefix	Out I'face	Out label
90	128.89	0	-
...	...	...	...



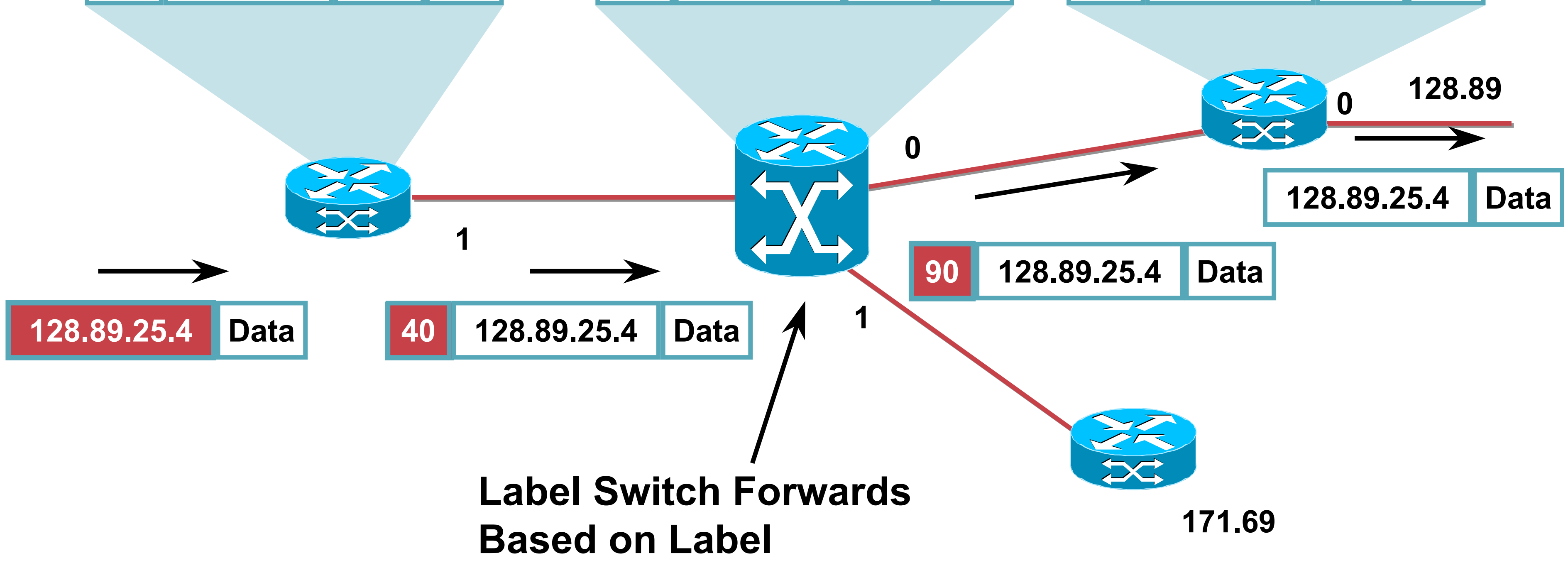
Label Distribution Protocol (LDP)
(Unsolicited Downstream Allocation)

MPLS Example: Forwarding Packets

In label	Address Prefix	Out I'face	Out label
-	128.89	1	40
-	171.69	1	50
...	...	...	...

In label	Address Prefix	Out I'face	Out label
40	128.89	0	90
50	171.69	1	70
...	...	...	...

In label	Address Prefix	Out I'face	Out label
90	128.89	0	-
...	...	...	...

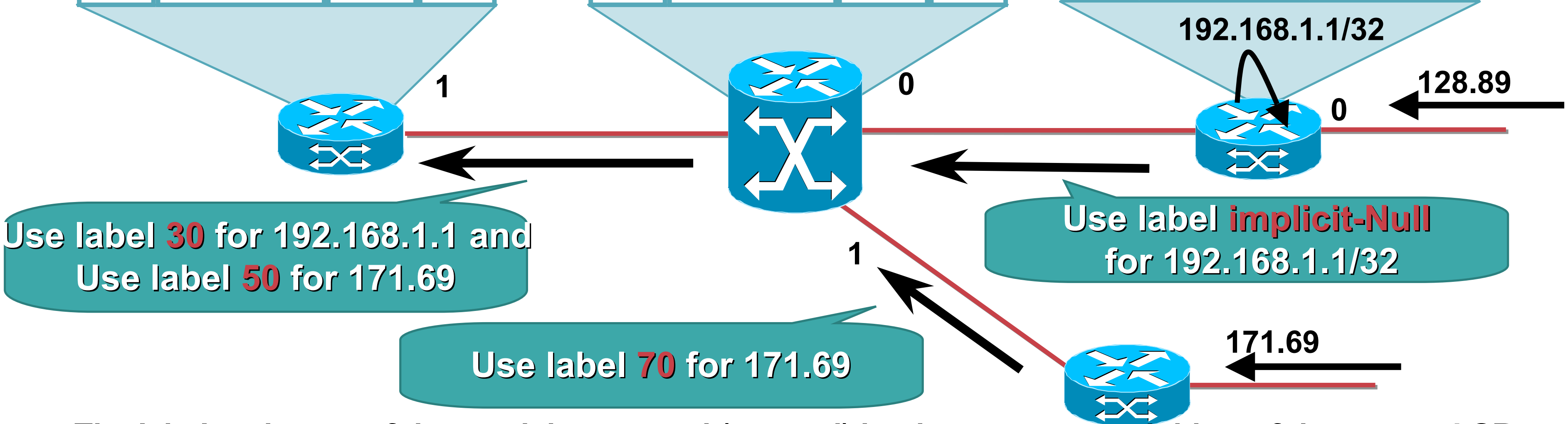


Penultimate Hop Popping

In label	Address Prefix	Out I'face	Out label
-	192.168.1.1	1	30
-	171.69	1	50
...	...	...	...

In label	Address Prefix	Out I'face	Out label
30	192.168.1.1	0	POP
50	171.69	1	70
...	...	...	...

In label	Address Prefix	Out I'face	Out label
Imp	192.168.1.1	0	-
...	...	...	...



- The label at the top of the stack is removed (popped) by the upstream neighbor of the egress LSR
- The egress LSR requests the "popping" through the label distribution protocol
  - Egress LSR advertises *implicit-null* label - Default on Cisco Routers
- One lookup is saved in the egress LSR
- Optionally *explicit-null* label (value = 0) can be advertised

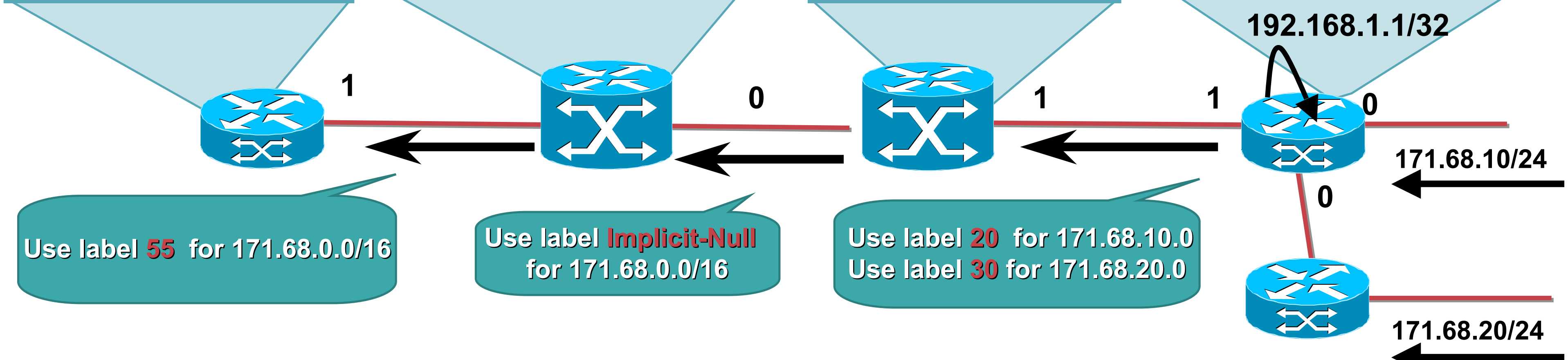
Aggregation and layer 3 summarisation

In label	Address Prefix	Out I'face	Out label
-	171.68.0.0	s1	55
...	...	...	...

In label	Address Prefix	Out I'face	Out label
55	171.68.0.0	s0	POP
-			
...	...	...	...

In label	Address Prefix	Out I'face	Out label
-	171.68.10.0	s1	20
-	171.68.20.0	s1	30
Imp	171.68.0.0	Null	...

In label	Address Prefix	Out I'face	Out label
20	171.68.10.0	e0	-
30	171.68.20.0	s1	
...	...	...	...



- The LSR which does summarisation will be the end node LSR of all LSPs related to the summary address
  - Aggregation point
- The LSR will have to examine the second level label of each packet
  - If no second label, the LSR has to examine the IP header and can lead to blackholing of traffic
  - No summarisation in ATM-LSRs

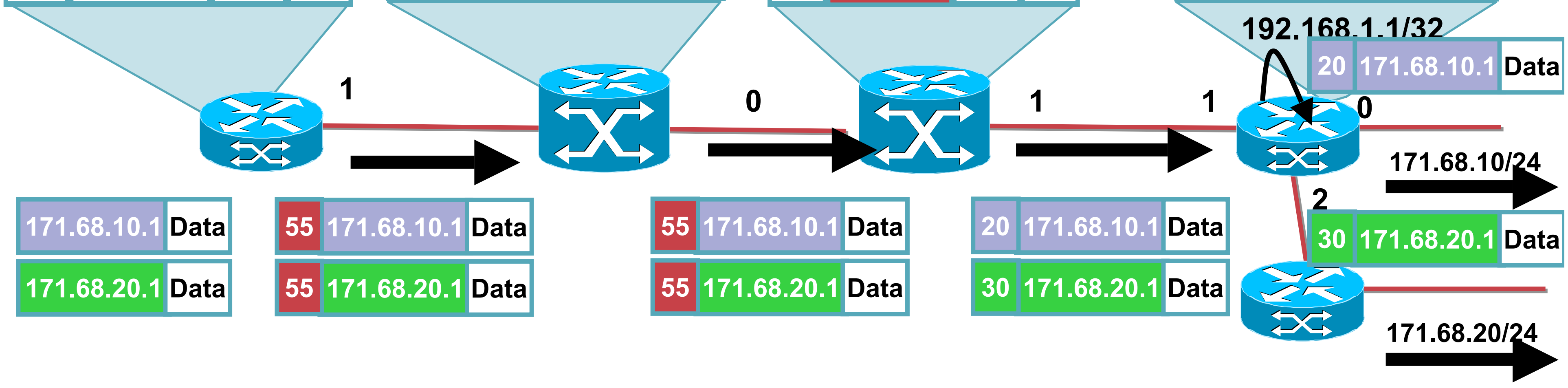
Aggregation and layer 3 summarisation (Packet Forwarding)

In label	Address Prefix	Out I'face	Out label
-	171.68.0.0	s1	55
...	...	...	...

In label	Address Prefix	Out I'face	Out label
55	171.68.0.0	s0	POP
-			
...	...	...	...

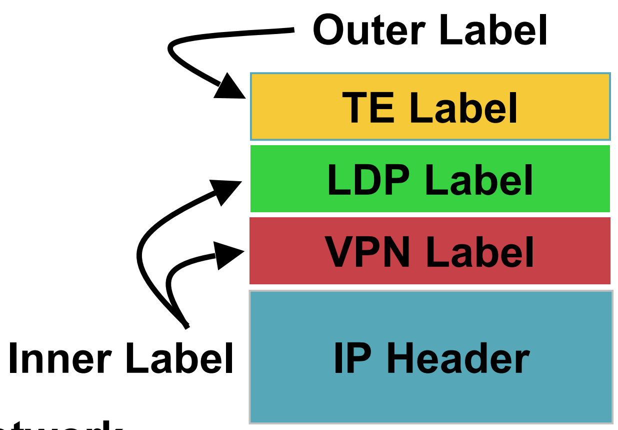
In label	Address Prefix	Out I'face	Out label
-	171.68.10.0	s1	20
-	171.68.20.0	s1	30
-	171.68.0.0	Null	...

In label	Address Prefix	Out I'face	Out label
20	171.68.10.0	e0	-
30	71.68.20.0	s2	
...	...	...	...



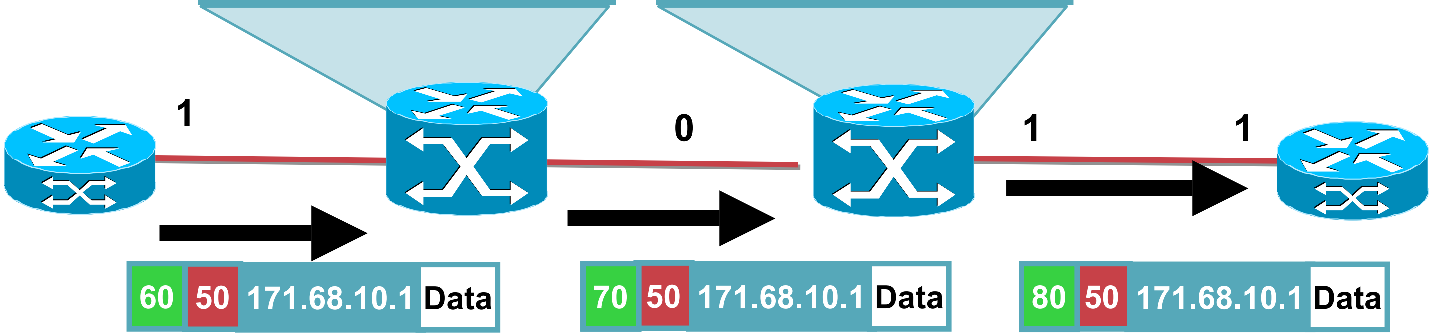
Label Stacking

- There may be more than one label in an MPLS packet
- Allows building services such as
  - MPLS VPNs
  - Traffic Engineering and Fast Re-route
  - VPNs over Traffic Engineered core
  - Any Transport over MPLS
- Outer label used to route/switch the MPLS packets in the network

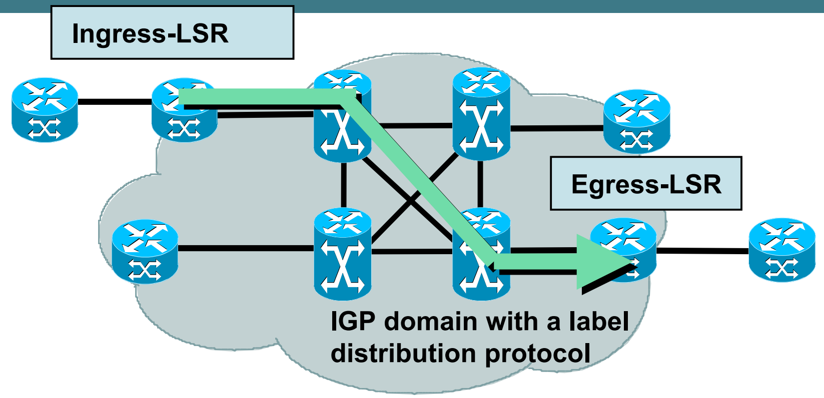


In label	Address Prefix	Out I'face	Out label
60	171.68.10.0	s0	70
-			
...	...	...	...

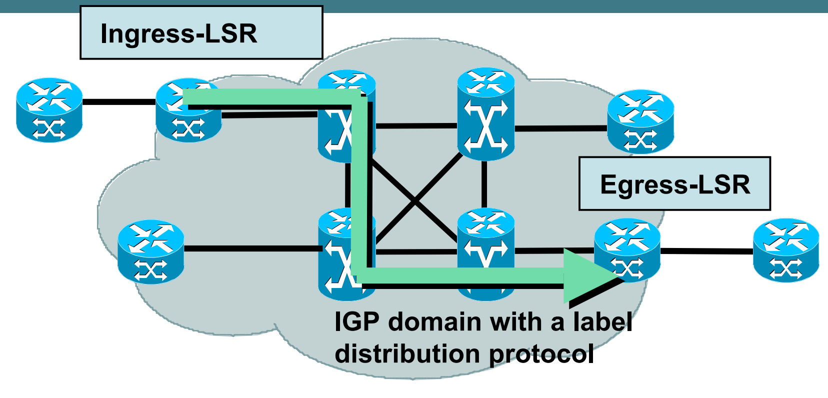
In label	Address Prefix	Out I'face	Out label
70	171.68.10.0	s0	80
-			
...	...	...	...



Label Switch Path (LSP)



LSP follows IGP shortest path



LSP diverges from IGP shortest path

- **FEC is determined in LSR-ingress**
- **LSPs derive from IGP routing information**
- **LSPs may diverge from IGP shortest path**
  - **LSP tunnels (explicit routing) with Traffic Engineering**

Basic Application Hierarchical Routing

© 2003 Cisco Systems, Inc. All rights reserved.

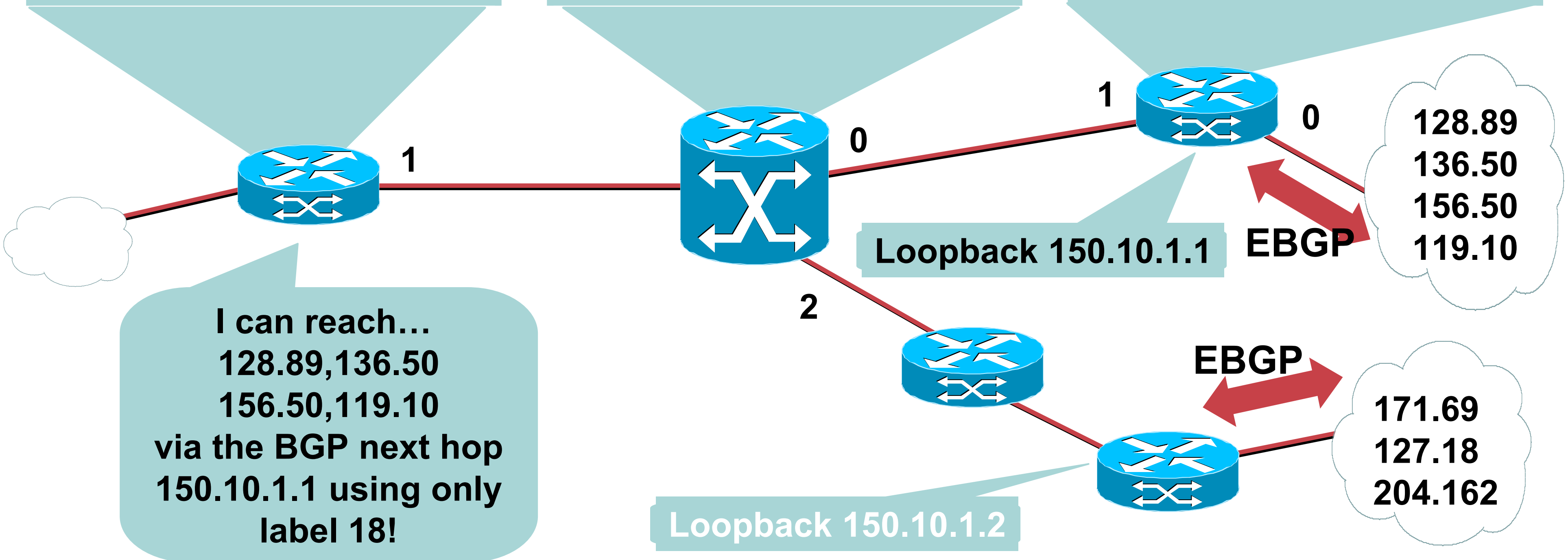
44

Internet Scalability

In Label	Address Prefix	Out I/F	Out Label
-	150.10.1.1	1	18
	150.10.1.2	1	17
	...	...	

In Label	Address Prefix	Out I/F	Out Label
18	150.10.1.1	0	Pop
17	150.10.1.2	2	22
	...	...	

In Label	Address Prefix	Out I/F	Out Label
Pop	150.10.1.1	-	-
	...	...	



Basic Application
Cell Based MPLS (IP+ATM)

© 2003 Cisco Systems, Inc. All rights reserved.

MPLS and ATM

- **Label Switching Steps:**
  - **Make forwarding decision using fixed-length Label**
  - **Rewrite label with new value**
    - **Similar to ATM cell switching**
- **Key differences:**
  - **Label set up: LDP vs ATM Forum Signaling**
  - **Label granularity: Per-prefix**

MPLS and ATM

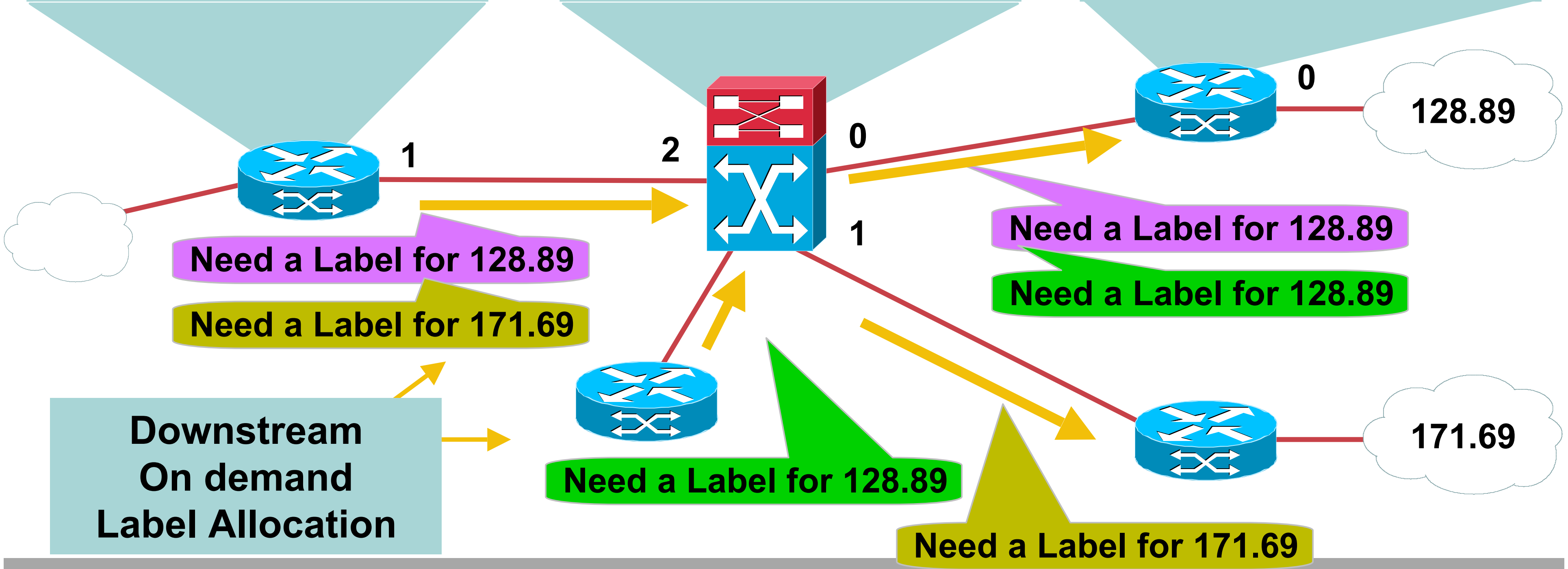
- **Common forwarding paradigm**
  - label swapping = ATM switching
- **Use ATM user plane**
  - use VPI/VCI for labels
  - Label is applied to each cell, not whole packet
- **Replace ATM Forum control plane with the MPLS control component:**
  - Network Layer routing protocols (e.g., OSPF, BGP, PIM) + Label Distribution Protocol (e.g., LDP)

Cell Based MPLS - Assigning Labels

In Label	Address Prefix	Out I/F	Out Label
-	128.89	1	40
-	171.69	1	50
	...	...	

In Label	Address Prefix	Out I/F	Out Label
40	128.89	0	90
80	128.89	0	100
50	171.69	1	70

In Label	Address Prefix	Out I/F	Out Label
90	128.89	0	-
100	128.89	0	-
	...	...	

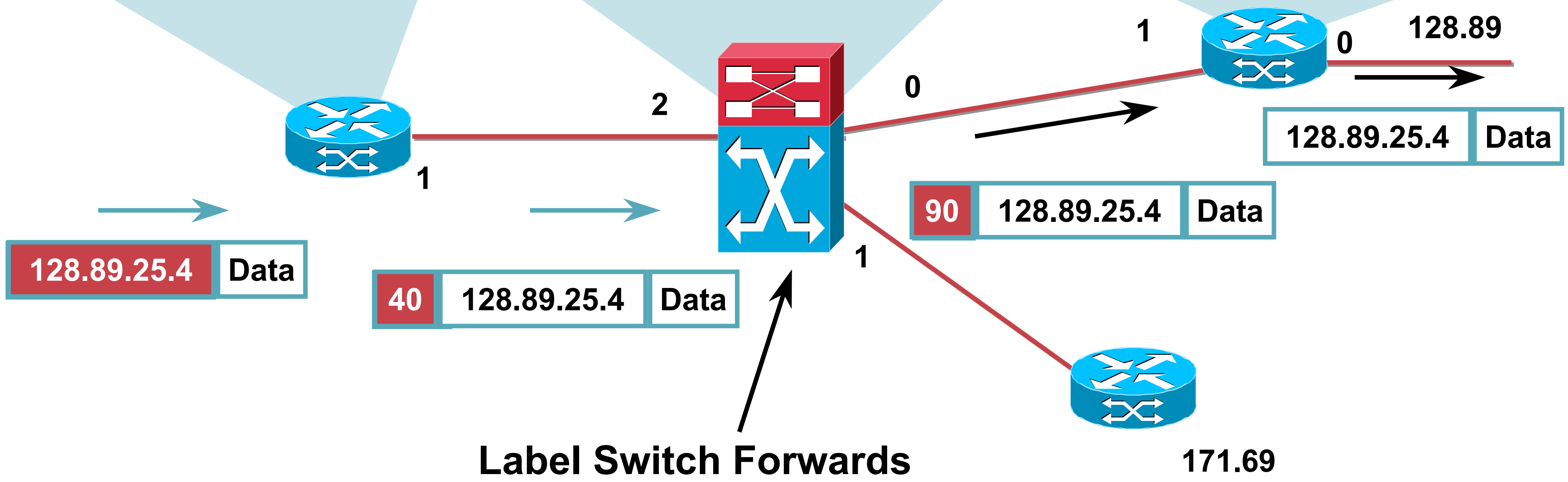


ATM Cell Based MPLS Example: Packet Forwarding

In label	Address Prefix	Out I'face	Out label
-	128.89	1	40
-	171.69	1	50
	...	...	

In label	In I/F	Address Prefix	Out I'face	Out label
40	2	128.89	0	90
80	3	128.89	0	100
50	2	171.69	1	70

In label	In I/F	Address Prefix	Out I'face	Out label
90	1	128.89	0	-
100	1	128.89	0	-
		...	...	



Label Switch Forwards Based on Label

Summary and Benefits

Summary

- **MPLS allows flexible packet classification and network resources optimisation**
- **Labels are distributed by different protocols**

  LDP, RSVP, BGP
- **Different distribution protocols may co-exist in the same LSR**
- **Labels have local (LSR) significance**

  No need for global (domain) wide label allocation/numbering

Benefits of MPLS

- De-couples IP packet forwarding from the information carried in the IP header of the packet
- Provides multiple routing paradigms (e.g., destination-based, explicit routing, VPN, multicast, CoS, etc…) over a common forwarding algorithm (label swapping)
- Facilitates integration of ATM and IP - from control plane point of view an MPLS-capable ATM switch looks like a router

MPLS VPN Overview

Agenda

- **VPN Concepts**
- **Terminology**
- **VPN Connection model**
- **Forwarding Example**

VPN Concepts

What is an MPLS-VPN?

- An IP network infrastructure delivering private network services over a public infrastructure
  - Use a layer 3 backbone
  - Scalability, easy provisioning
  - Global as well as non-unique private address space
  - QoS
  - Controlled access
  - Easy configuration for customers

VPN Models

- **There are two basic types of design models that deliver VPN functionality**
  - **Overlay Model**
  - **Peer Model**

The Overlay model

- **Private trunks over a TELCO/SP shared infrastructure**
  - **Leased/Dialup lines**
  - **FR/ATM circuits**
  - **IP (GRE) tunnelling**
- **Transparency between provider and customer networks**
- **Optimal routing requires full mesh over over backbone**

The Peer model

- Both provider and customer network use same network protocol and control plane
- CE and PE routers have routing adjacency at each site
- All provider routers hold the full routing information about all customer networks
- Private addresses are not allowed
- May use the virtual router capability

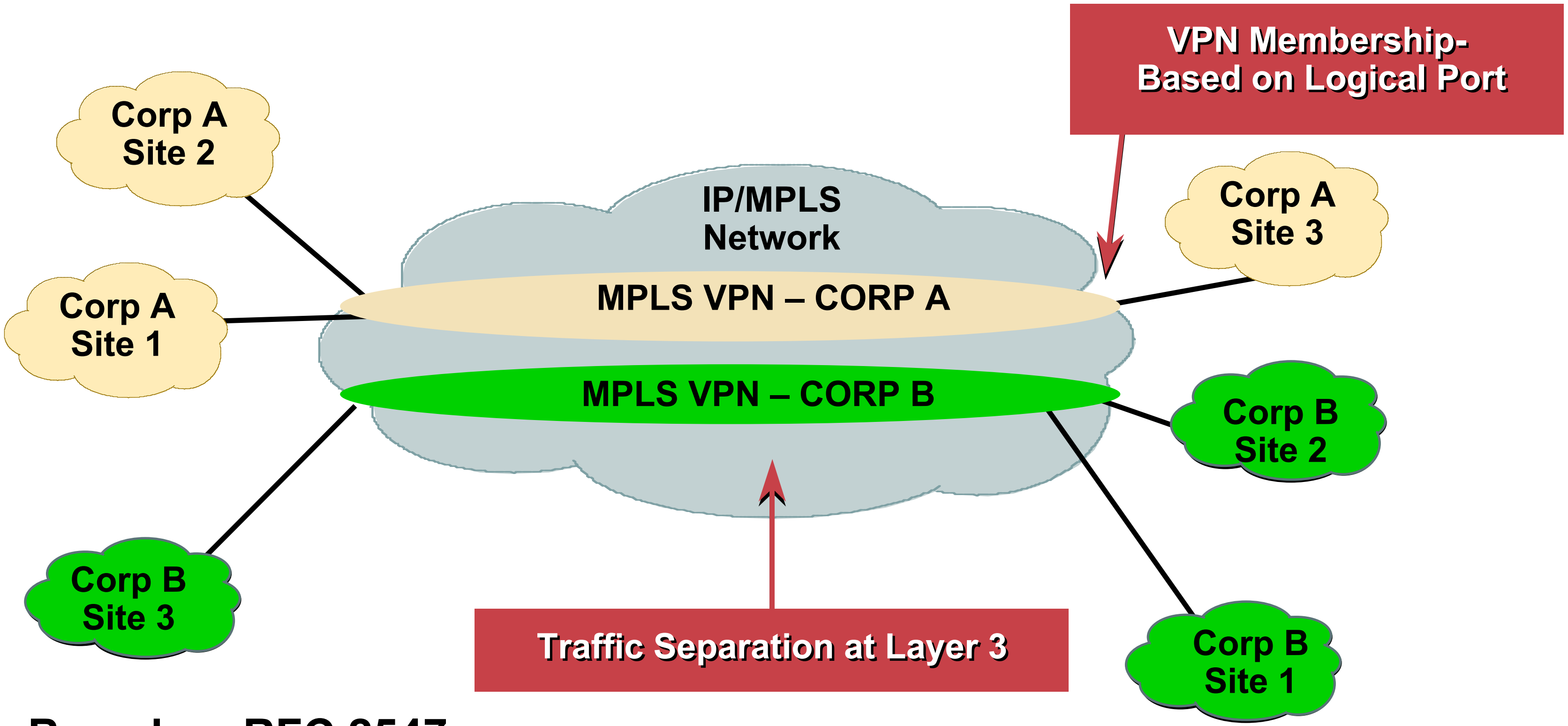
  Multiple routing and forwarding tables based on Customer Networks

MPLS-VPN = True Peer model

- **MPLS-VPN is similar in operation to peer model**
- **Provider Edge routers receive and hold routing information only about VPNs directly connected**
- **Reduces the amount of routing information a PE router will store**
- **Routing information is proportional to the number of VPNs a router is attached to**
- **MPLS is used within the backbone to switch packets (no need of full routing)**

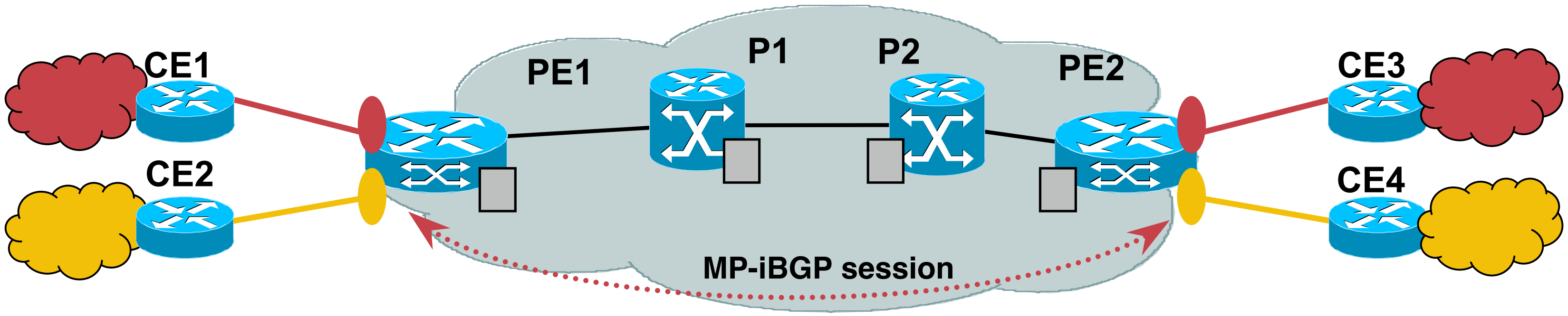
MPLS VPN Connection Model

MPLS-VPN Overview



- Based on RFC 2547
- Provide Any-to-Any connectivity at layer3 in a scalable manner.
- Only PE routers hold routes for attached VPNs
- Allows overlapping IP addresses between different VPNs
- MPLS for forwarding through service provider core.

MPLS VPN Connection Model



PE Routers

- Maintain separate Routing tables per VPN customer and one for Global routing
- Use MPLS with P routers
- Uses IP with CE routers
- Connects to both CE and P routers
- Distribute VPN information through MP-BGP to other PE router with VPN-IPv4 addresses, extended community, label

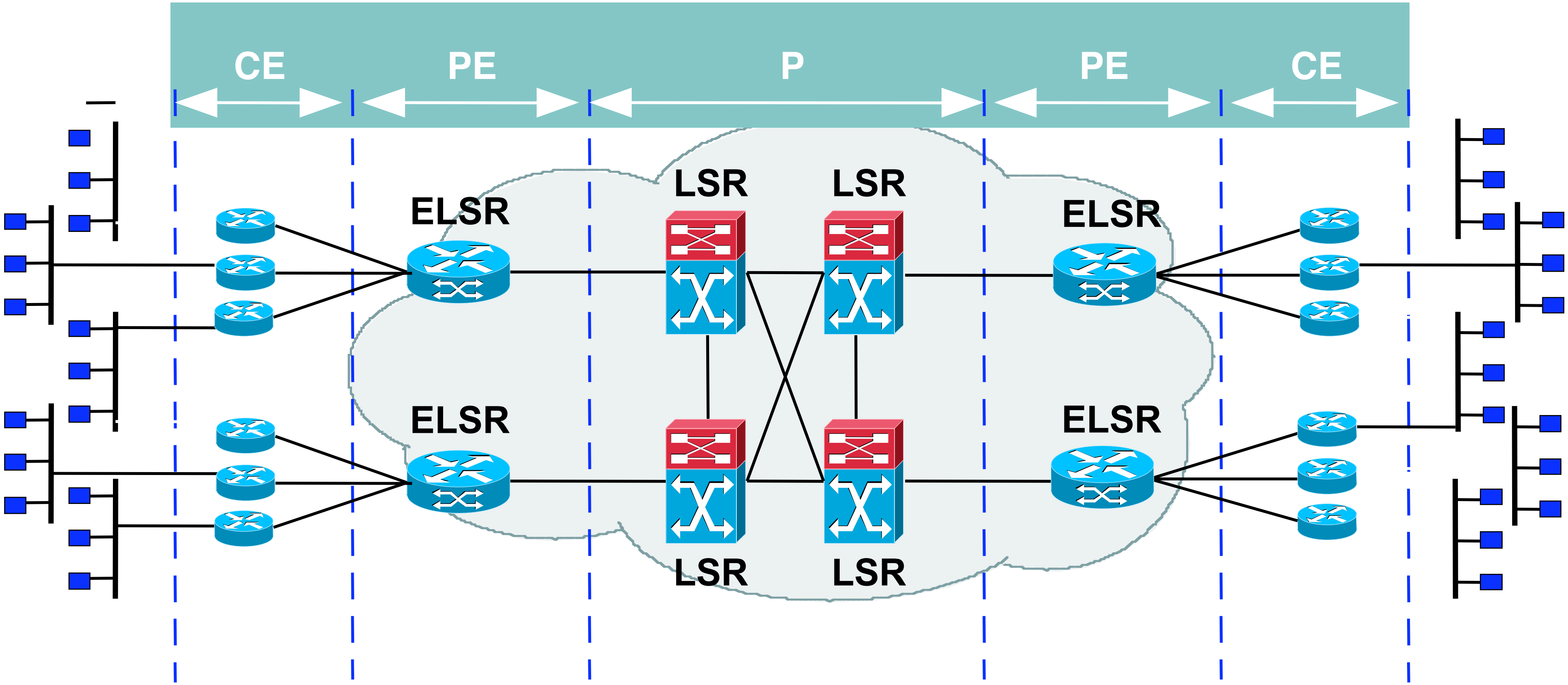
P Routers

- P routers are in the core of the MPLS cloud
- P routers do not need to run BGP and doesn't need to have any VPN knowledge
- Forward packets by looking at labels
- P and PE routers share a common IGP

MPLS VPN Connection Model

- A VPN is a collection of sites sharing a common routing information (routing table)
- A site can be part of different VPNs
- A VPN has to be seen as a community of interest (or Closed User Group)
- Multiple Routing/Forwarding instances (VRF) on PE

MPLS VPN Components



C Network
(Customer Control)

P Network
(Provider Control)

C Network
(Customer Control)

VPN Components

- **PE-CE Routing**
- **VRF Tables**

  Hold customer routes at PE

- **MP-BGP**
- **Route-Distinguisher**

  Allows MP-BGP to distinguish between identical customer routes that are in different VPNs

- **Route-Targets**

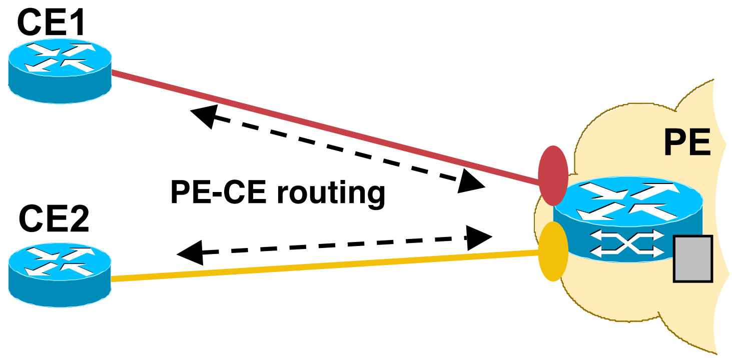
  Used to import and export routes between different VRF tables (creates Intranets and Extranets)

- **Route-maps**

  Allows finer granularity and control of importing exporting routes between VRFs instead of just using route-target

PE–CE Routing

PE-CE Routing

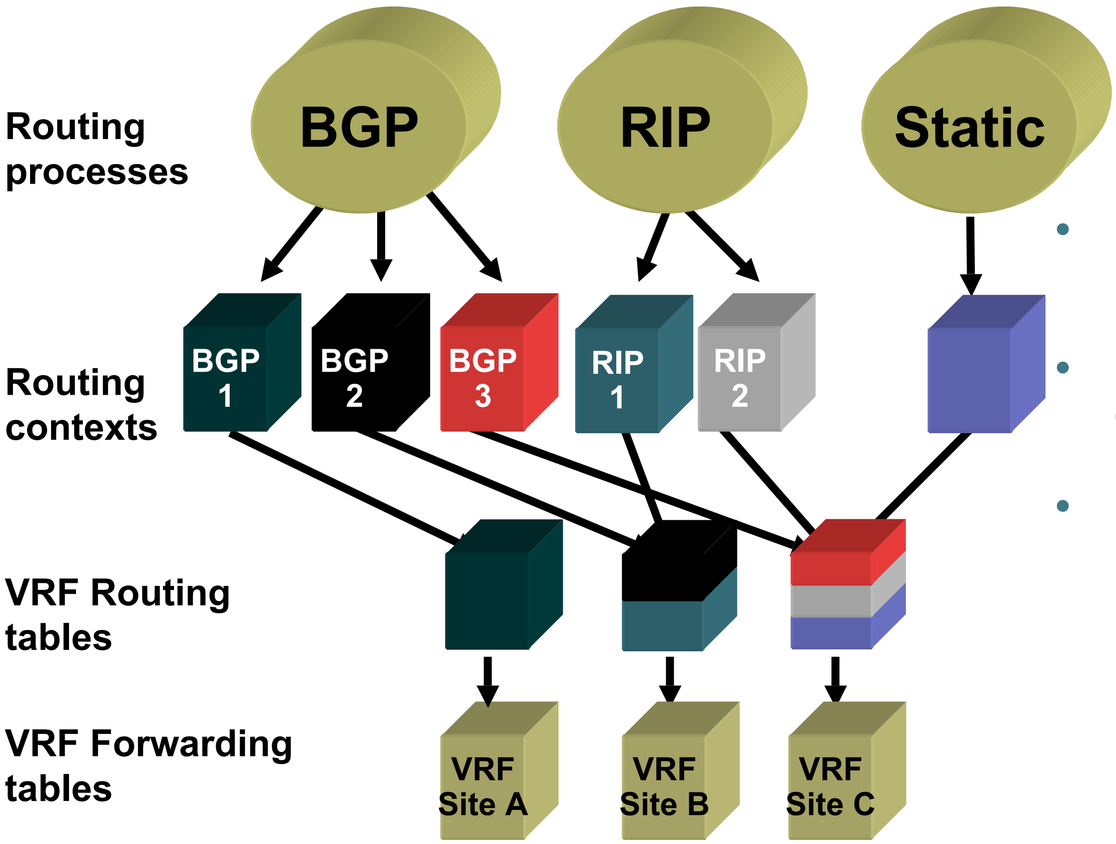


- PE and CE routers exchange routing information through eBGP, Static, OSPF, ISIS, RIP, EIGRP
- The CE router runs standard routing software, not aware it is connected to a VPN network

PE-CE routing protocols

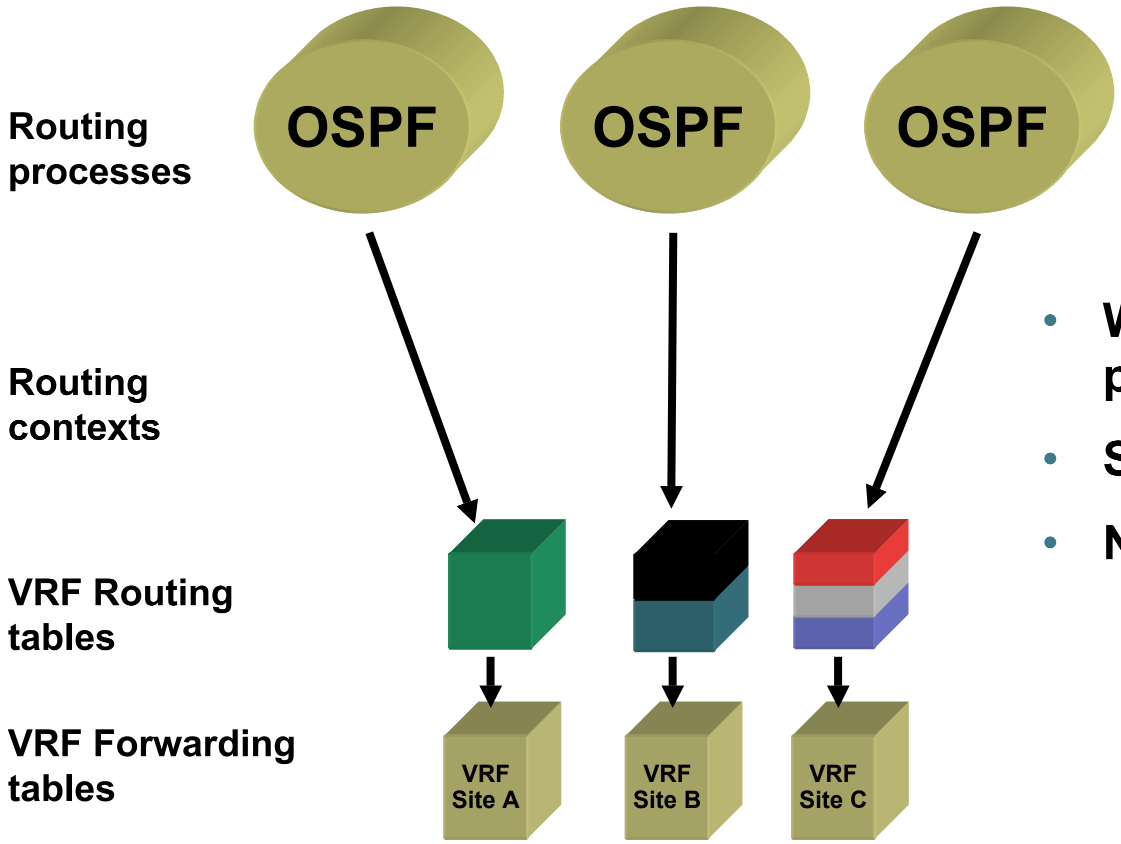
- Static/BGP are the most scalable
    - Single PE router can support 100s or 1000s of CE routers
- BGP is the most flexible
    - Particularly for multi-homing but not popular with Enterprise
    - Very useful if Enterprise requires Internet routes
- Use the others to meet customer requirements
    - OSPF popular with Enterprises – but sucks up processes
    - EIGRP not popular with Service Providers (Cisco proprietary)
    - IS-IS less prevalent in Enterprise environments
    - RIPv2 provides very simple functionality

Routing Protocol Contexts



- Routing processes run within specific routing contexts
- Populate specific VPN routing table and FIBs (VRF)
- Interfaces are assigned to VRFs

OSPF and Single Routing Instances



- With OSPF there is a single process per VRF
- Same for IS-IS
- No routing contexts

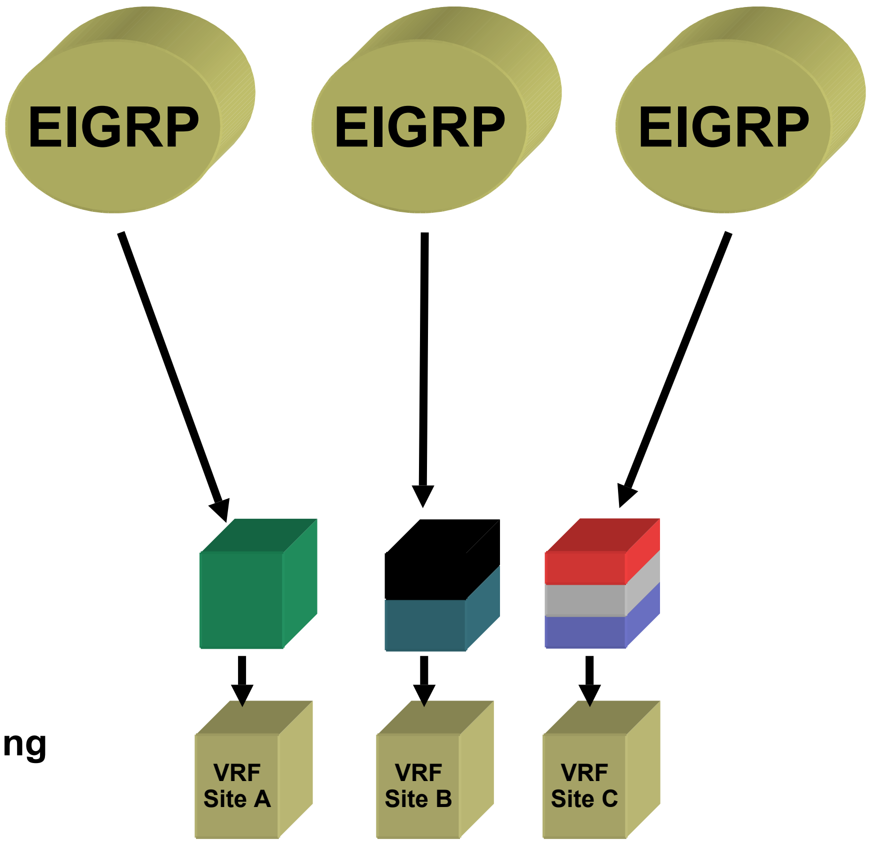
EIGRP PE-CE Routing

Routing processes

Routing contexts

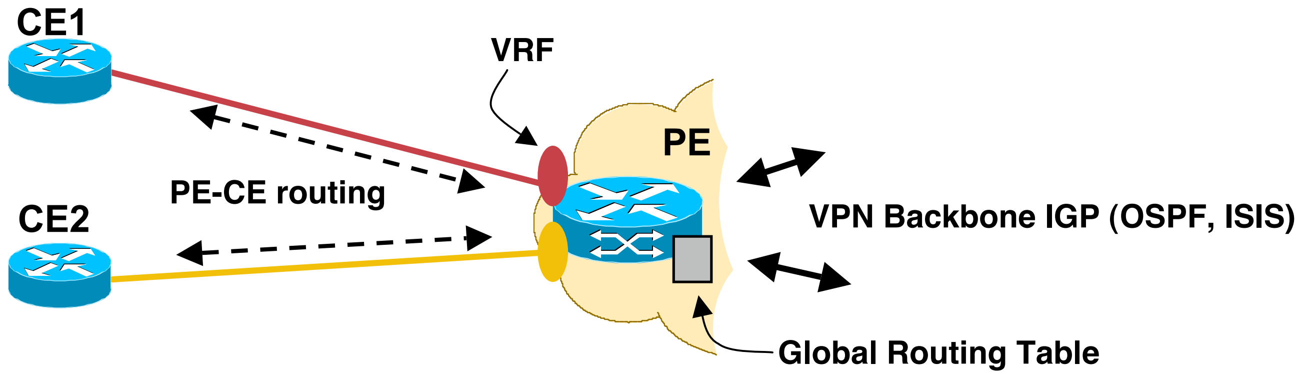
VRF Routing tables

VRF Forwarding tables



Routing Tables

Routing Tables



- PE routers maintain separate routing tables
- Global Routing Table

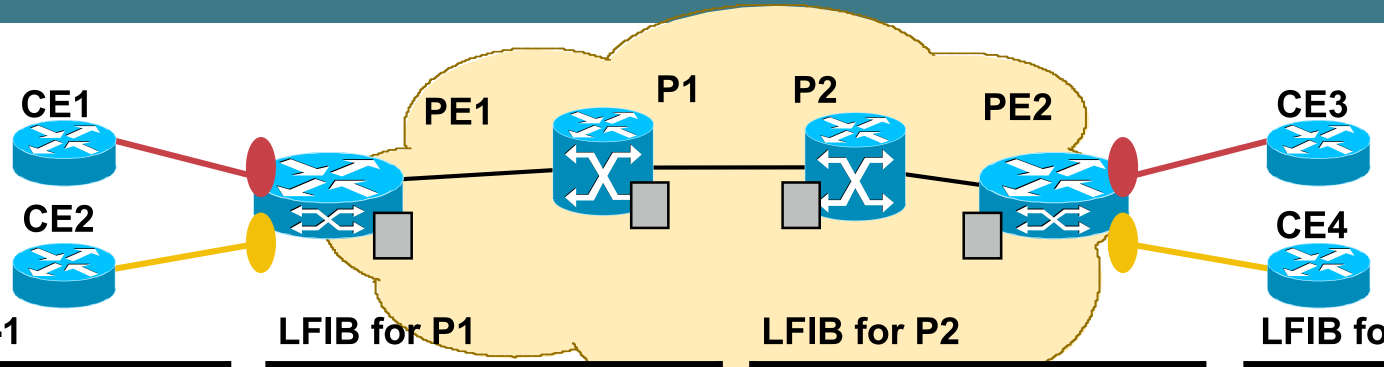
  All the PE and P routes populated by the VPN backbone IGP (ISIS or OSPF)
- VPN Routing and Forwarding Tables (VRF)

  Routing and Forwarding table associated with one or more directly connected sites (CEs)

  VRF are associated to (sub/virtual/tunnel) interfaces

  Interfaces may share the same VRF if the connected sites may share the same routing information

IGP and label distribution in the backbone



LFIB for PE-1

Des	Next Hop	IN	OUT
PE2	P1	17	50
P2	P1	18	65
P1	S0/0	19	POP

LFIB for P1

Des	Next Hop	IN	OUT
PE2	P2	50	34
P2	E0/2	65	POP
PE1	S3/0	67	POP

LFIB for P2

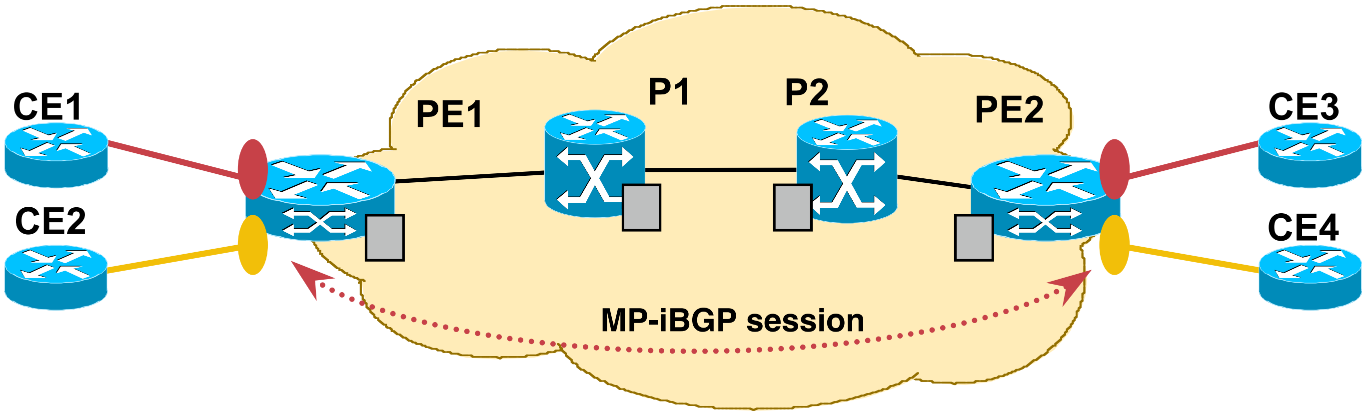
Des	Next Hop	IN	OUT
PE2	P1	34	POP
P1	E0/1	38	POP
PE1	P1	39	67

LFIB for PE2

Des	Next Hop	IN	OUT
P1	P2	44	38
P2	P2	36	65
PE1	P2	18	39

- All routers (P and PE) run an IGP and label distribution protocol
- Each P and PE router has routes for the backbone nodes and a label is associated to each route
- MPLS forwarding is used within the core

VPN Routing and Forwarding Table



- Multiple routing tables (VRFs) are used on PEs
- Each VRF contains customer routes
- Customer addresses can overlap
- VPNs are isolated
- Multi-Protocol BGP (MP-BGP) is used to propagate these addresses + labels between PE routers only

Multi-Protocol BGP

- **Propagates VPN routing information**

  Customer routes held in VPN Routing and Forwarding tables (VRFs)

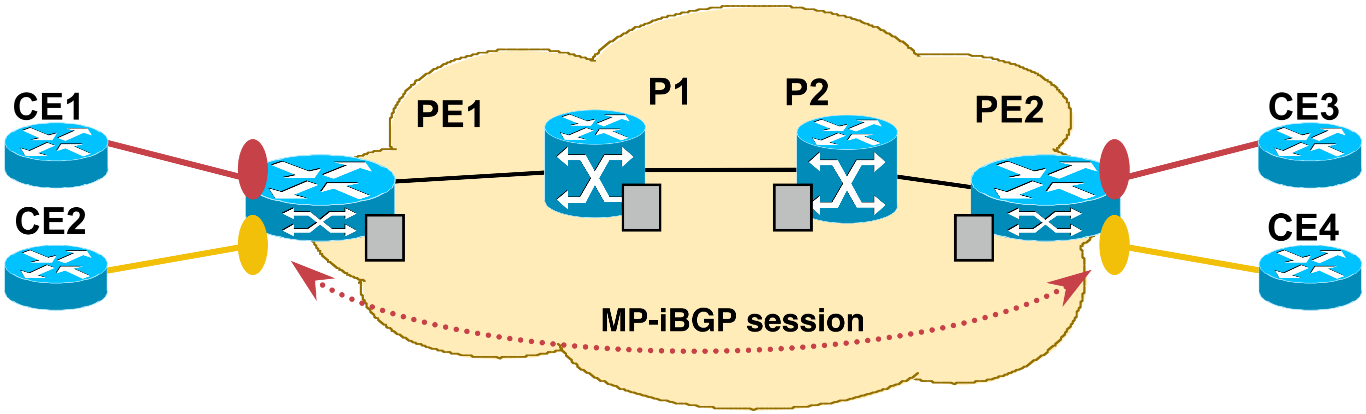
- **Only runs on Provider Edge**

  P routers are not aware of VPN’s only labels

- **PEs are fully meshed**

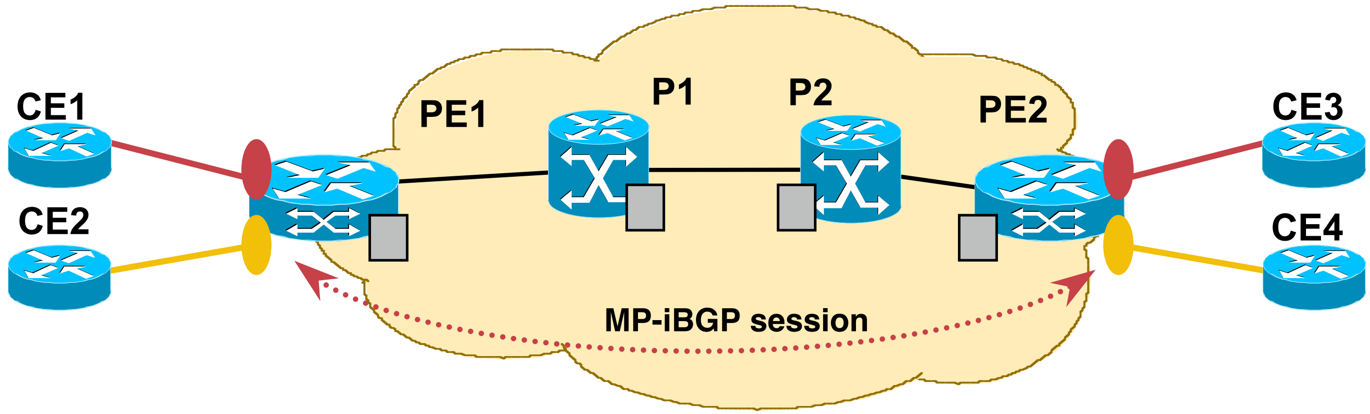
  Using Route Reflectors or direct peerings between PE routers

MPLS VPN Requirements



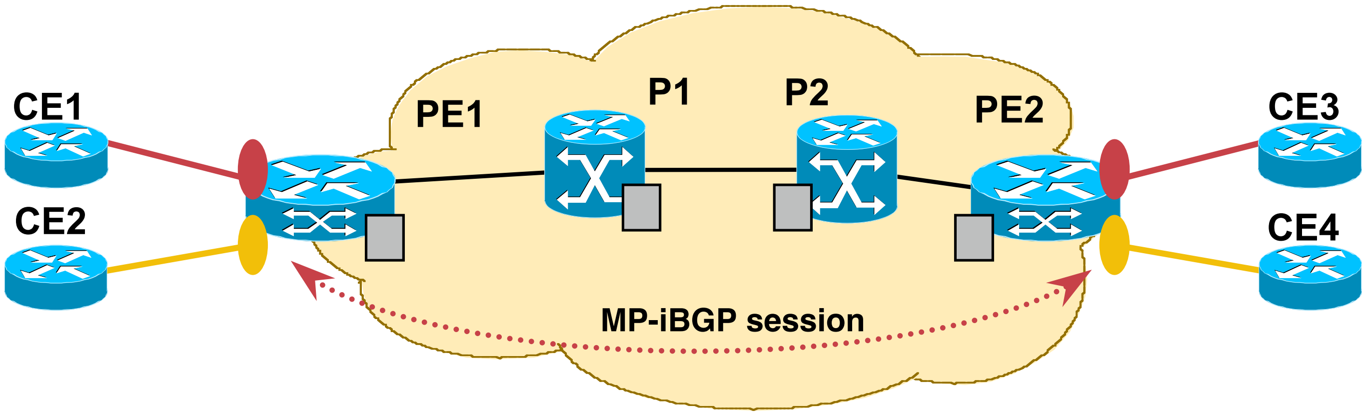
- **VPN services allow**
  - **Customers to use the overlapping address space**
  - **Isolate customer VPNs – Intranets**
  - **Join VPNs - Extranets**
- **MPLS-VPN backbone MUST**
  - **Distinguish between customer addresses**
  - **Forward packets to the correct destination**

VPN Address Overlap



- **BGP propagates ONE route per destination**
  - **Standard path selection rules are used**
- **What if two customers use the same address?**
- **BGP will propagate only one route - PROBLEM !!!**
- **Therefore MP-BGP must DISTINGUISH between customer addresses**

VPN Address Overlap

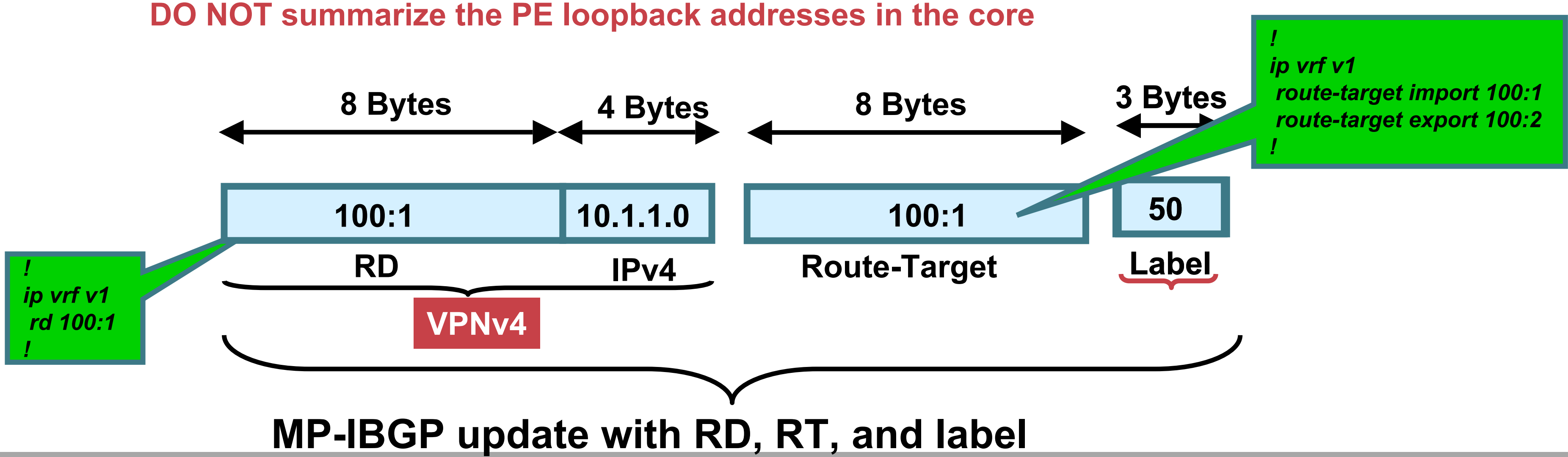


- **When PE router receives VPN routes from MP-BGP how do we know what VRF to place route in?**
- **How do we distinguish overlapping addresses between two VPNs**

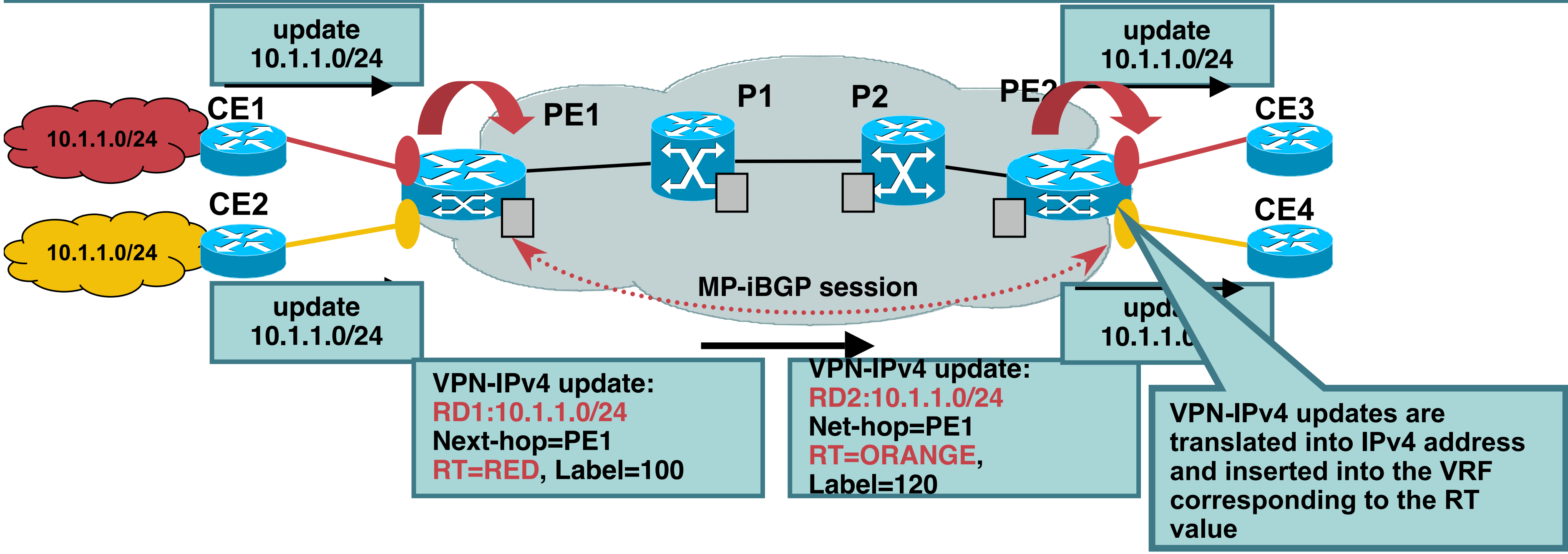
MPLS-VPN Architecture
Control Plane- MP-iBGP Update

- **PE routers exchange VPN-IPv4 updates through MP-iBGP sessions**
- **MP-BGP updates contain VPN-IPv4 addresses and labels**
- **Route Distinguisher makes the address unique across VPNs**
- **Extended Community Route-Target is used for import/export of VPN routes into VRFs**
- **The Label (for the VPNv4 prefix) is assigned only by the PE whose address is the next-hop attribute (Egress PE)**
- **PE addresses used as BGP next-hop must be uniquely known in the backbone IGP**

DO NOT summarize the PE loopback addresses in the core



Site-To-Site Route Propagation



- **MP-BGP prepends an Route Distinguisher (RD) to each VPN route in order to make it unique**
- **MP-BGP assign a Route-Target (RT) to each VPN route to identify its VPN membership.**
- **Routes with Matching RTs are inserted into appropriate VRF table at the receiving PE router.**
- **The label associated with the VPN route is stored and used to send packets towards the destination**

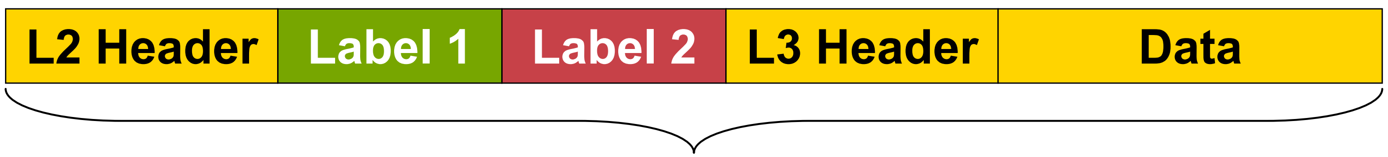
MPLS VPN Forwarding

MPLS VPN Protocols

- OSPF/IS-IS
  - Used as IGP provides reachability between all Label Switch Routers (PE <-> P <-> PE)
- TDP/LDP
  - Distributes label information for IP destinations in core
- MP-BGP4
  - Used to distribute VPN routing information between PE’s
- RIPv2/BGP/OSPF/eiGRP/ISIS/Static
  - Can be used to route between PE and CE

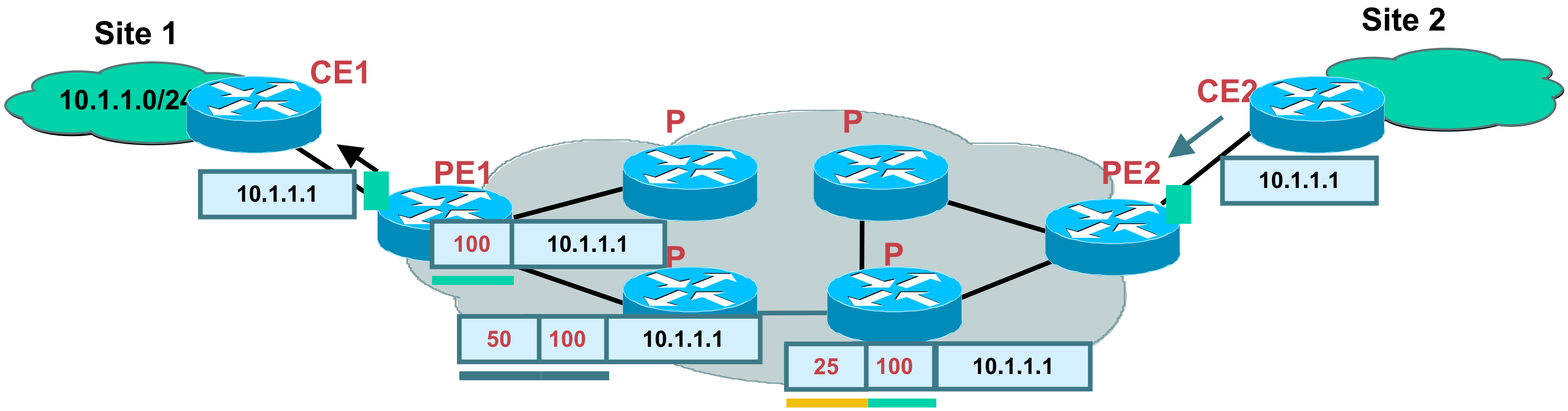
MPLS-VPN Architecture Forwarding Plane

- Forwarding is done through standard MPLS mechanisms using a 2 label deep label stack
  - More if Traffic Engineering or Carrier's Carrier
- The first label is distributed by LDP
  - Derived from an IGP route
  - Corresponds to a PE address (VPN egress point)
  - PE addresses are MP-BGP next-hops of VPN routes
- The second label is distributed MP-BGP
  - Corresponds to the actual VPN route
  - Identifies the PE outgoing interface or routing table



Frame, e.g. HDLC, PPP, Ethernet

MPLS-VPN Architecture
Forwarding Plane



- **PE2 imposes TWO labels for each packet going to the VPN destination 10.1.1.1**
- **The top label is LDP learned and derived from an IGP route**
  - **Represents LSP to PE address (exit point of a VPN route)**
- **The second label is learned via MP-BGP**
  - **Corresponds to the VPN address**

MPLS Tutorial SANOG

Introduction to MPLS Traffic Engineering

Agenda

- **Introduction**
- **Traffic Engineering by tweaking IGPs**
- **Limitations of the Overlay Model**

What is Traffic Engineering??

- **Preventing a situation where some parts of a service provider network are over-utilized (congested), while other parts under-utilized**
- **Reduce the overall cost of operations by more efficient use of bandwidth resources**

The ultimate goal is cost saving !

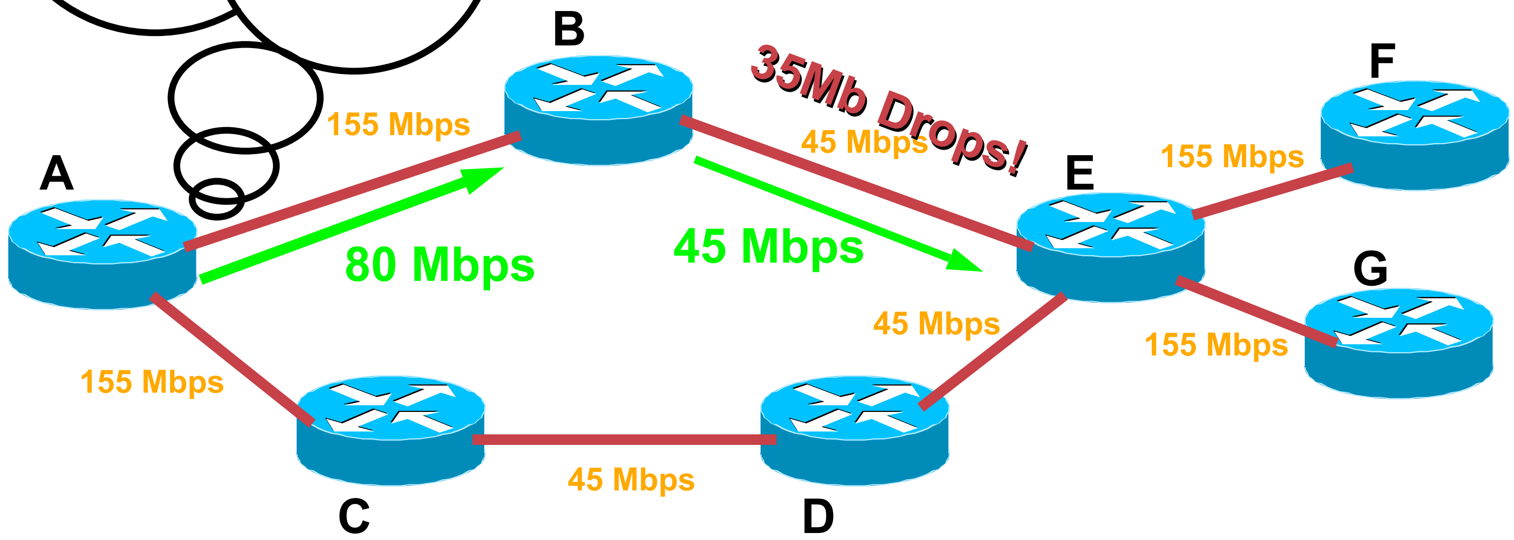
ISSUES WITH IGP ROUTING

- IGPs forward packets based on shortest path (metric).
- Flows from multiple sources may go over some common link(s) causing congestion.
- Alternate longer and underutilized path will not be used.
- IGP metric change may have side effects.

The Problem With Shortest-Path

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30

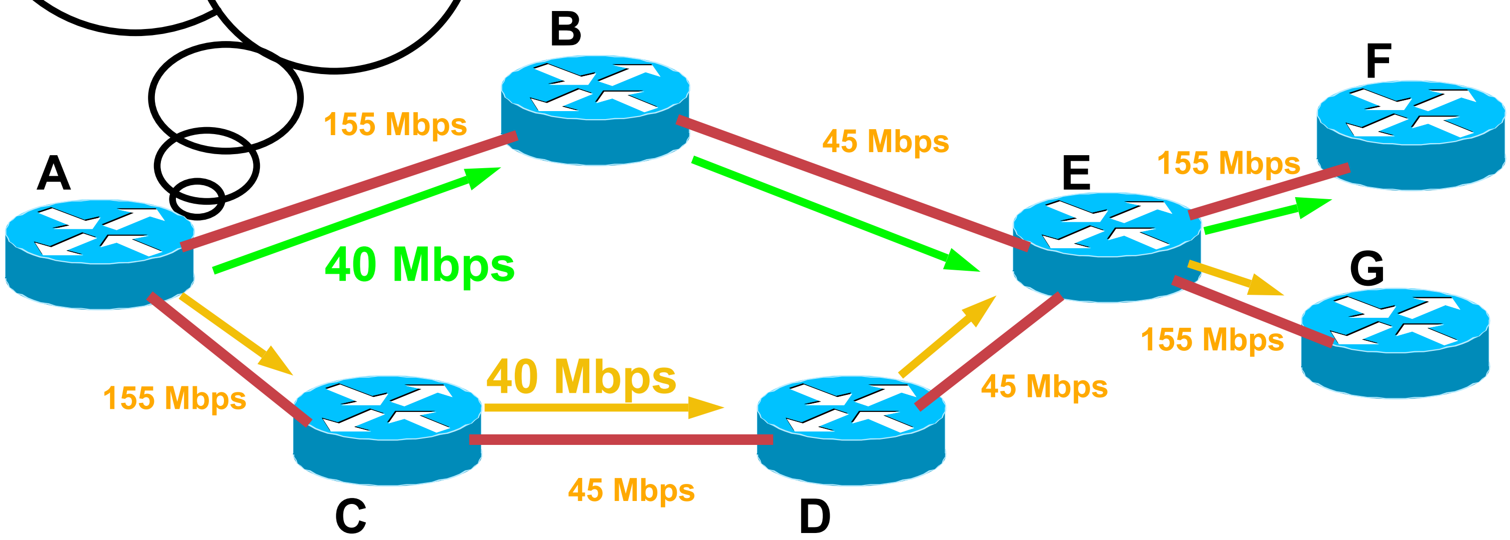
- Assume “A” has 40Mb of traffic for “F” and 40Mb of traffic for “G”
- Some links are 45 Mbps, some are 155 Mbps
- Massive (44%) packet loss between “B” and “E”
- Changing path to A->C->D->E won’t help



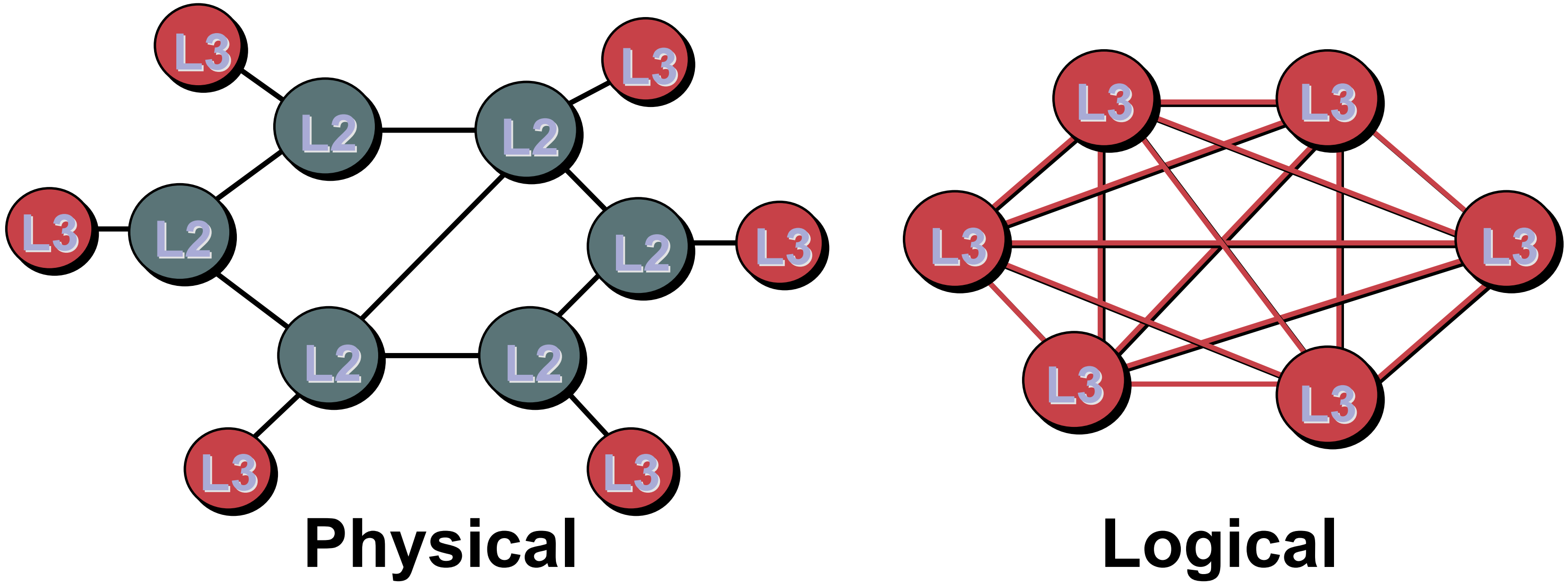
MPLS-TE Example

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	Tunnel0	30
G	Tunnel1	30

- **Assume "A" has 40Mb of traffic for "F" and 40Mb of traffic for "G"**
- **"A" computes paths on properties other than just shortest cost (available bandwidth)**
- **No congestion!**



The "Overlay" Solution



- **Routing at layer 2 (ATM or FR) is used for traffic engineering**
- **Full mesh of VCs between routers. Each router has a direct VC to every other router in the mesh.**

“Overlay” solution: drawbacks

- **Extra network devices (cost)**
- **More complex network management (cost)**
  - **two-level network without integrated network management**
  - **additional training, technical support, field engineering**
- **IGP routing scalability issue for meshes**

Traffic engineering with Layer 3 what is missing ?

- **Path Computation based just on IGP metric is not enough.**
- **Packet forwarding in IP network is done on a hop by hop basis, derived from IGP.**
- **Support for “explicit” routing (aka “source routing”) is not available.**

Motivation for Traffic Engineering

- **Increase efficiency of bandwidth resources**

  Prevent over-utilized (congested) links whilst other links are under-utilized

- **Ensure the most desirable/appropriate path for some/all traffic**

  Explicit-Path overrides the shortest path selected by the IGP

- **Replace ATM/FR cores**

  PVC-like traffic placement without IGP full mesh and associated O(N^2) flooding

- **The ultimate goal is COST SAVING**

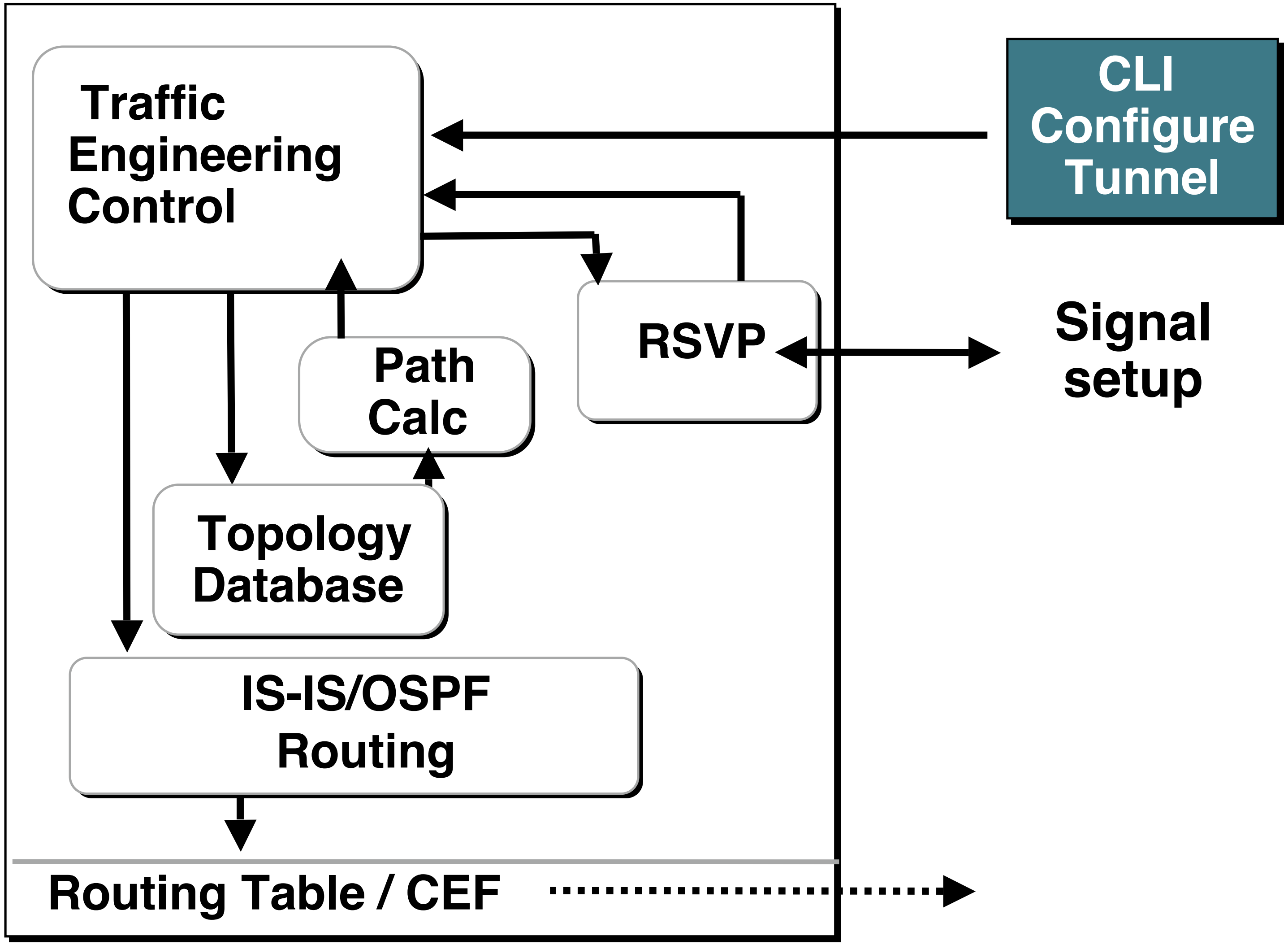
  Service development also progressing

TE tunnel basics

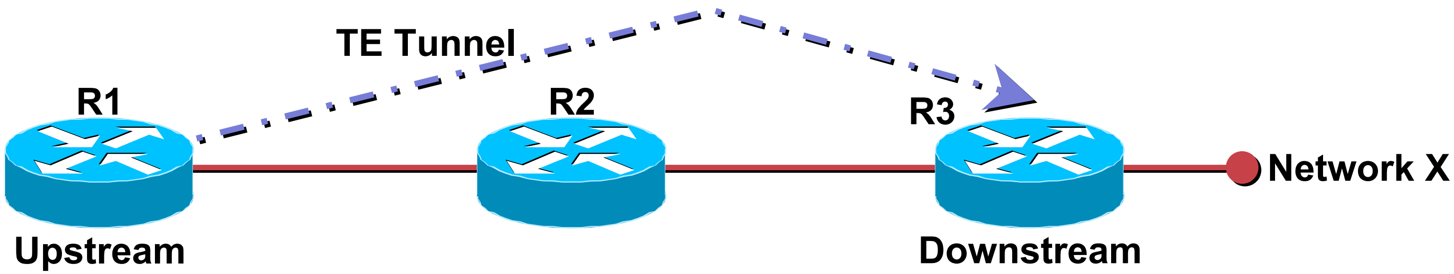
Agenda

- MPLS-TE router operation
- Tunnel attributes:
  - Bandwidth
  - Priority
  - Metric selection
  - Affinity
- Tunnel Path selection

Tunnel Setup



A Terminology Slide—Head, Tail, LSP, etc.



- **Head-End is a router on which a TE tunnel is configured (R1)**
- **Tail-End is the router on which TE tunnel terminates (R3)**
- **Mid-point is a router thru which the TE tunnel passes (R2)**
- **LSP is the Label Switched Path taken by the TE tunnel, here R1-R2-R3**
- **Downstream router is a router closer to the tunnel tail**
- **Upstream router is farther from the tunnel tail (so R2 is upstream to R3's downstream, R1 is upstream from R2's downstream)**

Trunk Attributes

- Tunnel attributes are characteristics the tunnel requires to have on the links along the LSP.
- Configured at the head-end of the trunk
- These are:
  - Bandwidth
  - Priority
  - Metric selection (TE vs. IGP metric)
  - Affinity

```
interface Tunnel0
  tunnel mpls traffic-eng bandwidth Kbps
  tunnel mpls traffic-eng priority pri [hold-pri]
  tunnel mpls traffic-eng path-selection metric {te|igp}
  tunnel mpls traffic-eng affinity properties [mask]
```

Tunnel Bandwidth

```
tunnel mpls traffic-eng bandwidth Kbps
```

- **Bandwidth required by the tunnel across the network**
- **If not configured, tunnel is requested with zero bandwidth.**

Priority

```
tunnel mpls traffic-eng <S> {H}
```

- Configured on tunnel interface
- S = setup priority (0–7)
- H = holding priority (0–7)
- Lower number means higher priority

Priority

- **Setup priority of new tunnel on a link is compared to the hold priority of an existing tunnel**
- **New tunnel with better setup priority will force preemption of already established tunnel with lower holding priority**
- **Preempted tunnel will be torn down and will experience traffic black holing. It will have to be re-signaled**
- **Recommended that S=H; if a tunnel can setup at priority "X", then it should be able to hold at priority "X" too!**
- **Configuring S > H is illegal; tunnel will most likely be preempted**
- **Default is S = 7, H = 7**

Metric Selection (TE vs. IGP metric)

```
tunnel mpls traffic-eng path-
selection metric {te|igp}
```

- **Configure admin weight == interface delay**
- **Configure VoIP tunnels to use TE metric to calculate the path cost**
- **Can be used as a Delay-sensitive metric**

Tunnel Affinity

- Tunnel is characterized by a
  - Tunnel Affinity: 32-bit resource-class affinity
  - Tunnel Mask: 32-bit resource-class mask (0= don't care, 1= care)

Link is characterized by a 32-bit resource-class attribute string called Link Affinity

Default-value of tunnel/link bits is 0

Default value of the tunnel mask = 0x0000FFFF

Tunnel Affinity (Cont.)

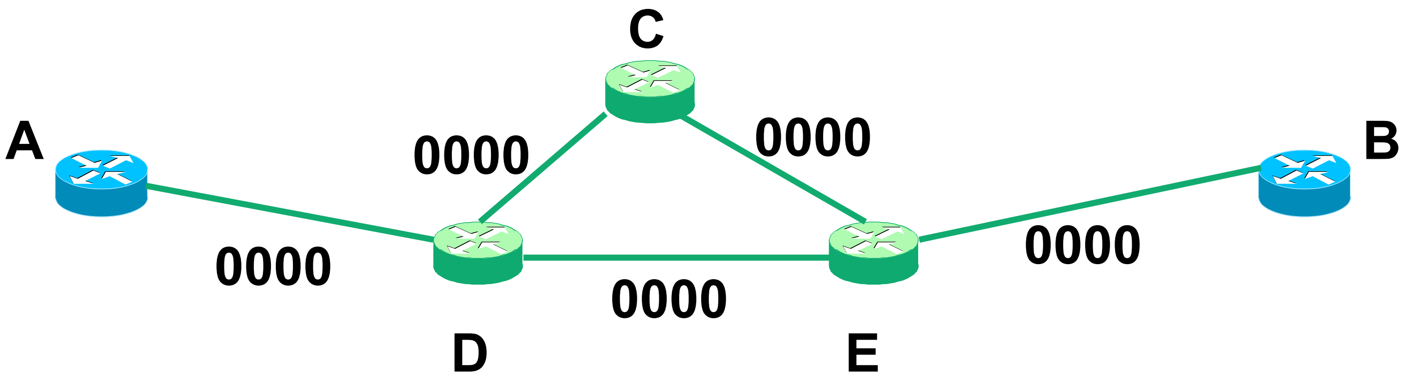
- **Affinity helps select which tunnels will go over which links**
- **A network with OC-12 and Satellite links will use affinities to prevent tunnels with VoIP traffic from taking the satellite links**

Tunnel can only go over a link if

(Tunnel Mask) AND (Link Affinity) == Tunnel Affinity

[redacted]

Example0: 4-bit string, default

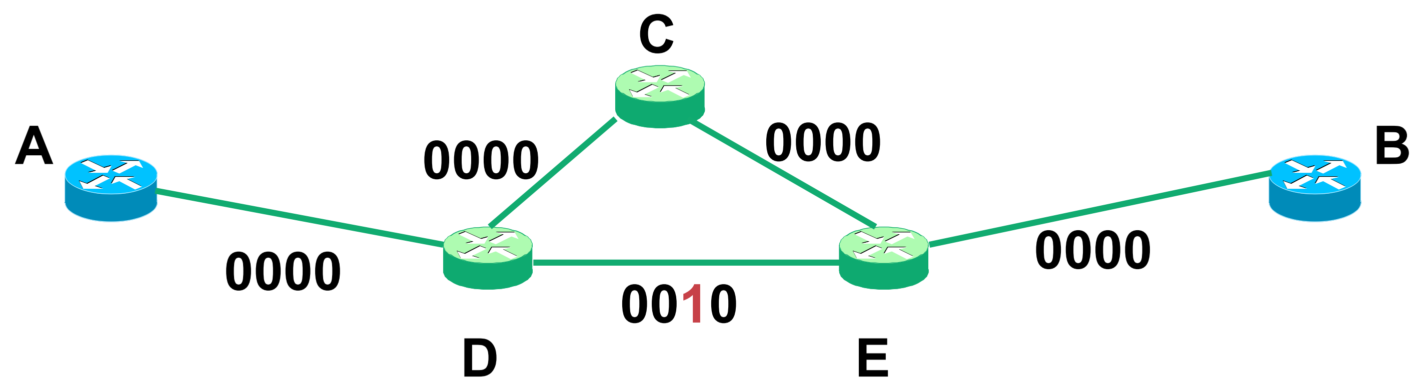


- **Trunk A to B:**

  tunnel = 0000, t-mask = 0011

- **ADEB and ADCEB are possible**

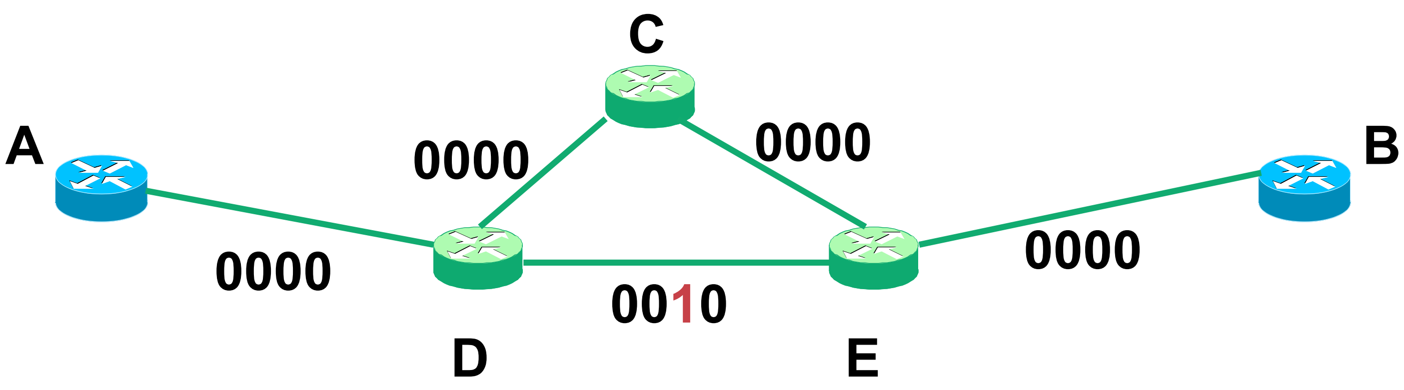
Example1a: 4-bit string



- Setting a link bit in the lower half drives all tunnels off the link, except those specially configured
- Trunk A to B:

  tunnel = 0000, t-mask = 0011
- Only ADCEB is possible

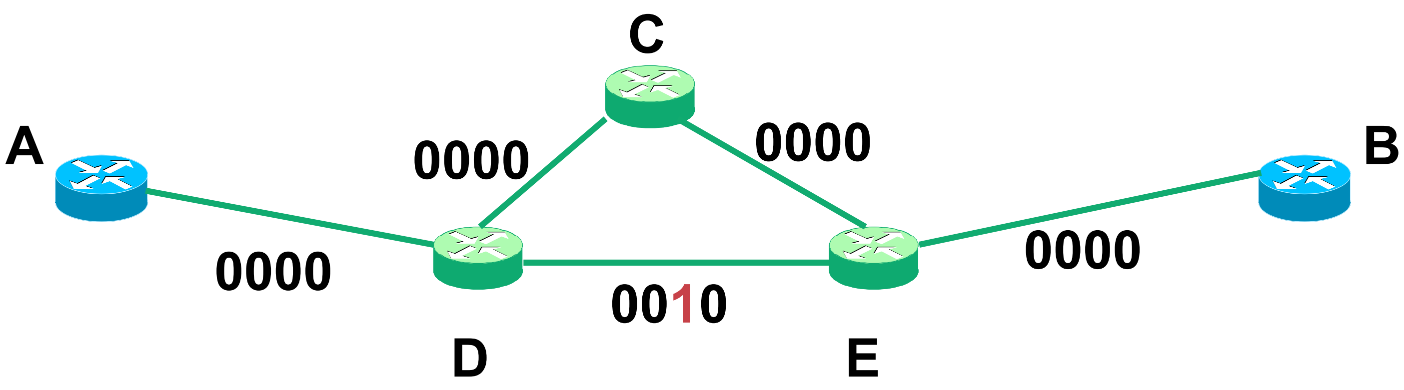
Example1b: 4-bit string



- A specific tunnel can then be configured to allow such links by clearing the bit in its affinity attribute mask
- Trunk A to B:

  tunnel = 0000, t-mask = 0001
- Again, ADEB and ADCEB are possible

Example1c: 4-bit string

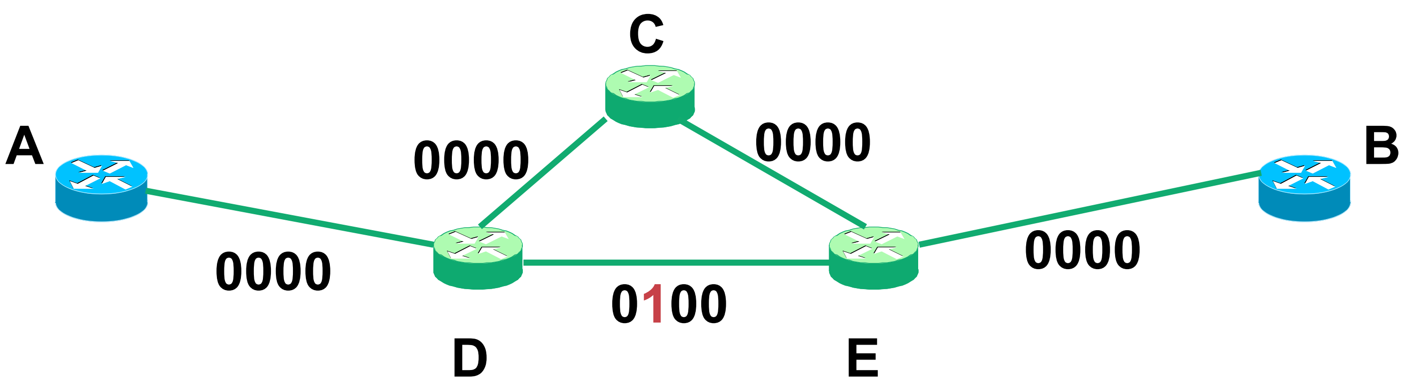


- A specific tunnel can be restricted to only such links by instead turning on the bit in its affinity attribute bits
- Trunk A to B:

  tunnel = 0010, t-mask = 0011

- No path is possible

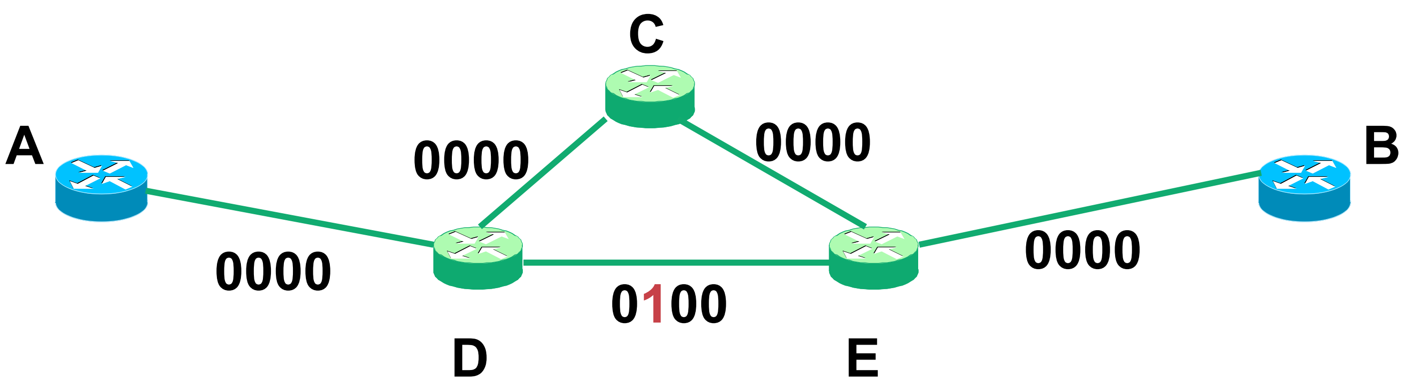
Example2a: 4-bit string



- **Setting a link bit in the upper half drives has no immediate effect**
- **Trunk A to B:**

  tunnel = 0000, t-mask = 0011
- **ADEB and ADCEB are both possible**

Example2b: 4-bit string

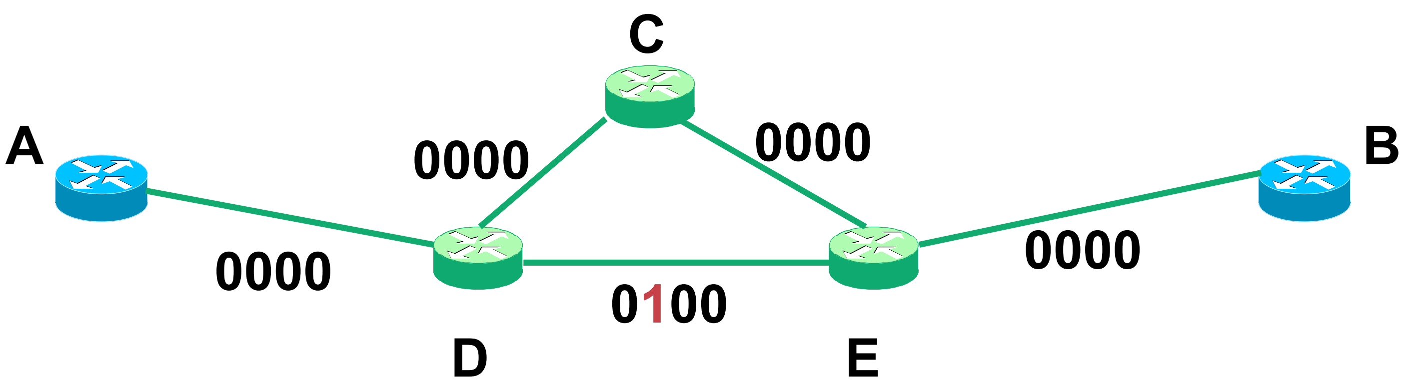


- A specific tunnel can be driven off the link by setting the bit in its mask
- Trunk A to B:

  tunnel = 0000, t-mask = 0111

- Only ADCEB is possible

Example2c: 4-bit string



- A specific tunnel can be restricted to only such links
- Trunk A to B:

  tunnel = 0100, t-mask = 0111
- No path is possible

Tunnel Path Selection

- Tunnel has two path options
  1. Dynamic
  2. Explicit
- Path is a set of next-hop addresses (physical or loopbacks) to destination
- This set of next-hops is called Explicit Route Address (ERO)

Dynamic Path Option

```
tunnel mpls traffic-eng path-option <prio>
dynamic
```

- **dynamic = router calculates path using TE topology database**
- **Router will take best IGP path that meets BW requirements**
- **If BW=0, tunnel could take the IGP path**

Explicit Path Option

```
tunnel mpls traffic-eng path-option
<prio> explicit <id|name> [ID|NAME]>
```

- explicit = take specified path
- Strict source-routing of IP traffic

Explicit Path Option (Cont.)

```
ip explicit-path <id|name> [ID|NAME]
    next-address 192.168.1.1
    next-address 192.168.2.1 {loose}
    ...
```

- explicit = take specified path
- Router sets up path you specify
- Strict source-routing of IP traffic
- Each hop is a physical interface or loop back

How does ERO come into play?

- **If dynamic path-option is used, TE topology database is used to COMPUTE the Explicit Path**
- **If explicit path-option is used, TE topology database is used to VERIFY the Explicit Path**

MPLS-TE: Link attributes, IGP enhancements, CSPF

Agenda

- **Link Attributes**
- **Information flooding**
- **IGP Enhancements for MPLS-TE**
- **Path Computation (C-SPF)**

Link Attributes

- Link attributes
  - Bandwidth per priority (0-7)
  - Link Affinity
  - TE-specific link metric

Bandwidth

```
ip rsvp bandwidth <x> <y>
```

- **Per-physical-interface command**
- **X = amount of reservable BW, in K**
- **Y = not used by MPLS-TE**

Link Affinity

```
mpls traffic-eng attribute-flags <0x0-0xFFFFFFFF>
```

- **Per-physical-interface command**

Administrative Weight

mpls traffic-eng administrative-weight <X>

- Per-physical-interface command
- X = 0–4,294,967,295
- Gives a metric that be considered for use instead of the IGP metric
- This can be used as a per-tunnel delay-sensitive metric for doing VoIP TE
- By default TE metric is used. However, when no TE metric is configured,

    IGP metric => TE metric

Information Distribution

- TE LSPs can (optionally) reserve bandwidth across the network
- Reserving bandwidth is one of the ways to find more optimal paths to a destination
- This is a control-plane reservation only
- Need to flood available bandwidth information across the network
- IGP extensions flood this information
    - OSPF uses Type 10 (area-local) Opaque LSAs
    - ISIS uses new TLVs

Information Distribution

- **A link-state protocol has to be used as the IGP (IS-IS or OSPF)**
- **A Link-state protocol is not a requirement for other MPLS applications (e.g. VPNs)**

Need for a Link-State Protocol

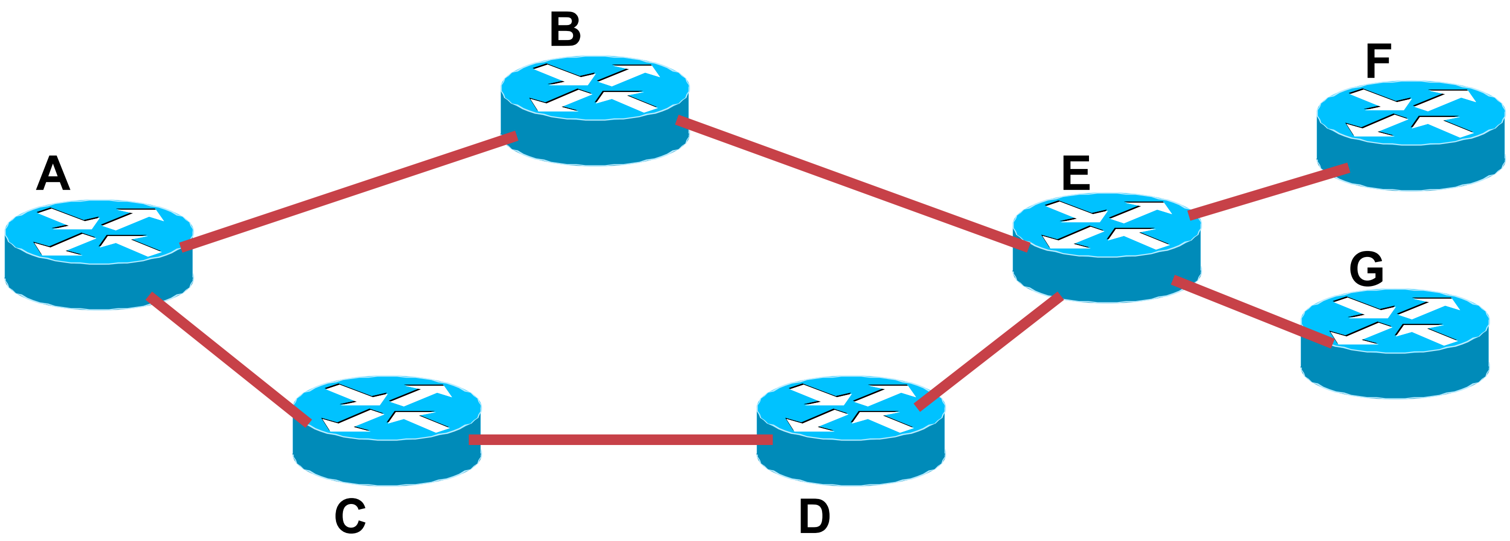
Why is a link-state protocol required?

- Path is computed at the source
- Source needs entire picture (topology) of the network to make routing decision
- Only link-state protocols flood link information to build a complete network topology

Need for a Link-State Protocol

Consider the following network:

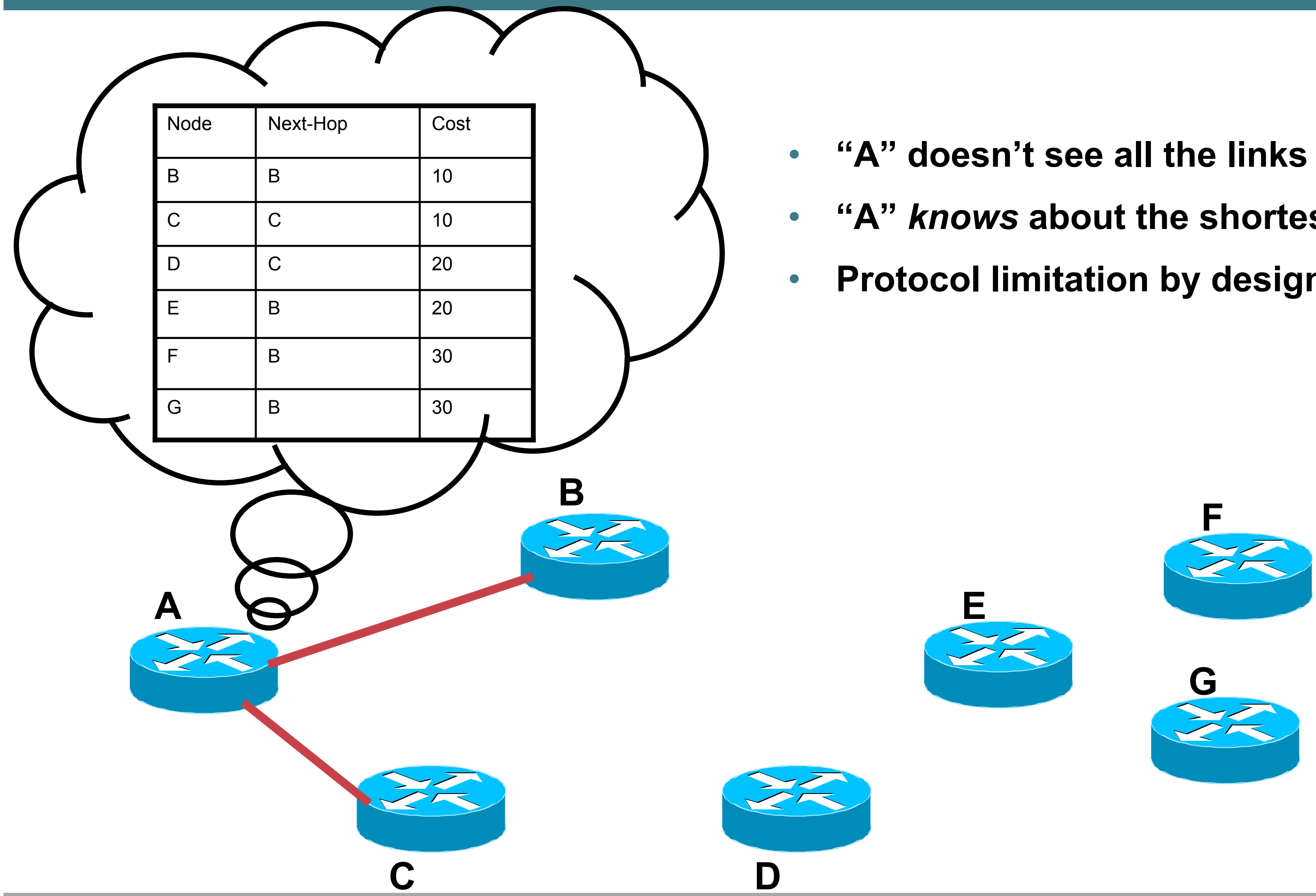
- **All links have a cost of 10**
- **Path from “A” to “E” is A->B->E, cost 20**
- **All traffic from “A” to {E,F,G} goes A->B->E**



What a Distance Vector Protocol Sees

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30

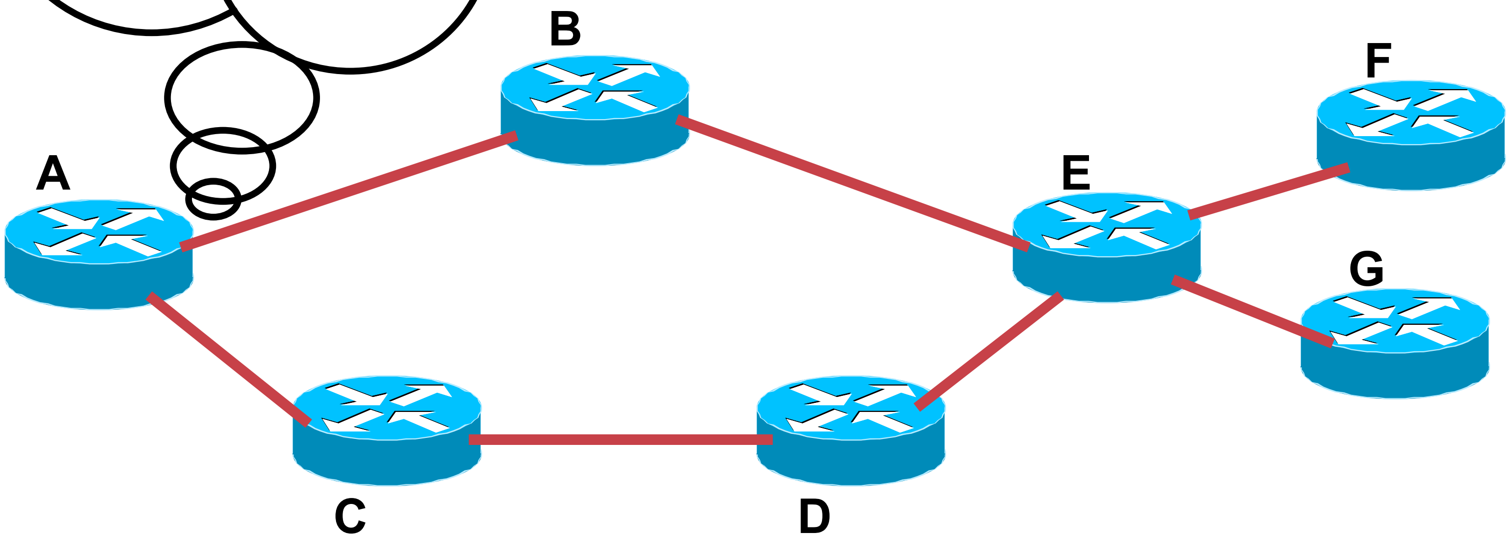
- "A" doesn't see all the links
- "A" *knows* about the shortest path
- Protocol limitation by design



What a Link-State Protocol Sees

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30

- “A” sees all links
- “A” *computes* the shortest path
- Routing table doesn’t change



Link-State Protocol Extensions/ IGP Flooding

- TE finds paths other than shortest-cost
- To do this, TE must have more info than just per-link cost
- OSPF and IS-IS have been extended to carry additional information
  - Physical bandwidth
  - RSVP configured bandwidth
  - RSVP Available bandwidth
  - Link TE metric
  - Link affinity

OSPF Extensions

- **OSPF**

  Uses Type 10 (Opaque Area-Local) LSAs

  See draft-katz-yeung-ospf-traffic

IS-IS Extensions

- **IS-IS**
  - **Uses Type 22 TLVs**
  - **See draft-ietf-isis-traffic**

- Extended IS neighbor subTLVs
  - subTLV #3 - administrative group (color)
  - subTLV #6 - IPv4 interface address
  - subTLV #8 – IPv4 neighbor address
  - subTLV #9 - maximum link bandwidth
  - subTLV#10 - maximum reservable link BW
  - subTLV #11 - current bandwidth reservation
  - subTLV #18 - default TE metric

Information Distribution

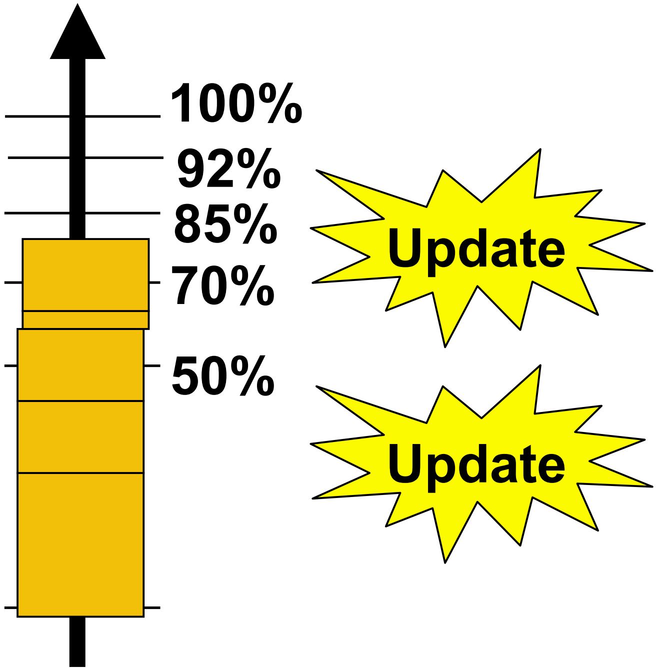
- Dynamics of ISIS and OSPF are unchanged

  Periodic flooding

  Hold-down timer to constrain the frequency of advertisements

- Current constraint information sent when IGP decides to re-flood
- TE admission control requests re-flooding on significant changes
  - *significant* is determined by a configurable set of thresholds
  - On link configuration changes
  - On link state changes
  - On LSP Setup failure
  - TE refresh timer expires (180 seconds default)

Significant Change



- **Each time a threshold is crossed, an update is sent**
- **Denser population as utilization increases**
- **Different thresholds for UP and Down**

```
router#sh mpls traffic-eng link bandwidth-allocation pos4/0
 .............<snip>...................
    Up Thresholds:       15 30 45 60 75 80 85 90 95 96 97 98 99 100 (default)
   Down Thresholds:     100 99 98 97 96 95 90 85 80 75 60 45 30 15 (default)
.............<snip>....................
```

Per-Priority Available BW

T=0

D advertises: AB(0)=100=…= AB(7)=100
AB(i) = 'Available Bandwidth at priority I"

T=1

Setup of a tunnel over L at priority=3 for 30 units

T=2

D advertises: AB(0)=AB(1)=AB(2)=100
AB(3)=AB(4)=…=AB(7)=70

T=3

Setup of an additional tunnel over L at priority=5 for 30 units

T=4

D advertises: AB(0)=AB(1)=AB(2)=100
AB(3)=AB(4)=70
AB(5)=AB(6)=AB(7)=40

This means that another tunnel having the piority < 3 and Bw > 70M would preempt the previous installed tunnel

Constrained-based Path Computation (C-SPF)

Path Calculation

- Modified Dijkstra at tunnel head-end
- Often referred to as CSPF
  - Constrained SPF
- …or PCALC (path calculation)
- Final result is explicit route meeting desired constrain

Path Calculation (C-SPF)

- **Shortest-cost path is found that meets administrative constraints**
- **These constraints can be**

  bandwidth

  link attribute (aka color, resource group)

  priority
- **The addition of constraints is what allows MPLS-TE to use paths other than *just* the shortest one**

Path Computation

"On demand" by the trunk's head-end:

- **for a new trunk**
- **for an existing trunk whose (current) LSP failed**
- **for an existing trunk when doing re-optimization**

Path Computation

Input:

- configured attributes of traffic trunks originated at this router
- attributes associated with resources
  - available from IS-IS or OSPF
- topology state information
  - available from IS-IS or OSPF

Path Computation

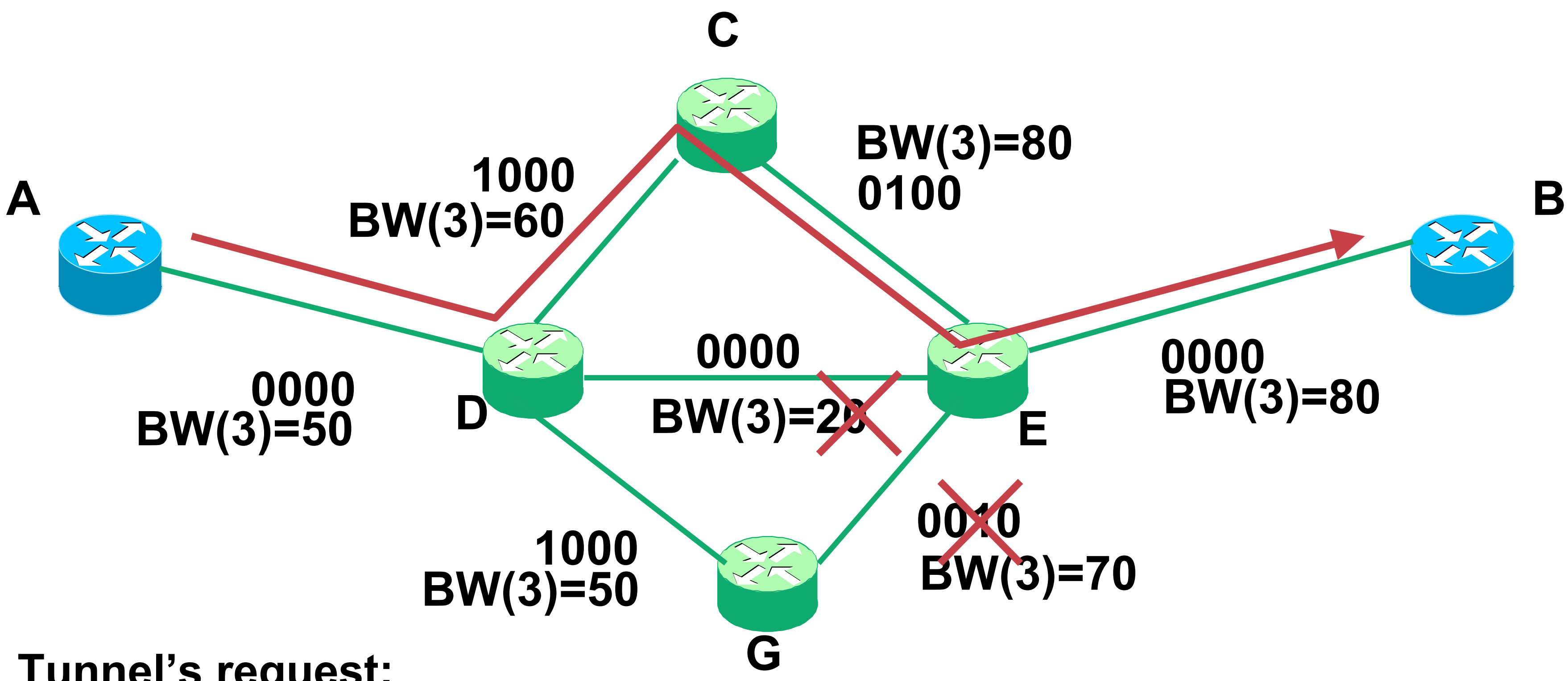
- Prune links if:
  - insufficient resources (e.g., bandwidth)
  - violates policy constraints
- Compute shortest distance path
  - TE uses its own metric
- Tie-break:
  1. Path with the highest available bandwidth
  2. Path with the smallest hop-count
  3. Path found first in TE topology database

Path Computation

Output:

- **explicit route - expressed as a sequence of router IP addresses**
  - **interface addresses for numbered links**
  - **loopback address for unnumbered links**
- **used as an input to the path setup component**

BW/Policy Example

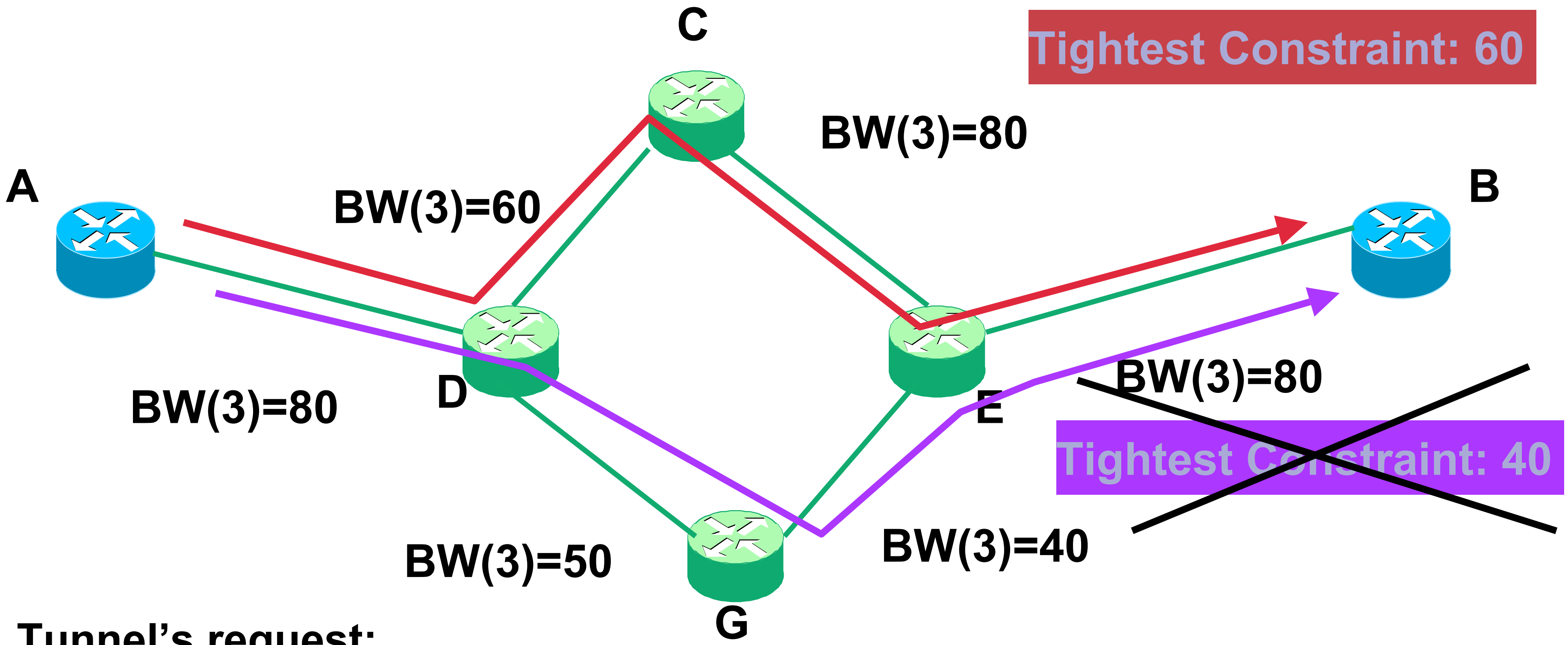


- **Tunnel's request:**

  Priority 3, BW = 30 units,

  Policy string: 0000, mask: 0011

Maximizing the Tightest Constraint

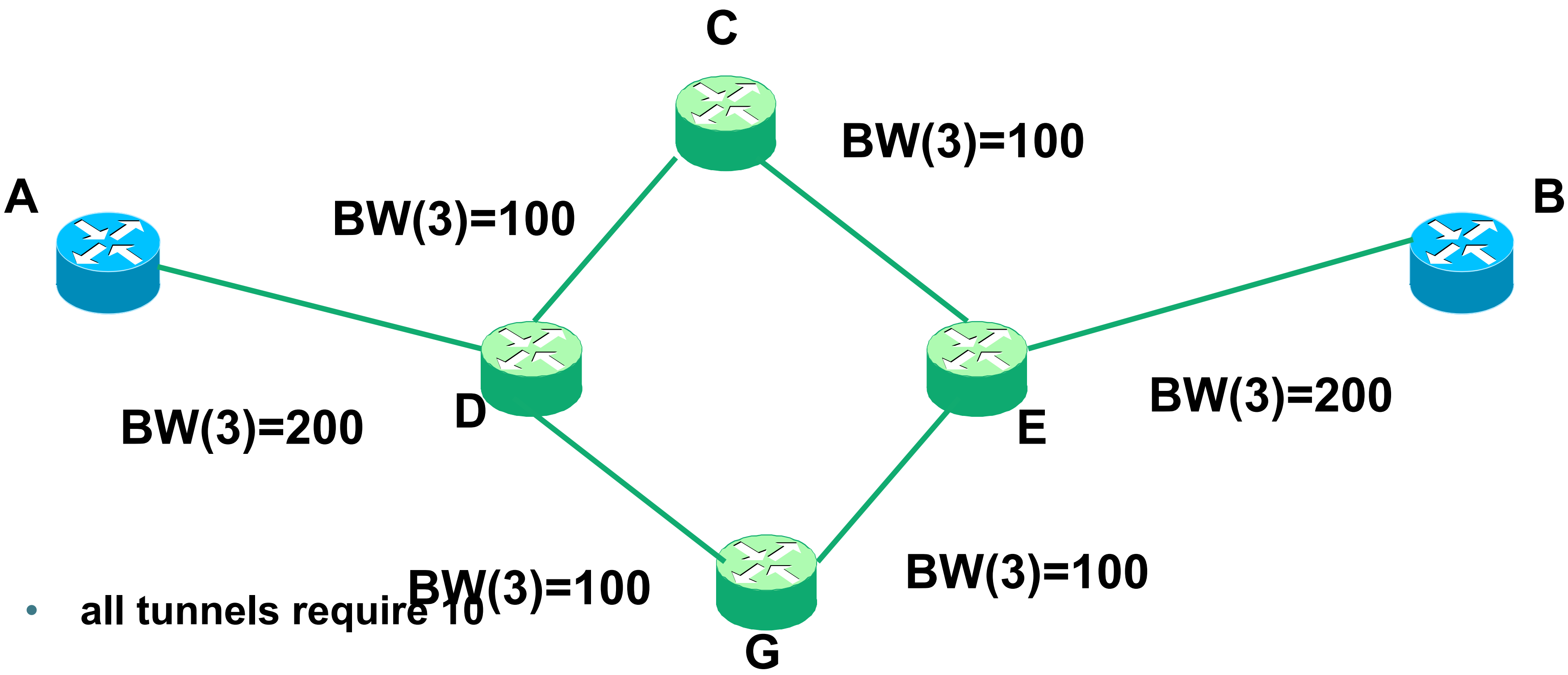


- **Tunnel's request:**

  Priority 3, BW = 30 units,

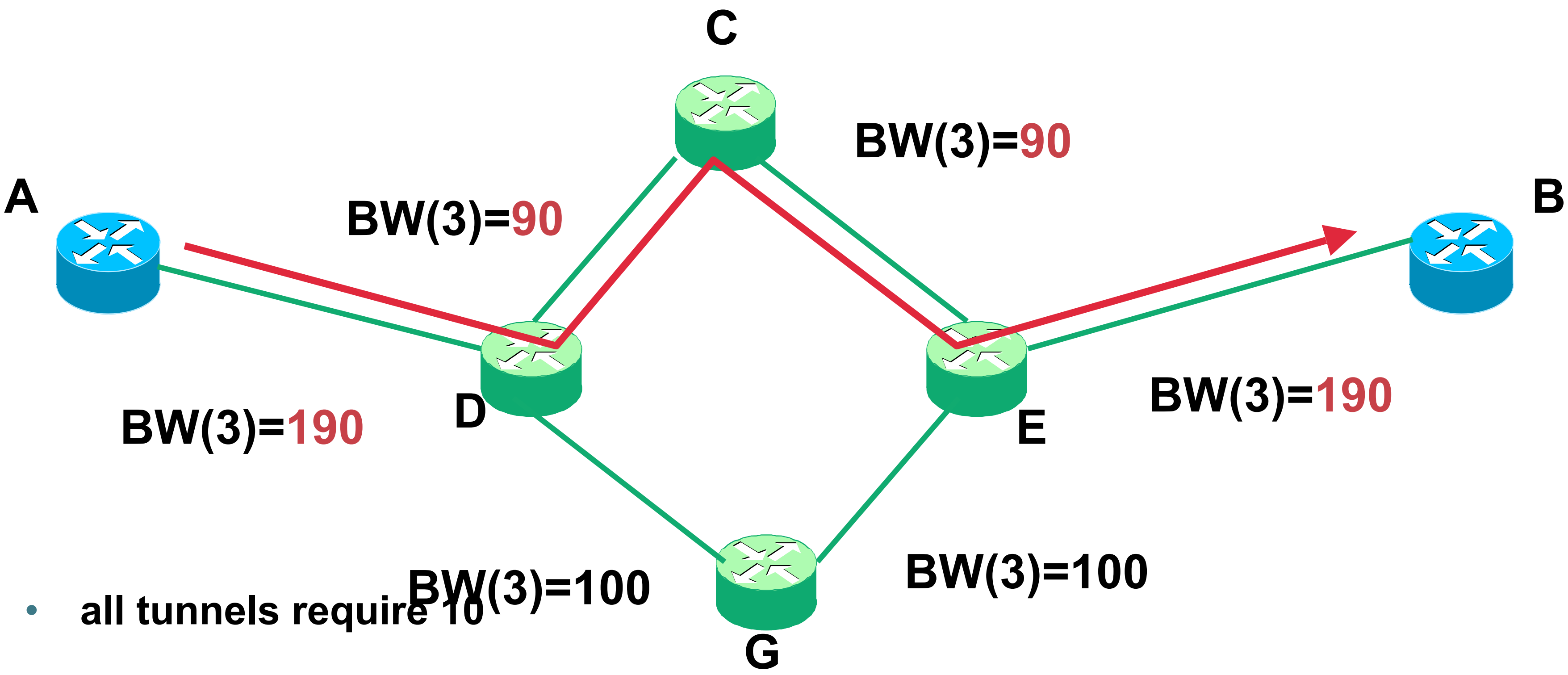
  Policy string: 0000, mask: 0011

Load-Balancing tunnels



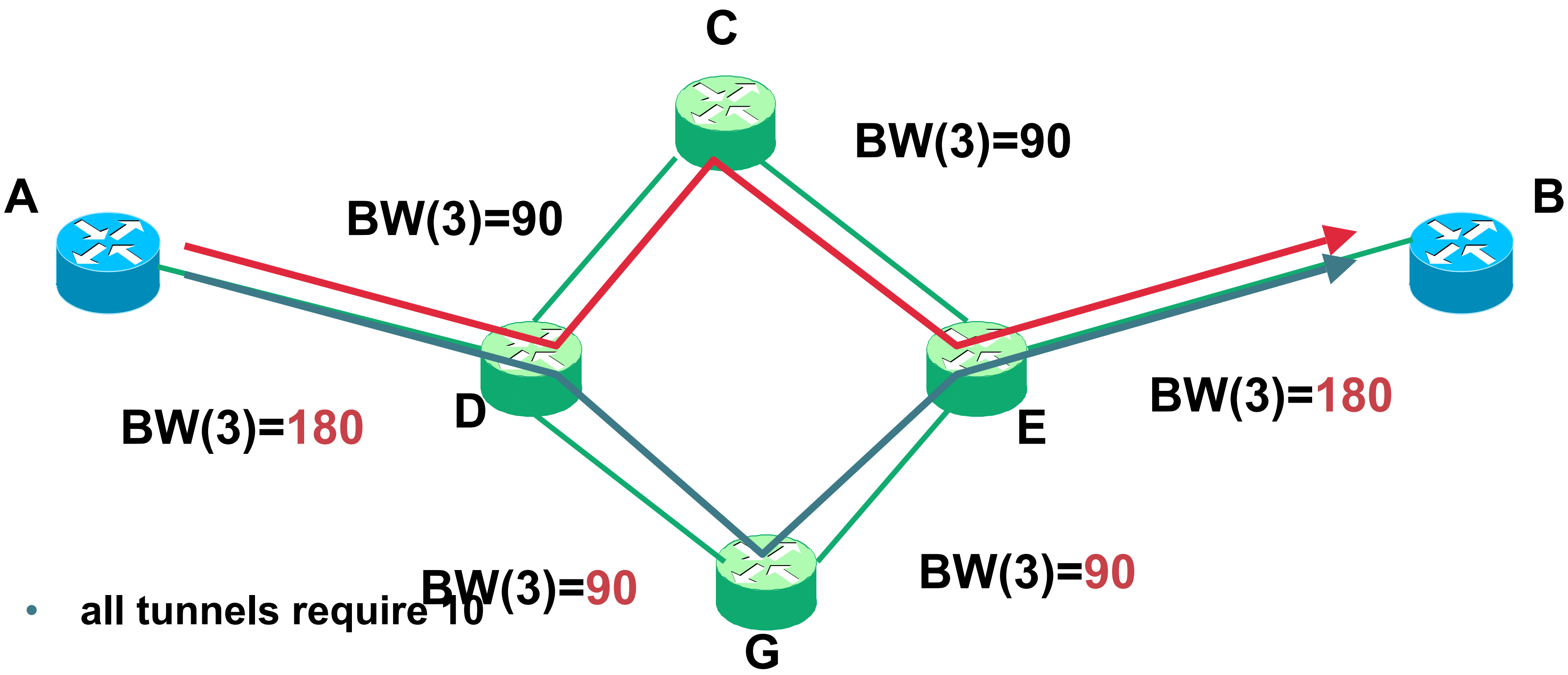
- all tunnels require 10

Load-Balancing tunnels



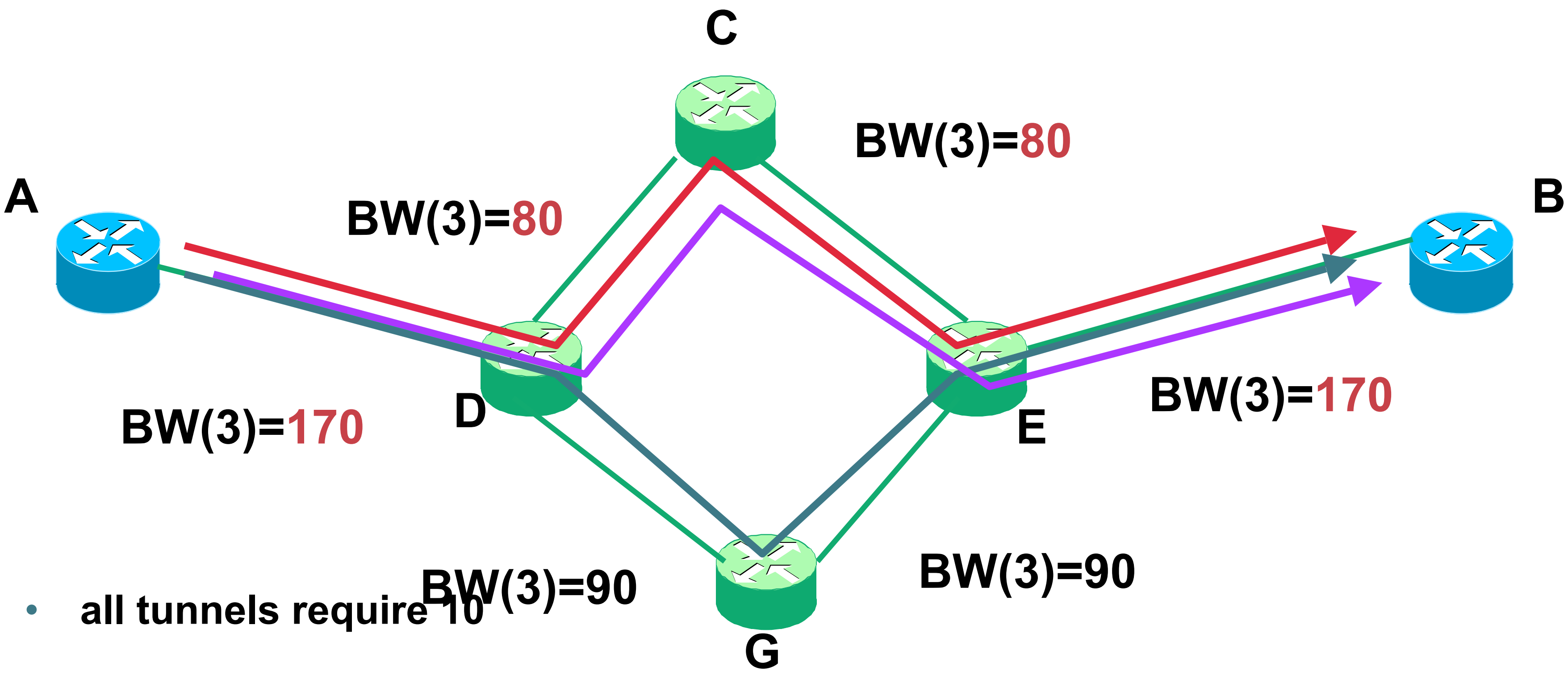
- all tunnels require 10

Load-Balancing tunnels



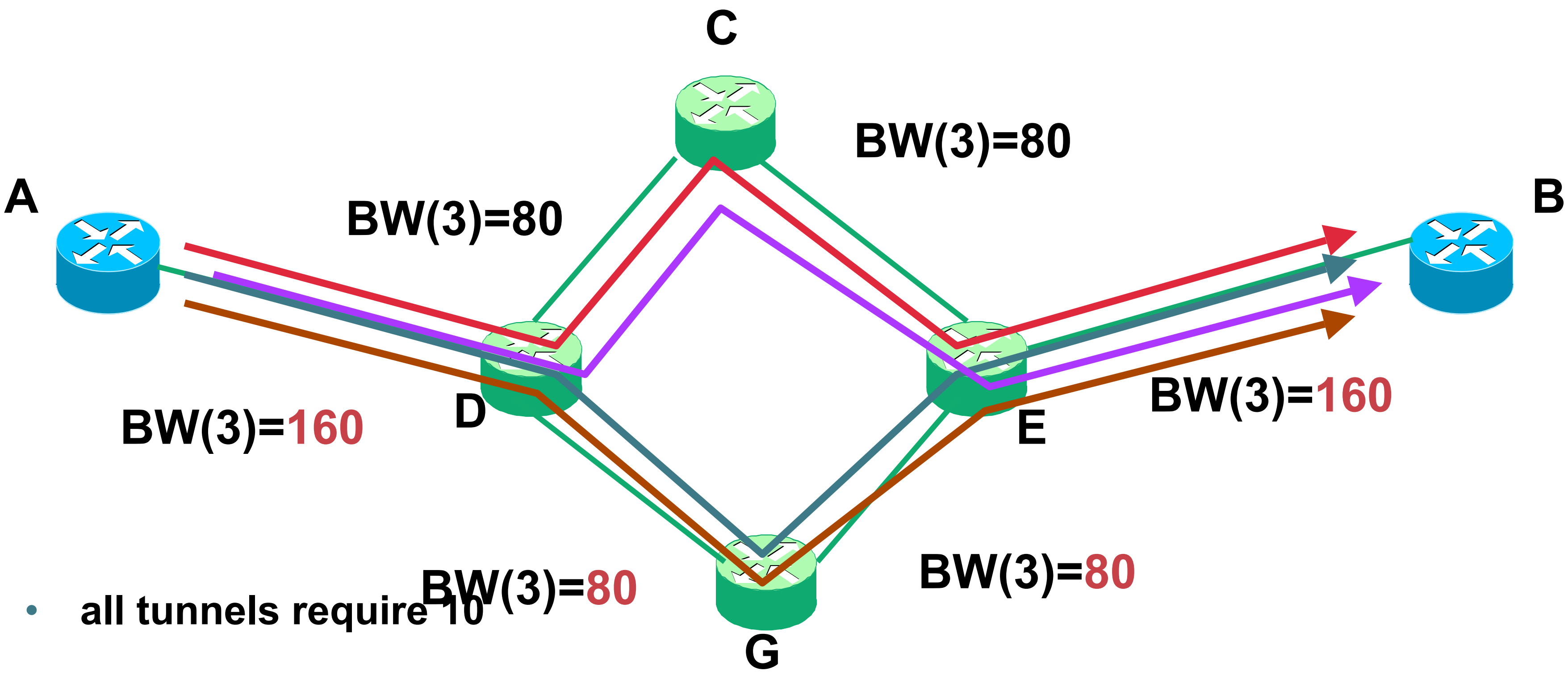
- all tunnels require 10

Load-Balancing tunnels



- all tunnels require 10

Load-Balancing tunnels



- all tunnels require 10

MPLS-TE: RSVP extensions, tunnel signaling and tunnel maintenance

Agenda

- Path Setup (RSVP Extensions)
- Path maintenance
- Reoptimization
- Mapping Traffic to Tunnels
- Using metrics with tunnels
- Load balancing with TE tunnels

Path Setup (RSVP Extensions)

Path Setup

- **After we calculate a path, we need to build an LSP across that path**
- **Path setup is done at the head-end of a trunk with RSVP + TE extensions**
- **RSVP sends PATH messages out, gets RESV messages back**
- **RFC2205, plus RFC 3209**

RSVP Extensions to RFC2205 for LSP Tunnels

- **Downstream-on-demand label distribution**
- **Instantiation of explicit label switched paths**
- **Allocation of network resources (e.g., Bandwidth) to explicit lsps**
- **Re-routing of established lsp-tunnels in a smooth fashion using the concept of make-before-break**
- **Tracking of the actual route traversed by an lsp-tunnel**
- **Diagnostics on lsp-tunnels**
- **Pre-emption options that are administratively controllable**

RSVP Extensions for TE

	PATH	RESV
LABEL_REQUEST	X	
LABEL		X
EXPLICIT_ROUTE	X	
RECORD_ROUTE	X	X
SESSION_ATTRIBUTE	X	

RSVP Label Allocation

- Labels are distributed from down-stream to up-stream
- Label Binding via PATH message - LABEL_REQUEST object
- Labels are allocated & distributed via RESV message using LABEL Object.

RSVP - ERO

- ERO - Explicit Route Object
- "PATH" message carries ERO (concatenation of hops which constitute explicitly routed path) given by the Head-End Router
- This is used in setting up for the LSP
- The path can be administratively specified or dynamically computed

RSVP - Record Route

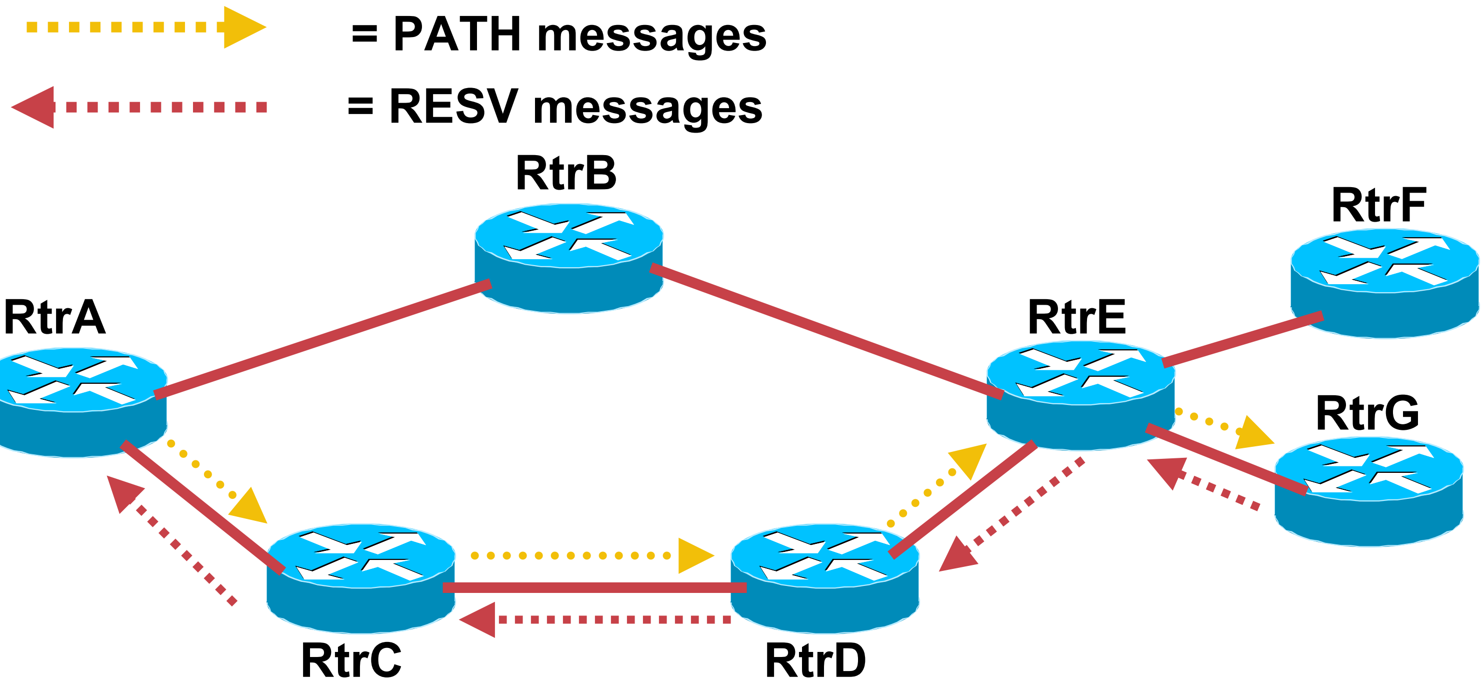
- **Added to the PATH message by the head-end Router.**
- **Every Router along the path records its IP address in the RRO.**
- **Used by the Head-End Router on how the actual LSP has traversed.**
- **Used for Loop Detection**

RSVP - Session Attribute

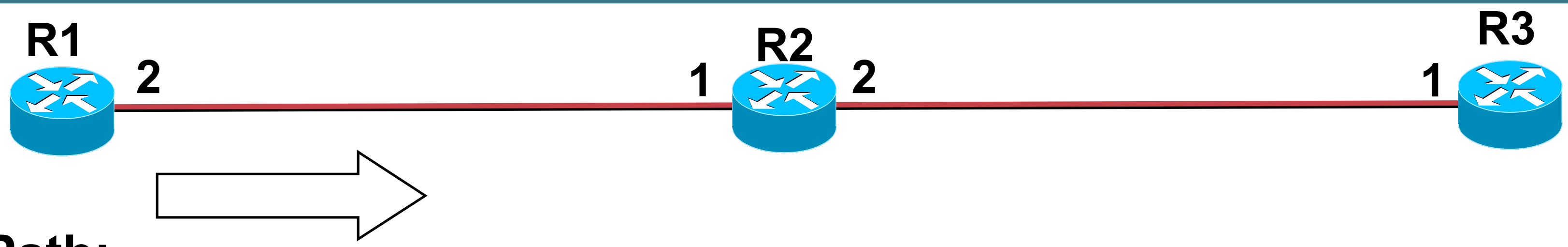
- **Added to "PATH" message by Head-End router to aid in session identification & diagnostics**
  - **setup priority**
  - **hold priorities**
  - **resource affinities**

Path Setup

- PATH message: “Can I have 40Mb along this path?”
- RESV message: “Yes, and here’s the label to use.”
- LFIB is set up along each hop
- PATH messages are refreshed every 30 seconds

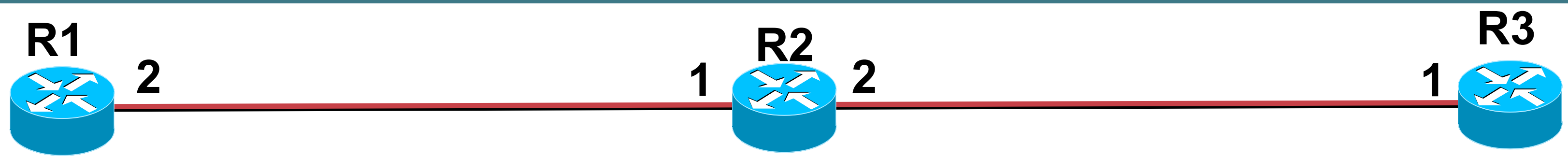


Path Setup - more details



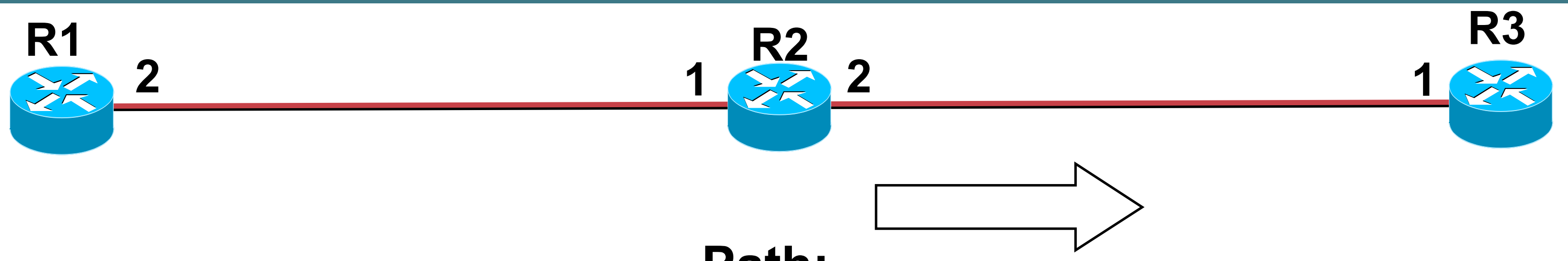
Path:
Common_Header
Session(R3-lo0, 0, R1-lo0)
PHOP(R1-2)
Label_Request(IP)
ERO (R2-1, R3-1)
Session_Attribute (S(3), H(3), 0x04)
Sender_Template(R1-lo0, 00)
Sender_Tspec(2Mbps)
Record_Route(R1-2)

Path Setup - more details



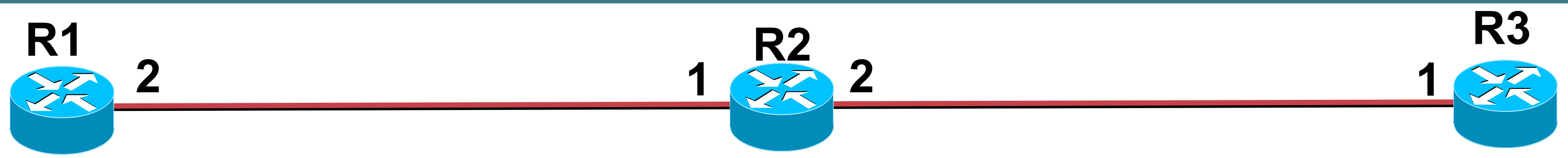
Path State:
Session(R3-lo0, 0, R1-lo0)
PHOP(R1-2)
Label_Request(IP)
ERO (R2-1, R3-1)
Session_Attribute (S(3), H(3), 0x04)
Sender_Template(R1-lo0, 00)
Sender_Tspec(2Mbps)
Record_Route (R1-2)

Path Setup - more details



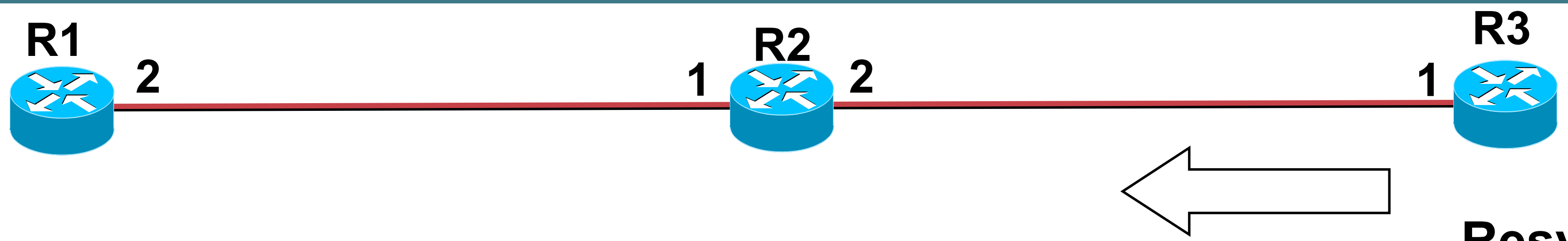
Path:
Common_Header
Session(R3-lo0, 0, R1-lo0)
PHOP(R2-2)
Label_Request(IP)
ERO (R3-1)
Session_Attribute (S(3), H(3), 0x04)
Sender_Template(R1-lo0, 00)
Sender_Tspec(2Mbps)
Record_Route (R1-2, R2-2)

Path Setup - more details



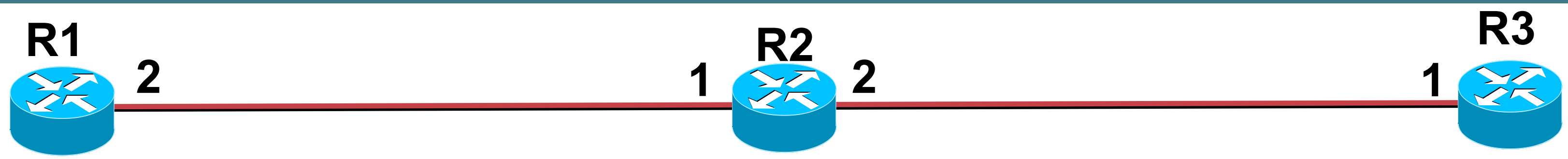
Path State:
Session(R3-lo0, 0, R1-lo0)
PHOP(R2-2)
Label_Request(IP)
ERO ()
Session_Attribute (S(3), H(3), 0x04)
Sender_Template(R1-lo0, 00)
Sender_Tspec(2Mbps)
Record_Route (R1-2, R2-2, R3-1)

Path Setup - more details



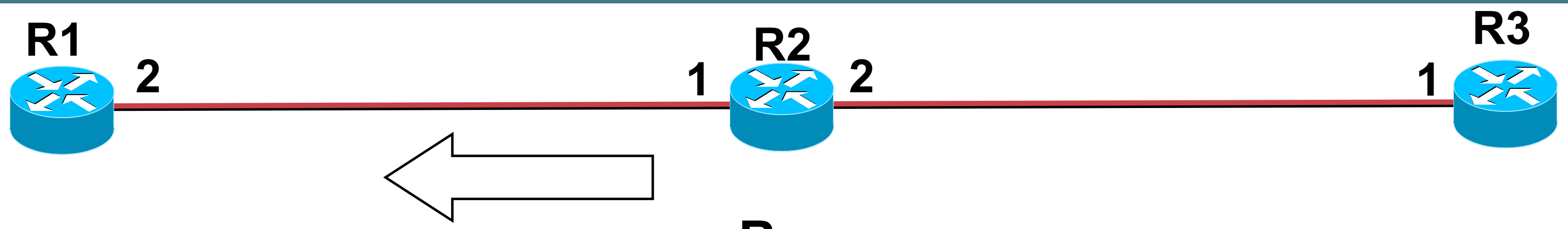
Resv:
Common_Header
Session(R3-lo0, 0, R1-lo0)
PHOP(R3-1)
Style=SE
FlowSpec(2Mbps)
Sender_Template(R1-lo0, 00)
Label=POP
Record_Route(R3-1)

Path Setup - more details



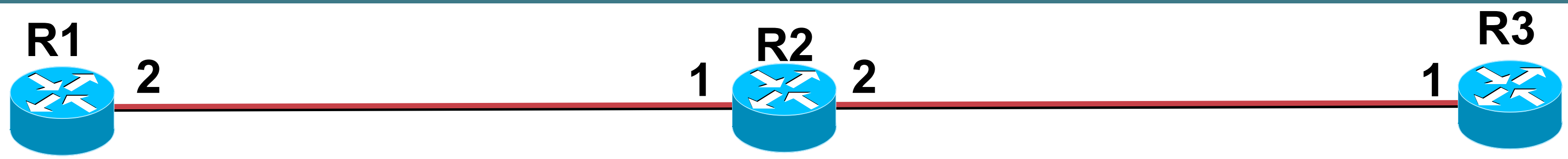
Resv State
Session(R3-lo0, 0, R1-lo0)
PHOP(R3-1)
Style=SE
FlowSpec (2Mbps)
Sender_Template(R1-lo0, 00)
OutLabel=POP
IntLabel=5
Record_Route(R3-1)

Path Setup - more details



Resv:
Common_Header
Session(R3-lo0, 0, R1-lo0)
PHOP(R2-1)
Style=SE
FlowSpec (2Mbps)
Sender_Template(R1-lo0, 00)
Label=5
Record_Route(R2-1, R3-1)

Path Setup - more details



Resv state:
Session(R3-lo0, 0, R1-lo0)
PHOP(R2-1)
Style=SE
FlowSpec (2Mbps)
Sender_Template(R1-lo0, 00)
Label=5
Record_Route(R1-2, R2-1, R3-1)

Trunk Admission Control

- Performed by routers along a Label Switched Path (LSP)
- Determines if resources are available
- May tear down (existing) LSPs with a lower priority
- Does the local accounting
- Triggers IGP information distribution when resource thresholds are crossed
- Since TE tunnels are unidirectional, we do admission control on outbound interfaces only

Path maintenance

Identifying TE-tunnels

SESSION Object

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td colspan="2">IPv4 tunnel tailend address
(32 bits)</td></tr>
<tr><td>Reserved. Must be zero
(16 bits)</td><td>Tunnel ID
(16 bits)</td></tr>
<tr><td colspan="2">IPv4 tunnel headend address
(32 bits)</td></tr>
</table>

SENDER_TEMPLATE / FILTER_SPEC

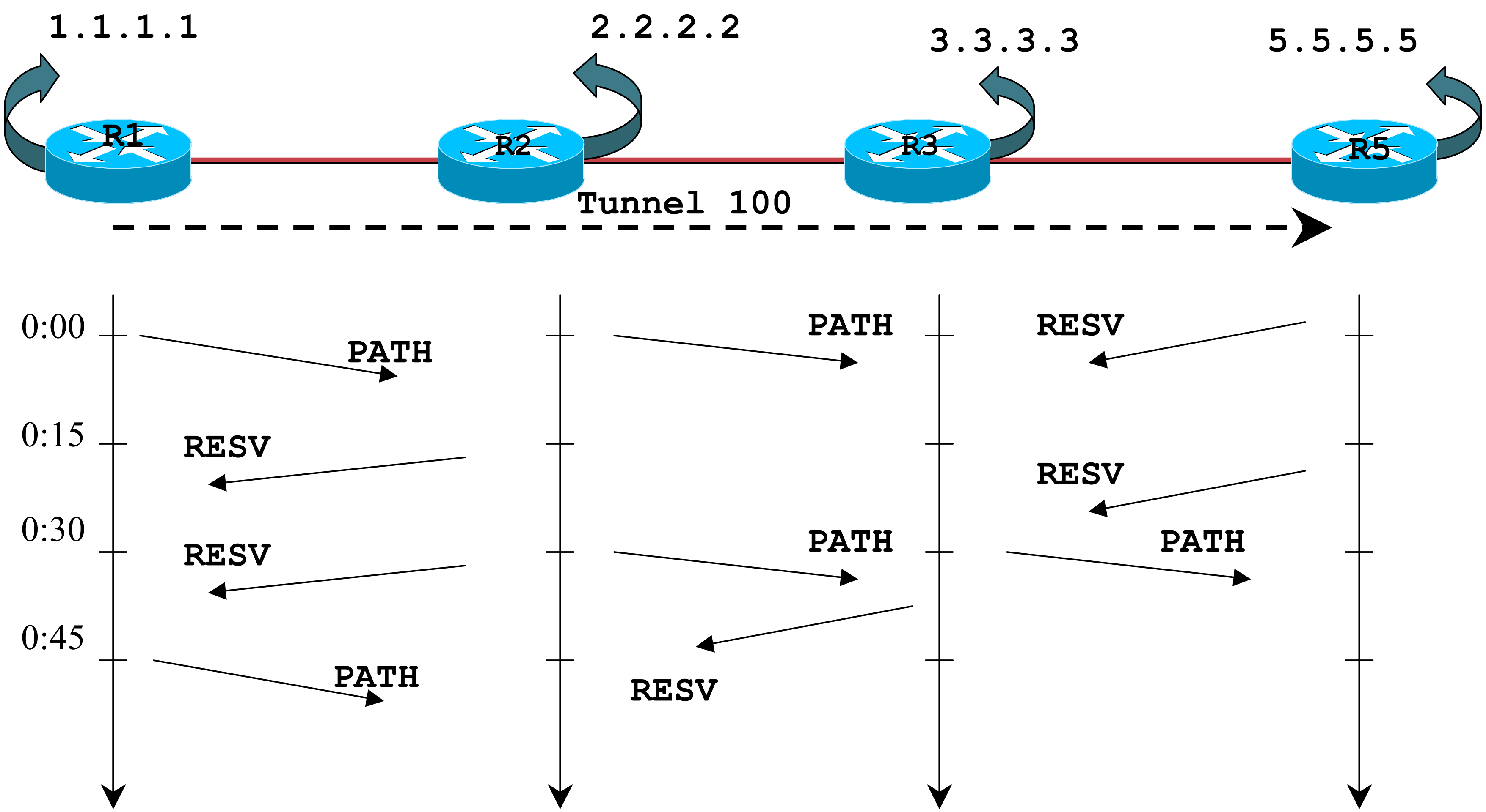
```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td colspan="2">IPv4 tunnel headend address
(32 bits)</td></tr>
<tr><td>Must be zero
(16 bits)</td><td>LSP ID
(16 bits)</td></tr>
</table>

Path Maintenance

- Once the TE tunnel is setup, PATH and RESV messages are used to maintain the tunnel state
- RSVP is a soft-state protocol, relying on PATH & RESV messages for state refresh
- PATH & RESV messages are sent out on average, every 30 seconds
- If we miss 4 consecutive PATH or RESV messages, we consider the RSVP reservation dead

Path Maintenance in action

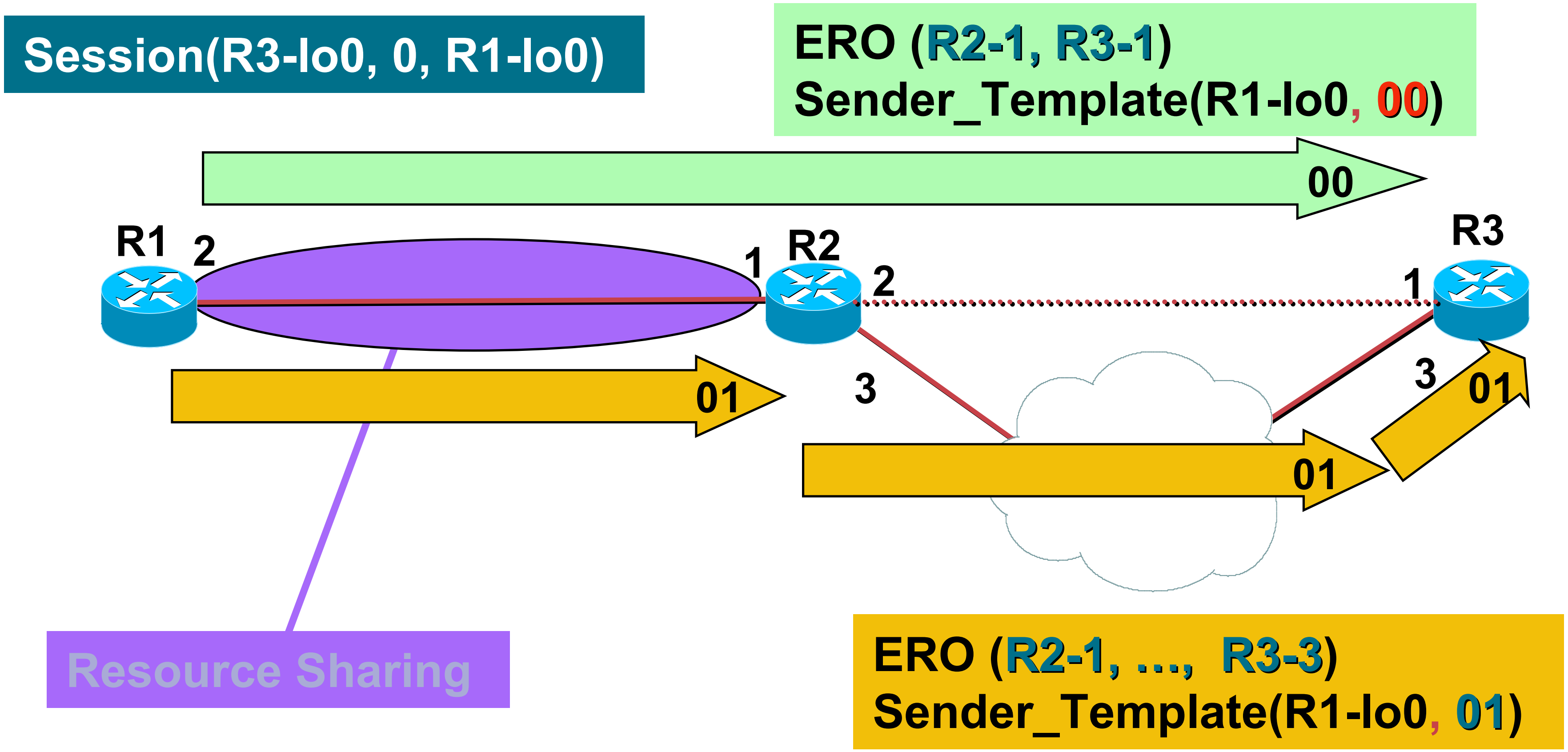


Re-optimization

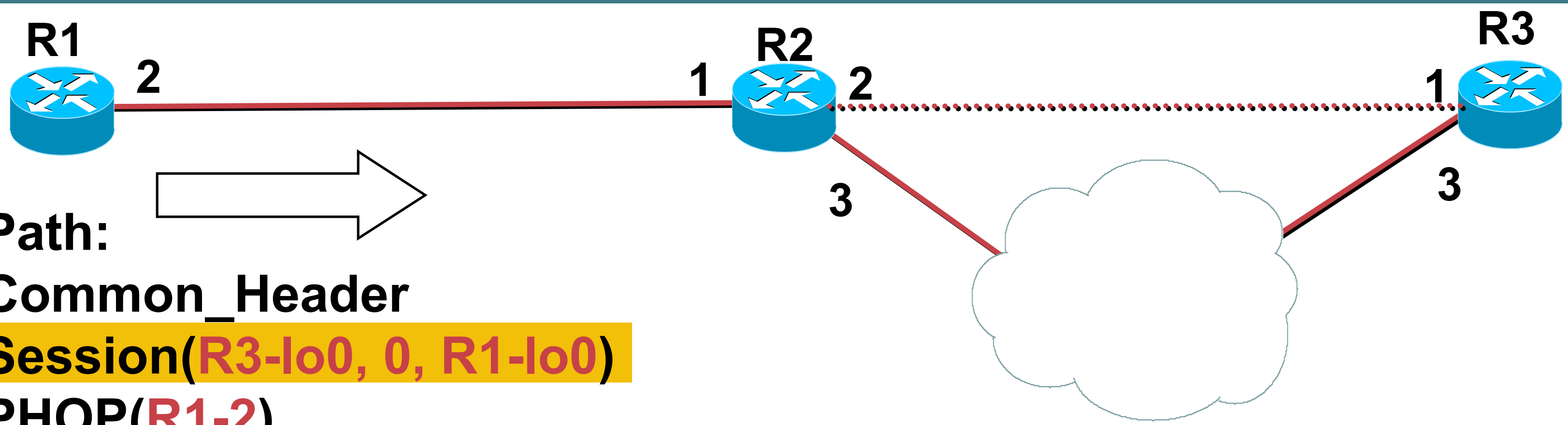
Make-Before-Break objectives

- **Avoid tearing tunnel before the new tunnel instance comes up. This could cause traffic disruption**
- **Avoid double counting bandwidth on the common link carrying the new and the old tunnel**

Make before break in action

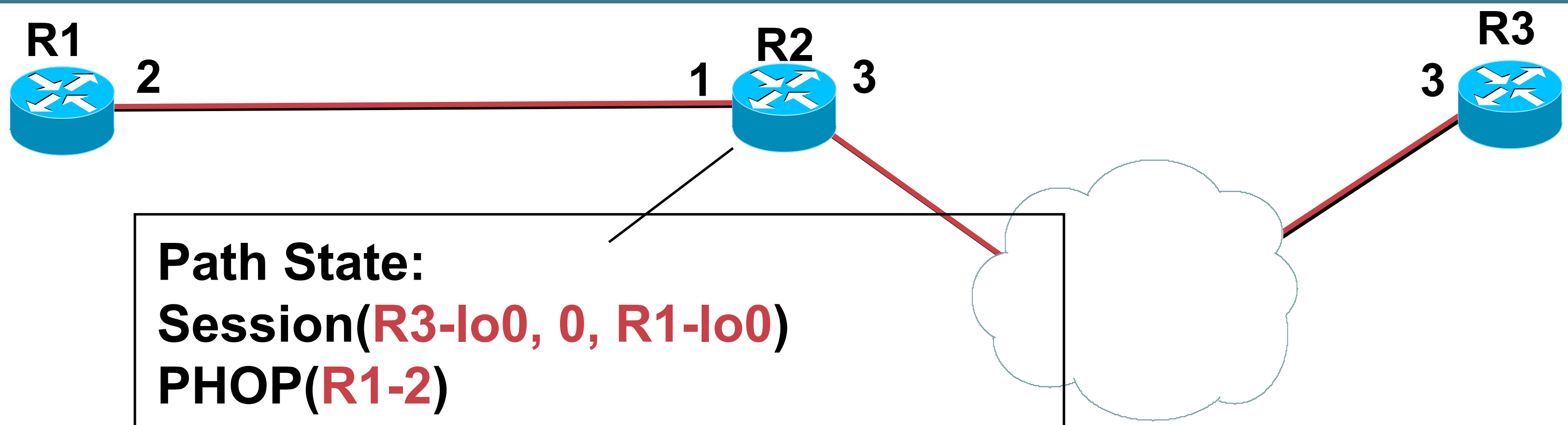


Make before break in action



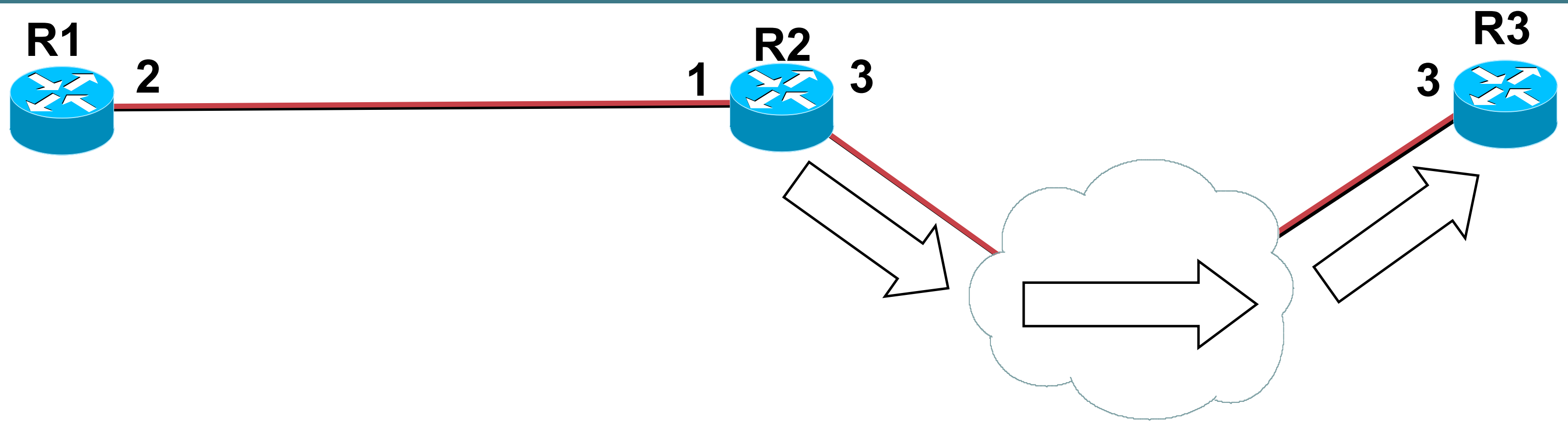
Path:
Common_Header
Session(R3-lo0, 0, R1-lo0)
PHOP(R1-2)
Label_Request(IP)
ERO (R2-1, ….,R3-3)
Session_Attribute (S(3), H(3), 0x04)
Sender_Template(R1-lo0, 01)
Sender_Tspec(3Mbps)
Record_Route(R1-2)

Make before break in action

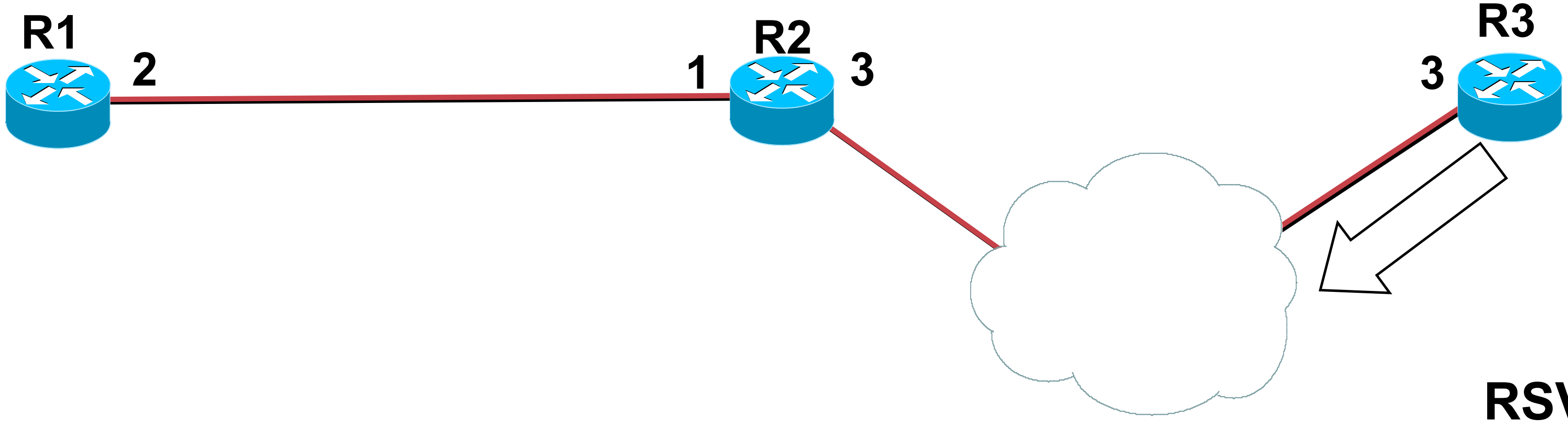


Path State:
Session(R3-lo0, 0, R1-lo0)
PHOP(R1-2)
Label_Request(IP)
ERO (R2-1, ….,R3-3)
Session_Attribute (S(3), H(3), 0x04)
Sender_Template(R1-lo0, 01)
Sender_Tspec(3Mbps)
Record_Route (R1-2)

Make before break in action

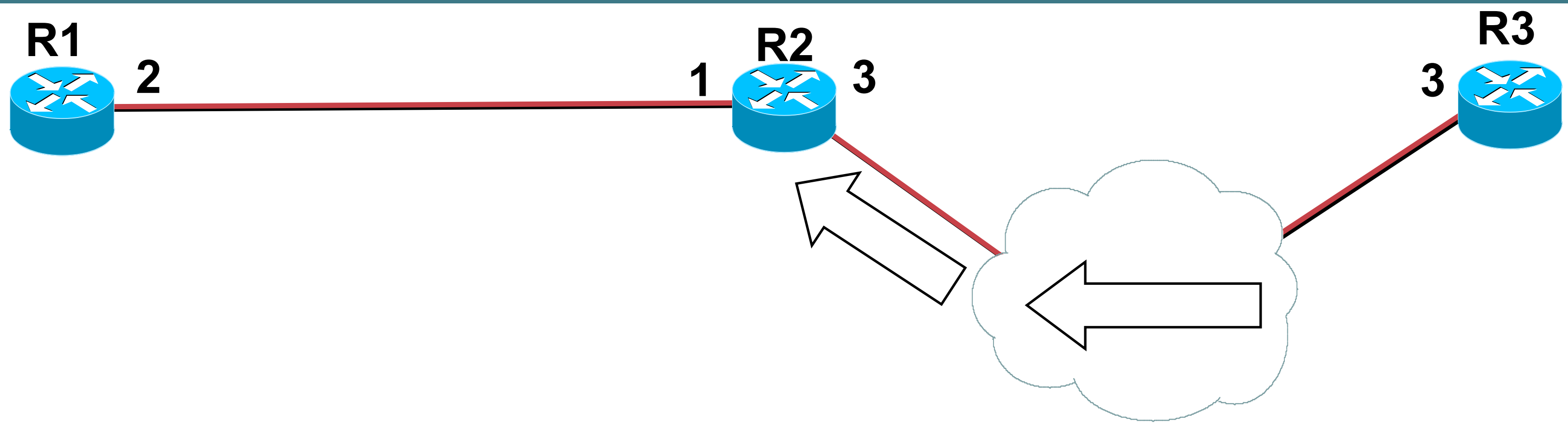


Make before break in action

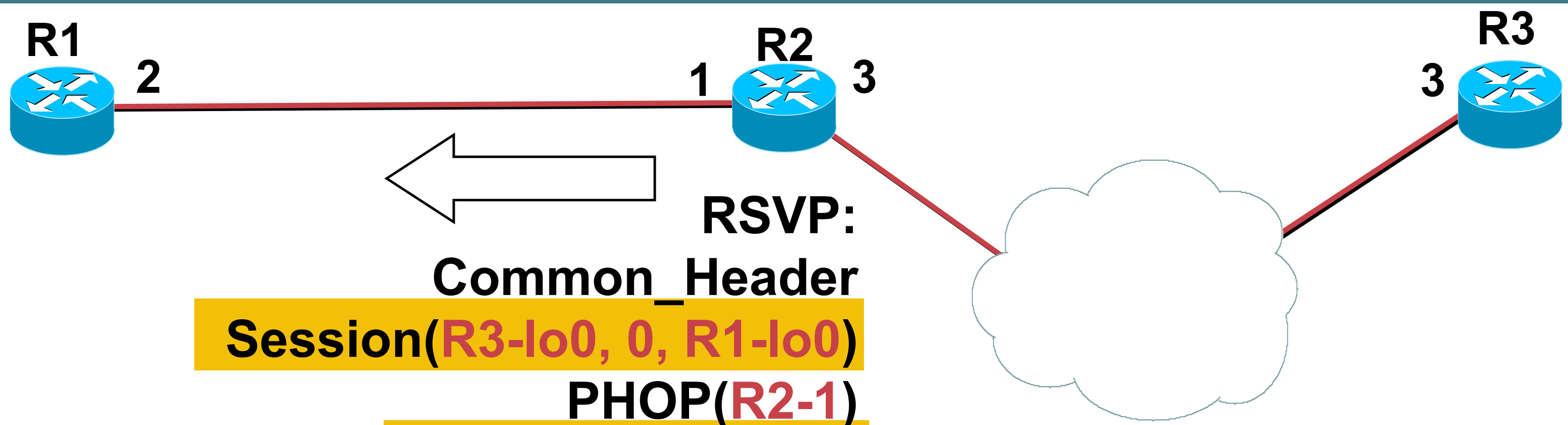


RSVP:
Common_Header
Session(R3-lo0, 0, R1-lo0)
PHOP(R3-3)
Style=SE
FlowSpec(3Mbps)
Sender_Template(R1-lo0, 01)
Label=POP
Record_Route(R3-3)

Make before break in action

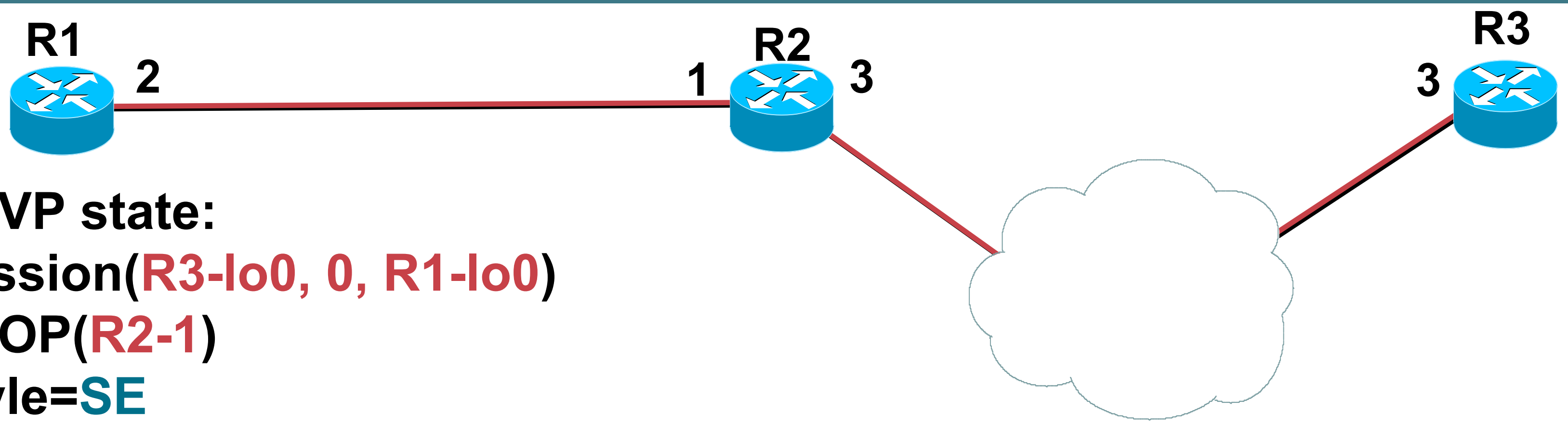


Make before break in action



RSVP:
Common_Header
Session(R3-lo0, 0, R1-lo0)
PHOP(R2-1)
Style=SE
FlowSpec (3Mbps)
Sender_Template(R1-lo0, 01)
Label=6
Record_Route(R2-1, …, R3-3)
Sender_Template(R1-lo0, 00)
Label=5
Record_Route(R2-1, R3-1)

Make before break in action



RSVP state:
Session(R3-lo0, 0, R1-lo0)
PHOP(R2-1)
Style=SE
FlowSpec (3Mbps)
Sender_Template(R1-lo0, 01)
Label=6
Record_Route(R2-1, …, R3-3)
Sender_Template(R1-lo0, 00)
Label=5
Record_Route(R2-1, R3-1)

Re-optimization

- Periodically, a tunnel can rerun PCALC to see if a better path exists to destination.
- Better path will have a lower IGP metric or fewer hops
- If better path exists, headend signals the tunnel via the better path using “make before break”
- Reoptimization happens in the orer of tunnel ID

Re-optimization Triggers

- **Periodic:** by default triggered every 3600 seconds (or CLI configured period) for all TE tunnels in the

  order of priority (0 thru 7)

  within each priority based on the tunnel ID

```
mpls traffic-eng reoptimize timers frequency <1-604800 sec>
```

- **Event triggered:** event such as a link coming up will trigger reoptimization

- **Manual:** reoptimize one or all tunnels at the command prompt

```
mpls traffic-eng reoptimize   (all tunnels)
mpls traffic-eng reoptimize Tunnel <0-2147483647>  (per tunnel)
```

Disabling Re-optimization

- **One or all tunnels can be disabled for reoptimization if we think that the tunnel does not need reoptimization**

```
mpls traffic-eng reoptimize timers frequency 0 (disables all
   tunnels)


interface tunnel0

tunnel mpls traffic-eng path-option 1 dynamic lockdown (disable
   tunnel0)
```

MPLS-TE: traffic aspects of TE tunnels

Agenda

- Mapping Traffic to Paths
- Using metrics with tunnels
- Load balancing with TE tunnels
- Monitoring traffic with TE tunnels

Mapping Traffic to Path

Routing Traffic Down a Tunnel

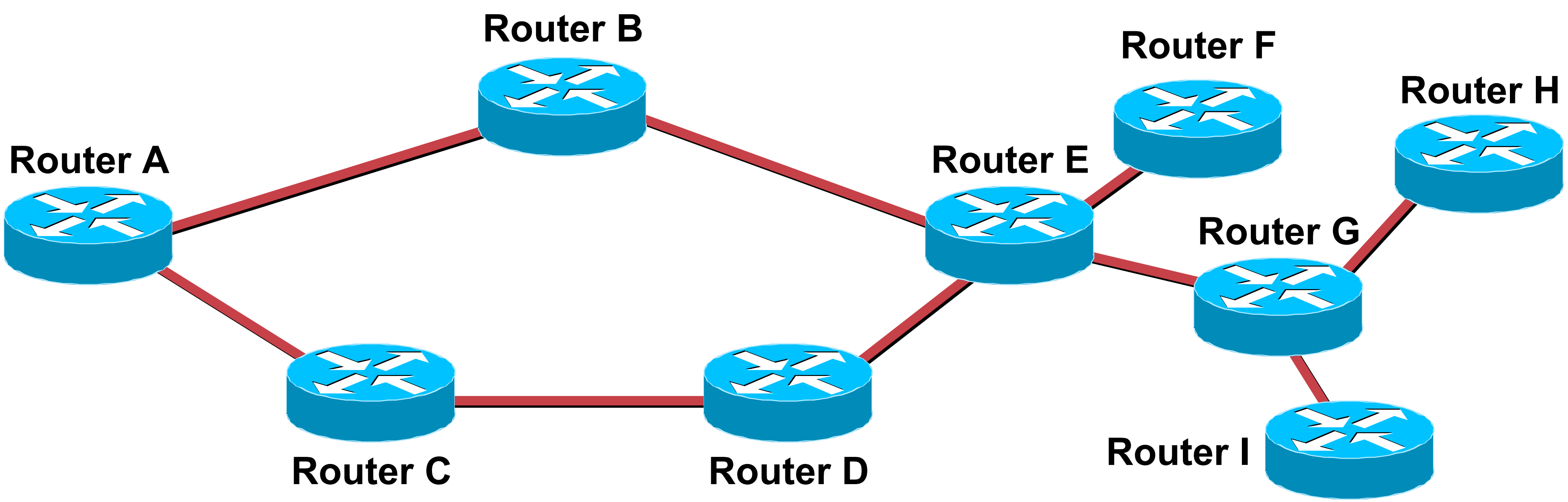
- Once RESV reaches headend, tunnel interface comes up
- How to get traffic down the tunnel?
  1. Autoroute
  2. Forwarding adjacency
  3. Static routes
  4. Policy routing

Autoroute

- **Tunnel is treated as a directly connected link to the tail**
- **IGP adjacency is NOT run over the tunnel!**
  - **Unlike an ATM/FR VC**
- **Autoroute limited to single area/level only**

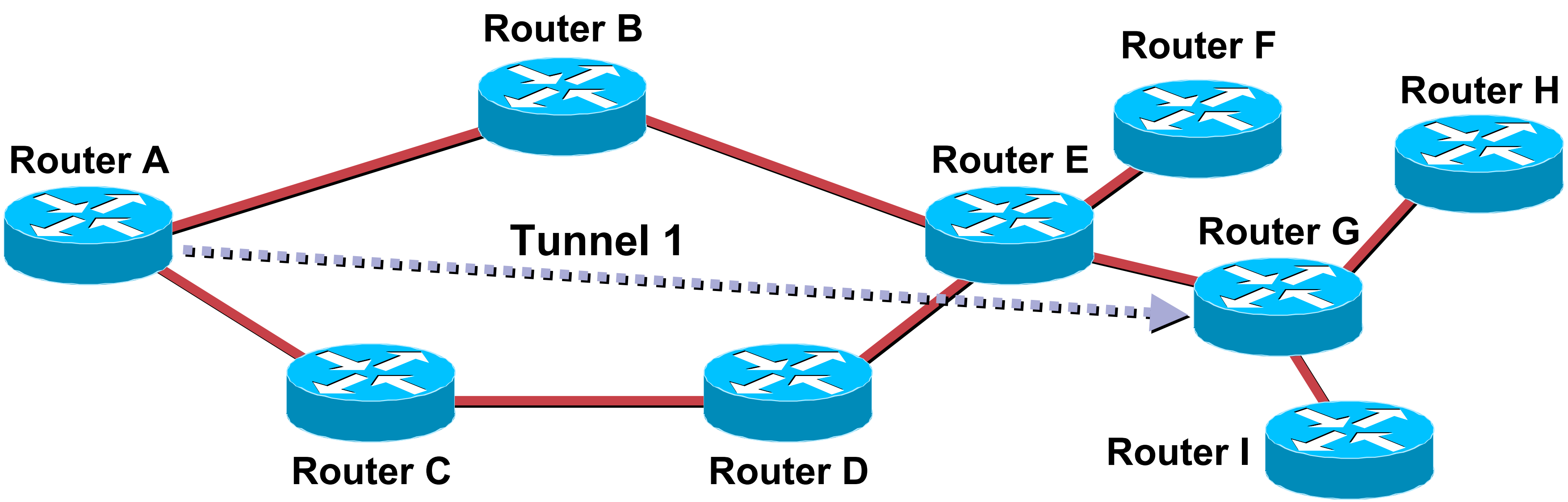
Autoroute

This Is the Physical Topology



Autoroute

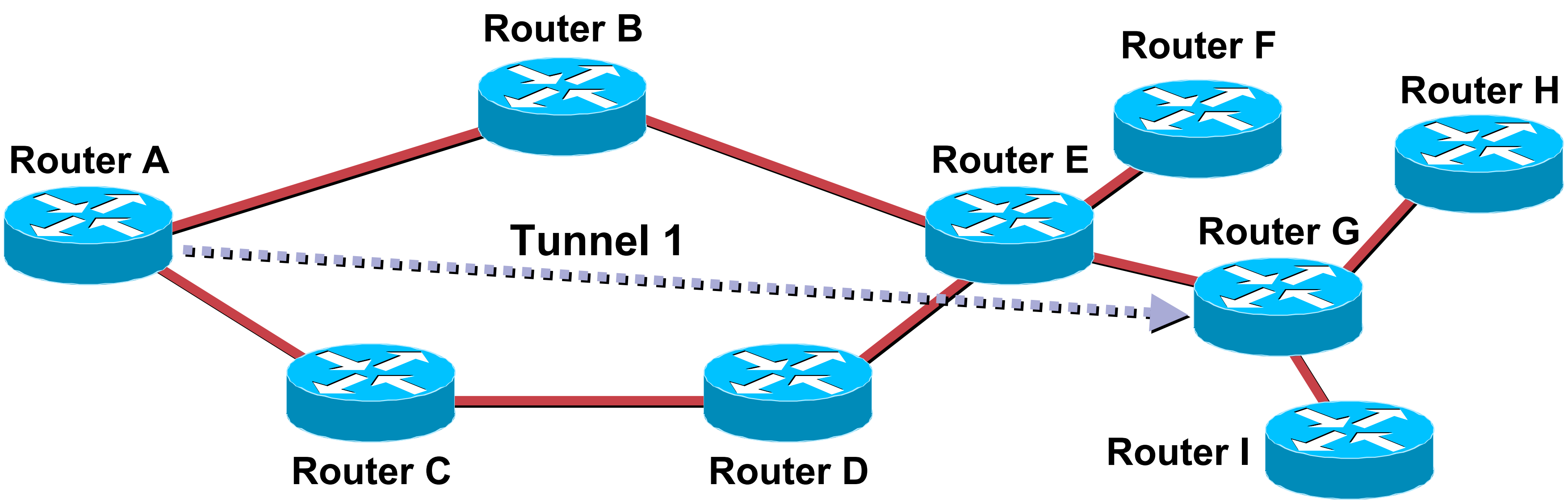
- **This is Router A's logical topology**
- **By default, other routers don't see the tunnel!**



Autoroute

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	Tunnel 1	30
H	Tunnel 1	40
I	Tunnel 1	40

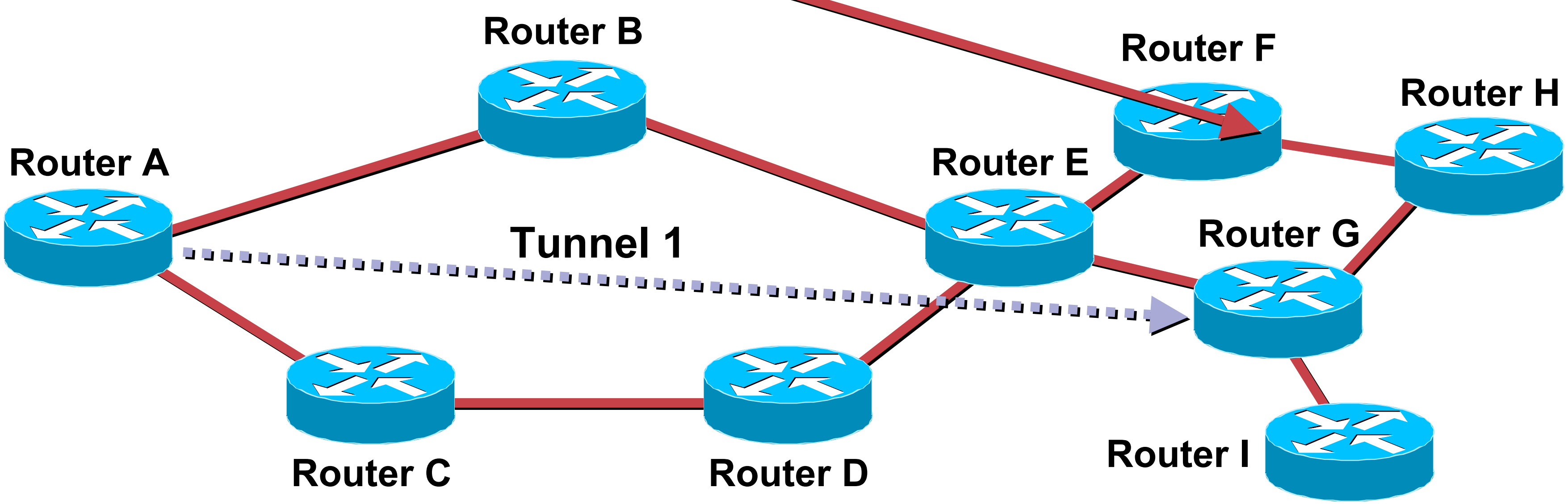
- Router A's routing table, built via auto-route
- Everything "behind" the tunnel is routed via the tunnel



Autoroute

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	Tunnel 1	30
H	Tunnel 1 & B	40
I	Tunnel 1	40

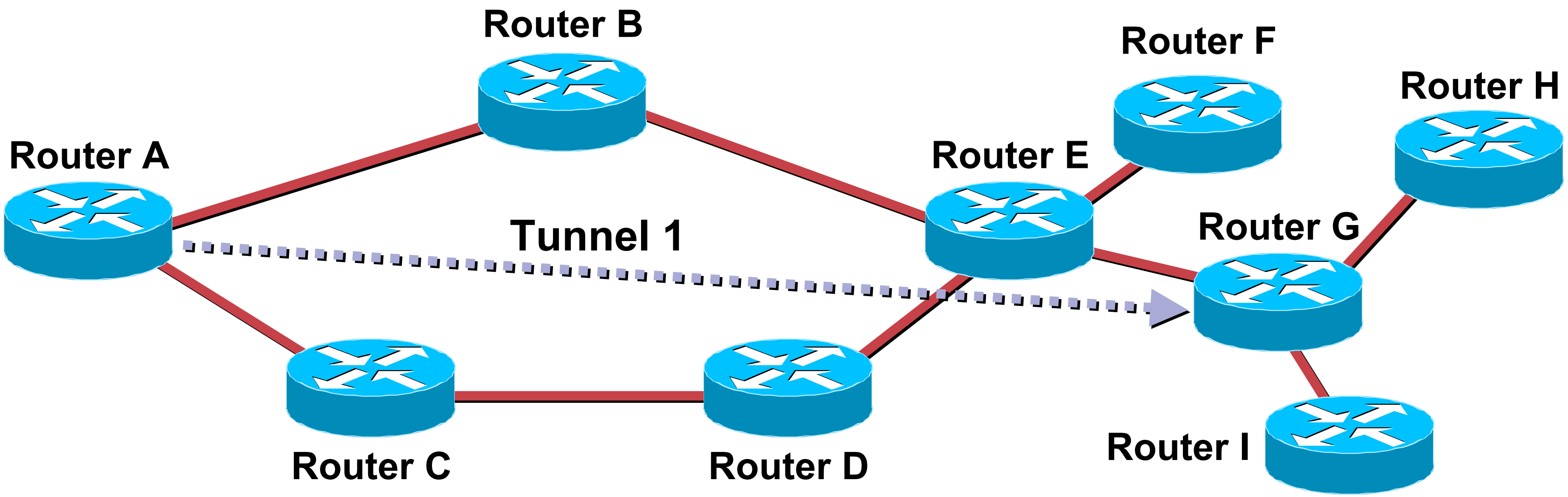
- **If there was a link from F to H, Router A would have 2 paths to H (A->G->H and A->B->E->F->H)**
- **Nothing else changes**



Autoroute

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	Tunnel 1	30
H	Tunnel 1	40
I	Tunnel 1	40

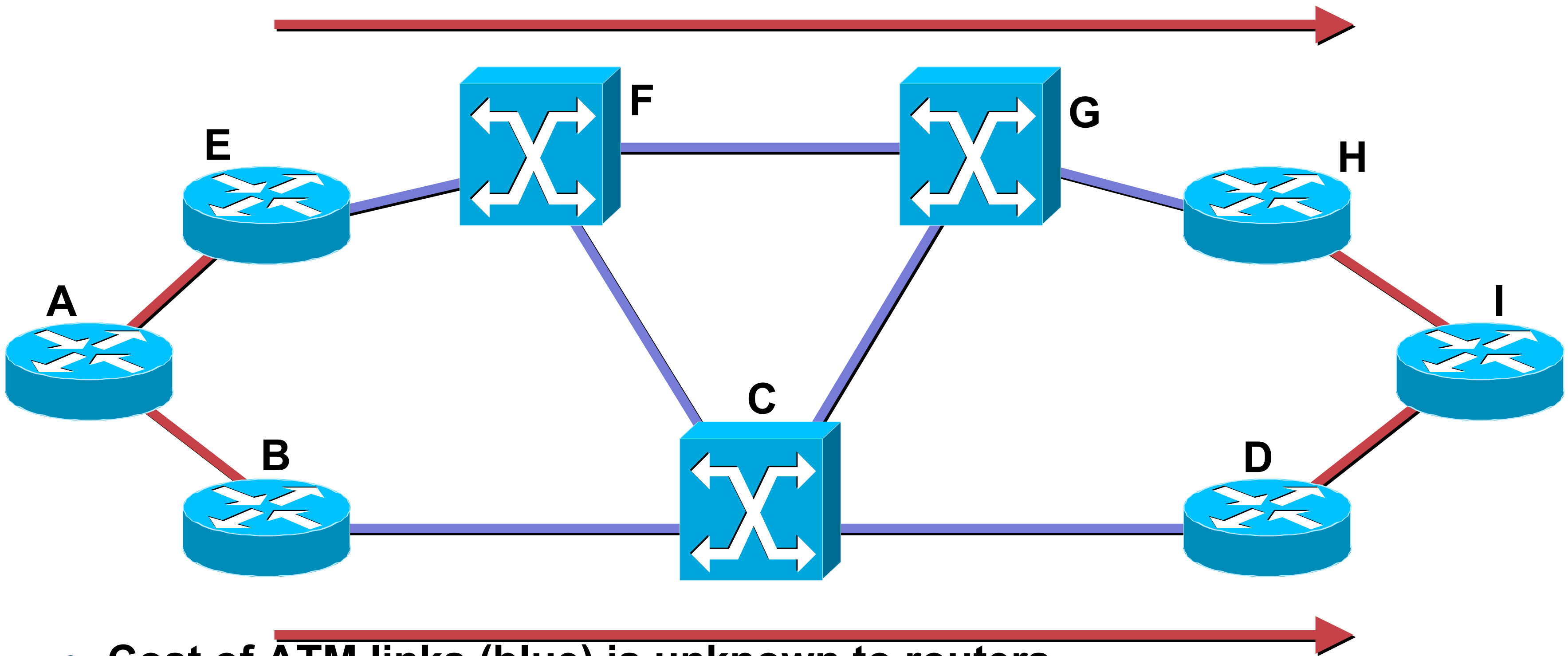
```
interface Tunnel1
 tunnel mpls traffic-eng autoroute announce
```



Forwarding Adjacency

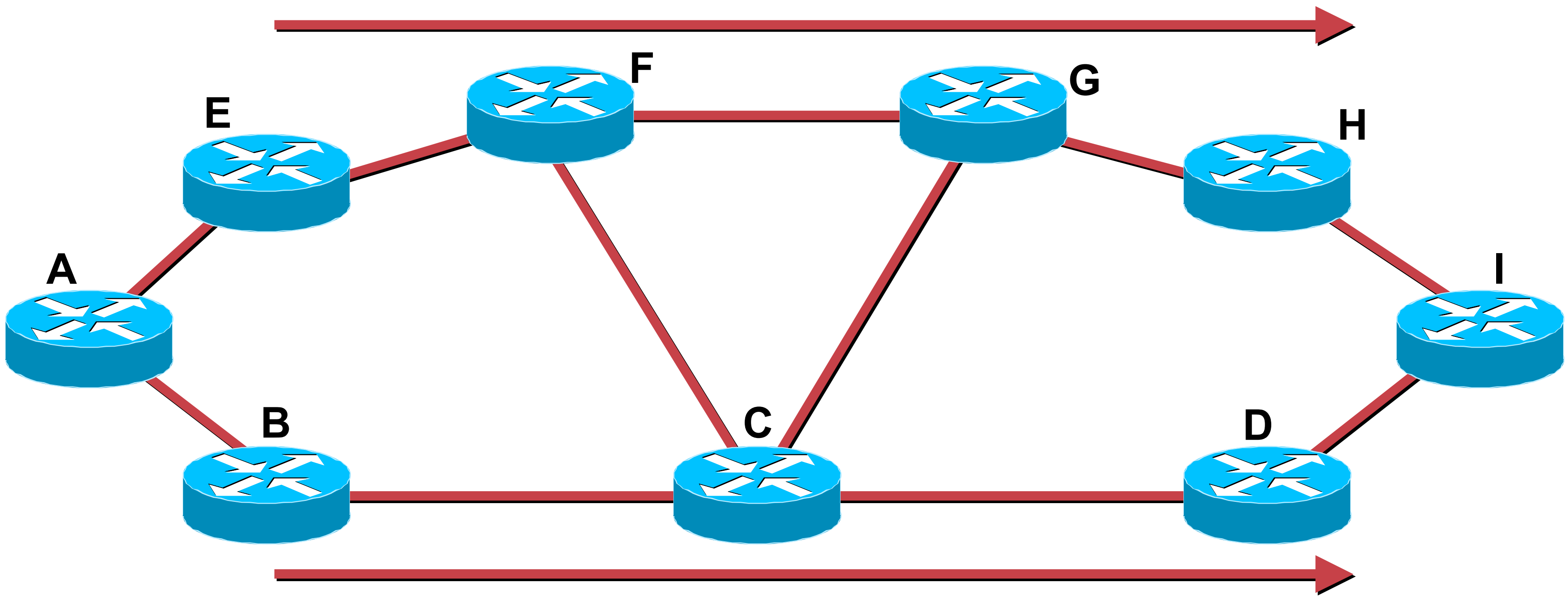
- **With autoroute, the LSP is not advertised into the IGP**
- **This is the right behavior if you’re adding TE to an IP network, but maybe not if you’re migrating from ATM/FR to TE**
- **Sometimes advertising the LSP into the IGP as a link is necessary to preserve the routing outside the ATM/FR cloud**

ATM Model



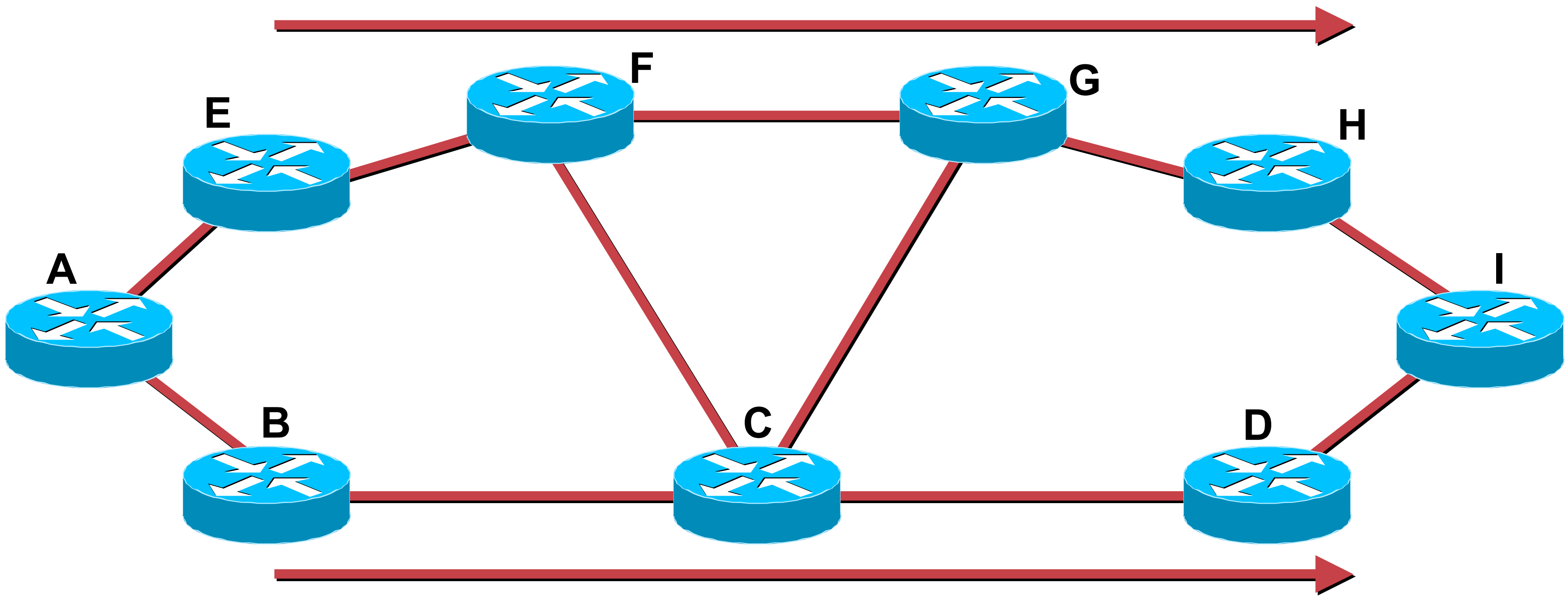
- Cost of ATM links (blue) is unknown to routers
- A sees two links in IGP—E->H and B->D
- A can load-share between B and E

Before FA



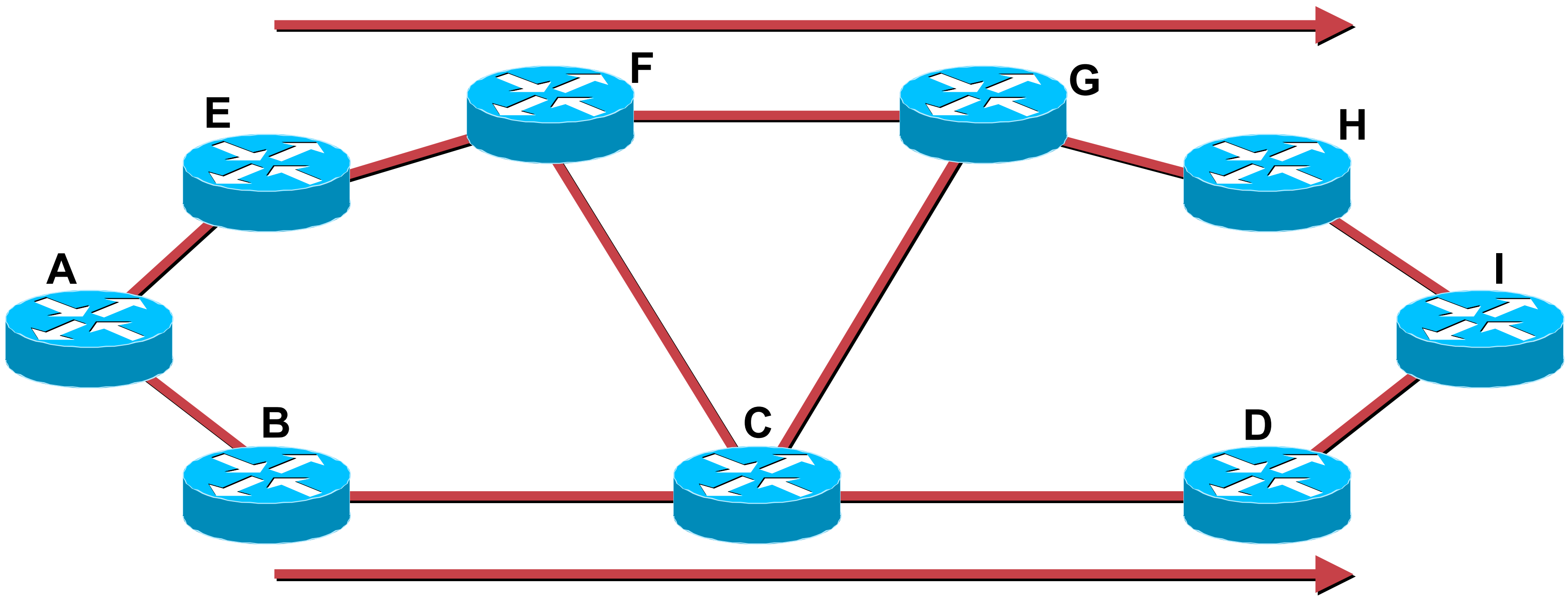
- All links have cost of 10
- A's shortest path to I is A->B->C->D->I
- A doesn't see TE tunnels on {E,B}, alternate path never gets used!
- Changing link costs is undesirable, can have strange adverse effects

FA Advertises TE Tunnels in the IGP



- With forwarding-adjacency, A can see the TE tunnels as links
- A can then send traffic across both paths
- This is desirable in some topologies (looks just like ATM did, same methodologies can be applied)

FA Advertises TE Tunnels in the IGP



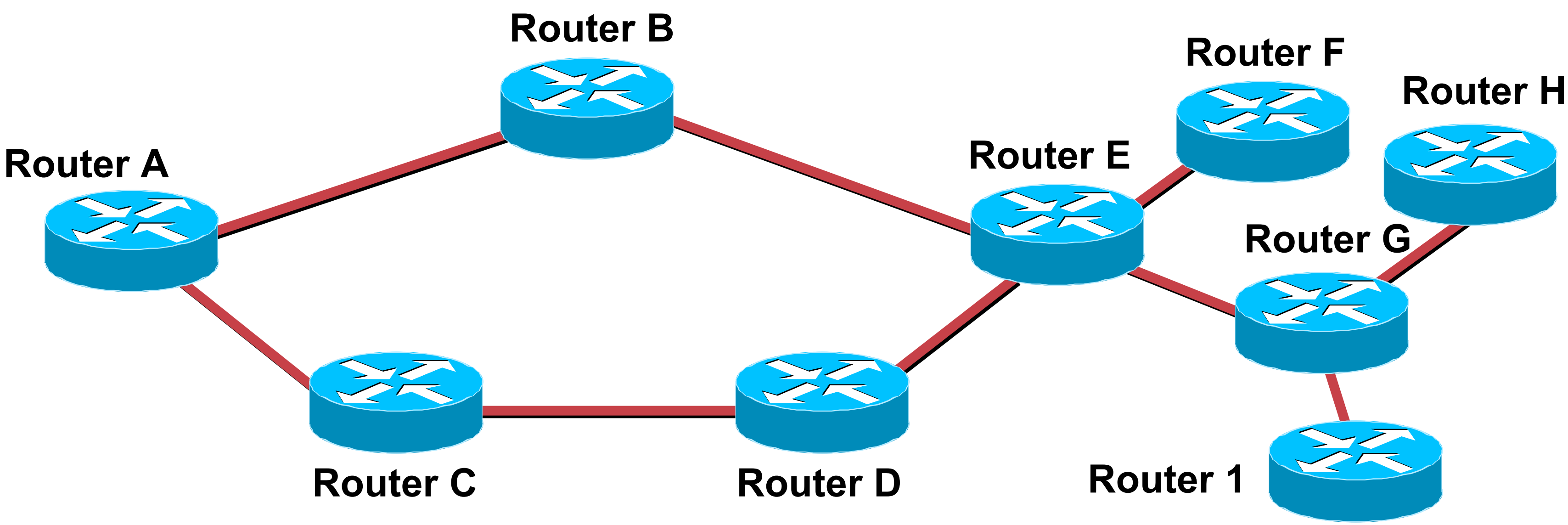

```
tunnel mpls traffic-eng forwarding-adjacency
isis metric <x> level-<y>
```

OR

```
ip ospf cost <x>
```

Static Routing

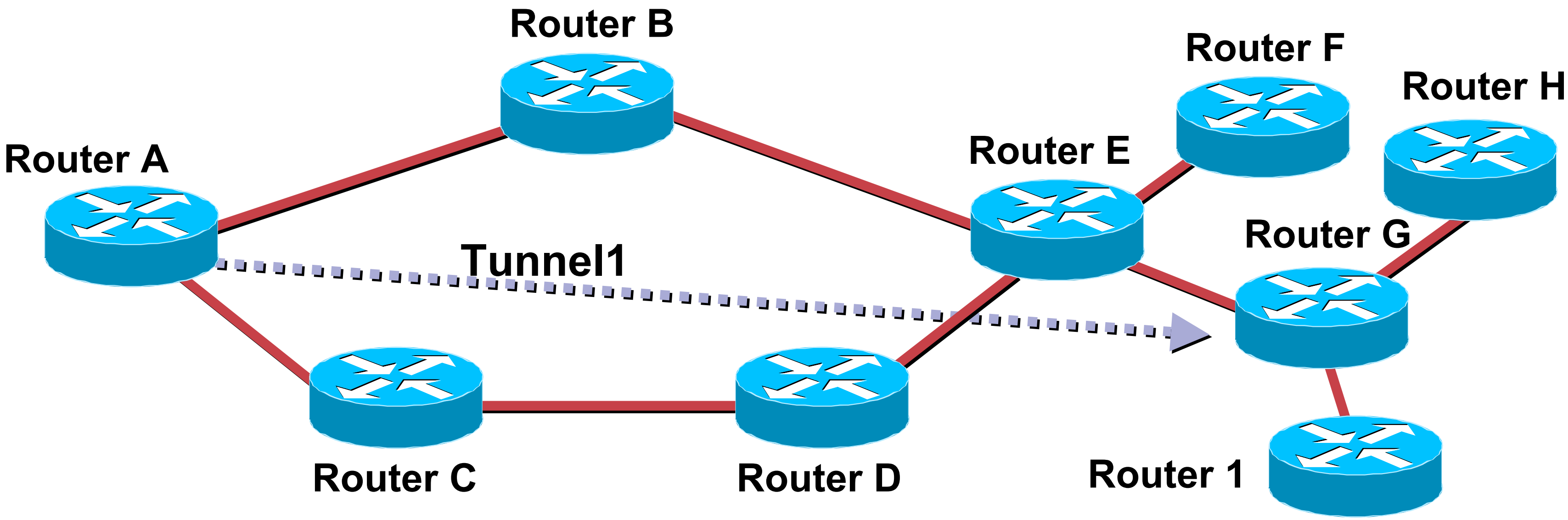
```
RtrA(config)#ip route H.H.H.H 255.255.255.255 Tunnel1
```



Static Routing

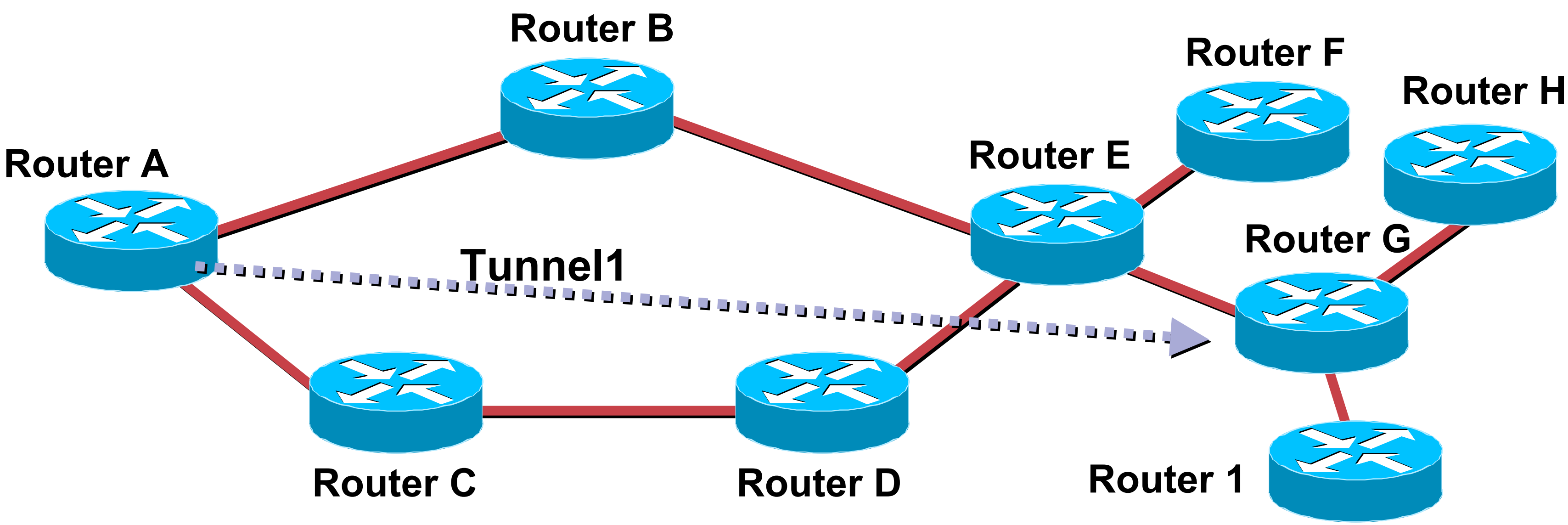
Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30
H	Tunnel 1	40
I	B	40

- Router H is known via the tunnel
- Router G is not routed to over the tunnel, even though it's the tunnel tail!



Policy Routing

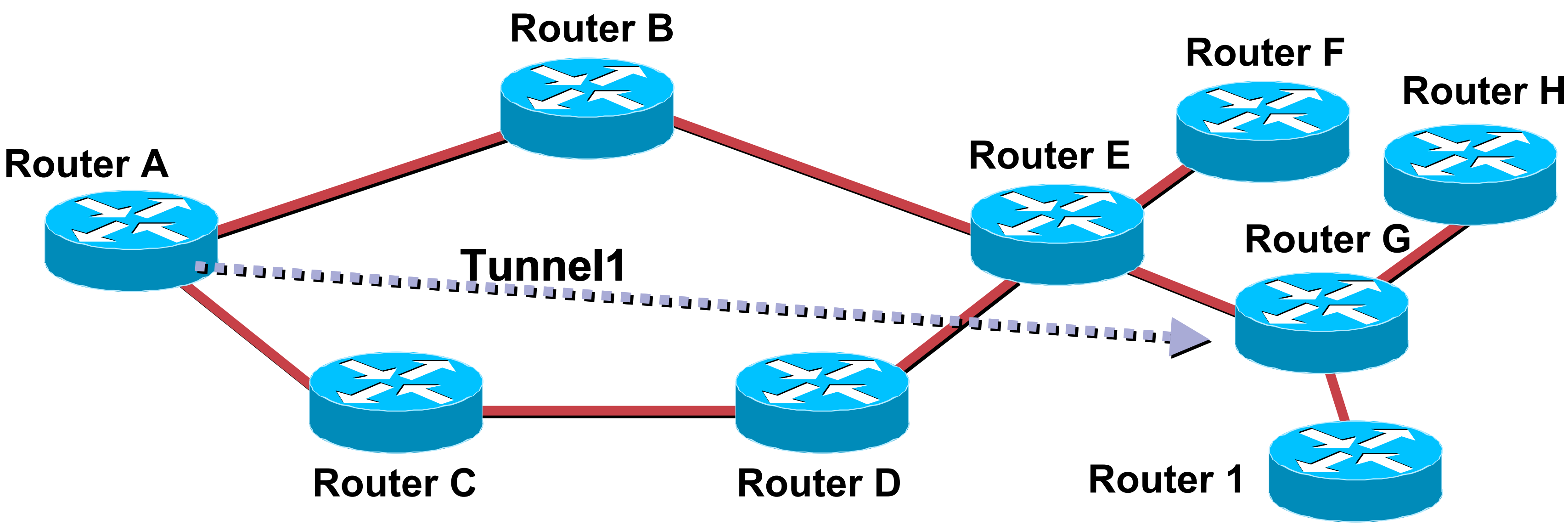
```
RtrA(config-if)#ip policy route-map set-tunnel
RtrA(config)#route-map set-tunnel
RtrA(config-route-map)#match ip address 101
RtrA(config-route-map)#set interface Tunnel1
```



Policy Routing

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30
H	B	40
I	B	40

- Routing table isn't affected by policy routing



Enhancement to SPF - metric check

Tunnel metric:

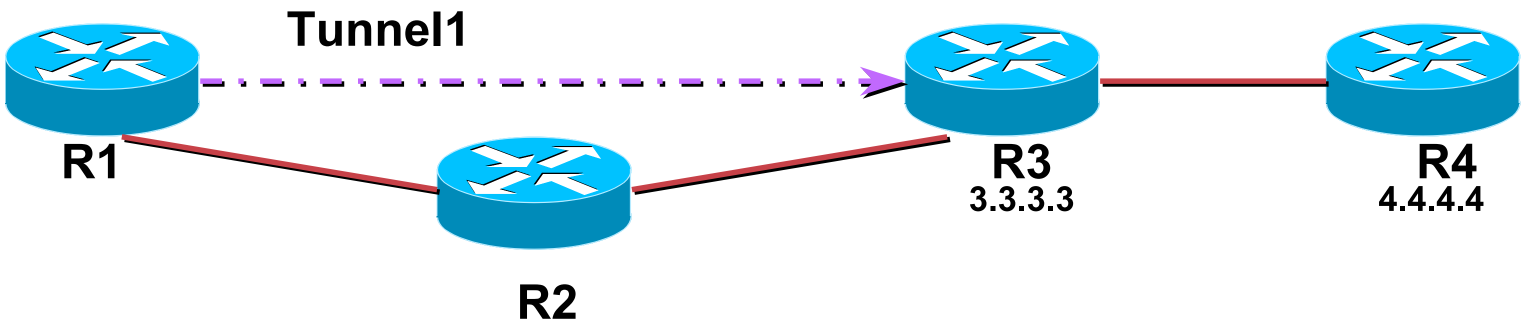
A. Relative +/- X
B. Absolute Y (only for ISIS)
C. Fixed Z

Example:

Metric of native IP path to the found node = 50

1. Tunnel with relative metric of -10 => 40
2. Tunnel with relative metric of +10 => 60
3. Tunnel with absolute metric of 10 => 10

Absolute/Relative/Fixed Metric in action

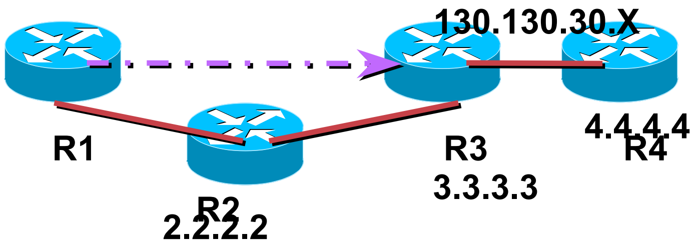


Routing Table on R1 (with all link metrics=10)

IP Addr	Cost	Next-Hop	Interface
4.4.4.4	30	3.3.3.3	Tunnel1
3.3.3.3	20	3.3.3.3	Tunnel1

Relative Metric in action

Metric to the tunnel tailend is the same **"Relative metric".** Anything downstream to the tunnel tail is added to the relative metric



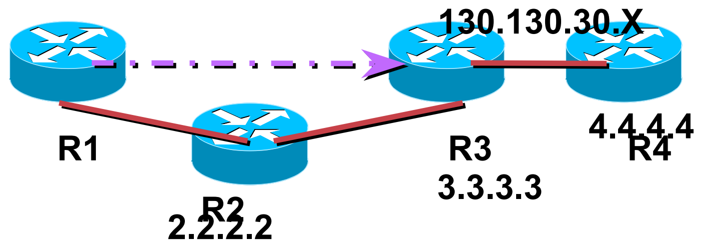

```
R1(config-if)#interface tunnel1
R1(config-if)#tunnel mpls traffic-eng autoroute metric relative -5
```

Routing Table on R1

IP Addr	Cost	Next-Hop	Interface
4.4.4.4	25	3.3.3.3	Tunnel1
3.3.3.3	15	3.3.3.3	Tunnel1

Fixed Metric in action

Metric to the tunnel tailend is the same **“Fixed metric”.** Anything downstream to the tunnel tail is added to the fixed metric



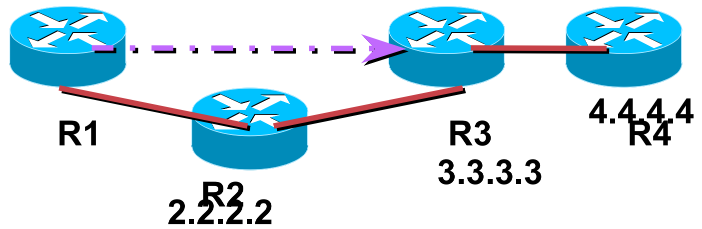

```
R1(config-if)#interface tunnel1
R1(config-if)#tunnel mpls traffic-eng autoroute metric 5
```

Routing Table on R1

IP Addr	Cost	Next-Hop	Interface
4.4.4.4	15	3.3.3.3	Tunnel1
3.3.3.3	5	3.3.3.3	Tunnel1

Absolute Metric in action

Metric to the tunnel tailend and downstream destinations is the same “**Absolute metric**” value




```
R1(config-if)#interface tunnel1
R1(config-if)#tunnel mpls traffic-eng autoroute metric absolute
  2
```

Routing Table on R1

IP Addr	Cost	Next-Hop	Interface
4.4.4.4	2	3.3.3.3	Tunnel1
3.3.3.3	2	3.3.3.3	Tunnel1

Load Sharing with TE tunnels

Unequal Cost Load Balancing

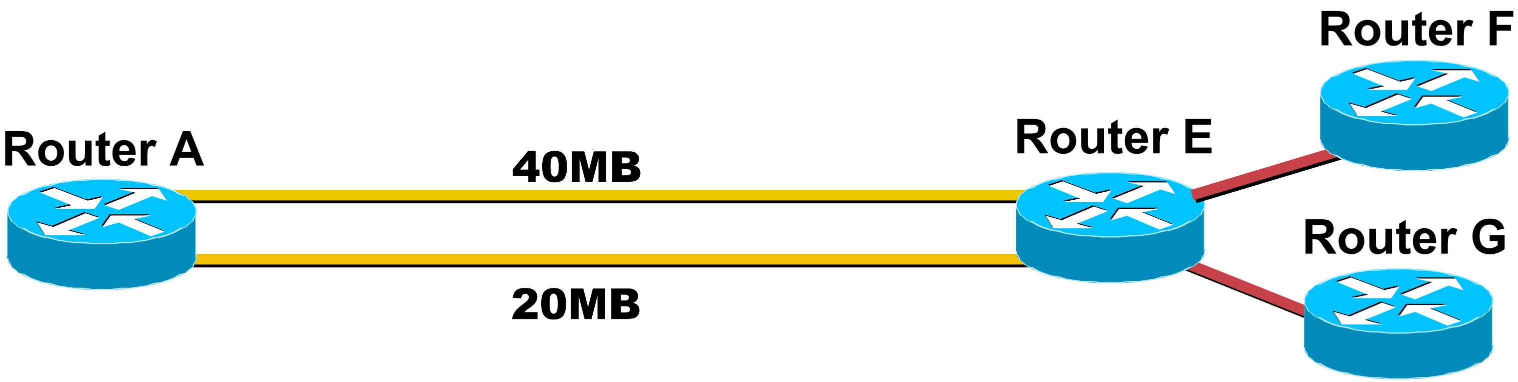
- **IP routing has equal-cost load balancing, but not unequal cost***

***EIGRP Has 'Variance', but That's Not as Flexible**

Unequal Cost Load Balancing

- **A TE tunnel does not load share traffic between itself and the native IP path it takes**
- **Multiple parallel tunnels can load share traffic based on bandwidth. This can be equal or unequal cost load balancing**
- **TE tunnels and native IP links can load share traffic, provided the destination is downstream to the tunnel destination. In this case load sharing is equal cost**

Unequal Cost Example




```
gsr1#show ip route 192.168.1.8
Routing entry for 192.168.1.8/32
  Known via "isis", distance 115, metric 83, type level-2
  Redistributing via isis
  Last update from 192.168.1.8 on Tunnel0, 00:00:21 ago
  Routing Descriptor Blocks:
  * 192.168.1.8, from 192.168.1.8, via Tunnel0
      Route metric is 83, traffic share count is 2
    192.168.1.8, from 192.168.1.8, via Tunnel1
      Route metric is 83, traffic share count is 1
```

Monitoring Traffic in TE tunnels

Monitoring Traffic in TE tunnels

- TE tunnels do not police traffic. This means that we could send 10 Gbps of traffic via a 10 Mbps tunnel.
- No automatic correlation between tunnel bandwidth and real traffic thru tunnel
- Auto Bandwidth enables a tunnel to adjust bandwidth based on traffic flow

Auto Bandwidth

- **Tunnel monitors traffic say every 5 minutes and records the largest sample. At the end of 24 hour period, the tunnel applies the largest sample to its bandwidth statement in the configuration**
- **We can also define a floor and ceiling to bandwidth beyond which no change will be applied to bandwidth statement**

Enabling Auto-Bandwidth

```
mpls traffic-eng auto-bw timers frequency
<0-604800>
```

- **Global command**
- **Enables tunnels to sample load at the configured frequency**
- **Should not be less than the “load interval” on the interface**

Enabling Auto-Bandwidth

```
tunnel mpls traffic-eng auto-bw ?
  collect-bw Just collect Bandwidth info on this tunnel
  frequency  Frequency to change tunnel BW
  max-bw     Set the Maximum Bandwidth for auto-bw on this tunnel
  min-bw     Set the Minimum Bandwidth for auto-bw on this tunnel
  <cr>
```

- **Per-tunnel command**
- **Periodically changes tunnel BW reservation based on traffic out tunnel**
- **Timers are tunable to make auto-bandwidth more or less sensitive**
    - **Tradeoff: Quicker reaction versus more churn**

MPLS-TE: Advanced TE topics

Agenda

- **MPLS-TE Rerouting**
- **Fast Reroute (Link, Node and Path)**
- **Inter-area/Inter-AS TE**

MPLS TE rerouting

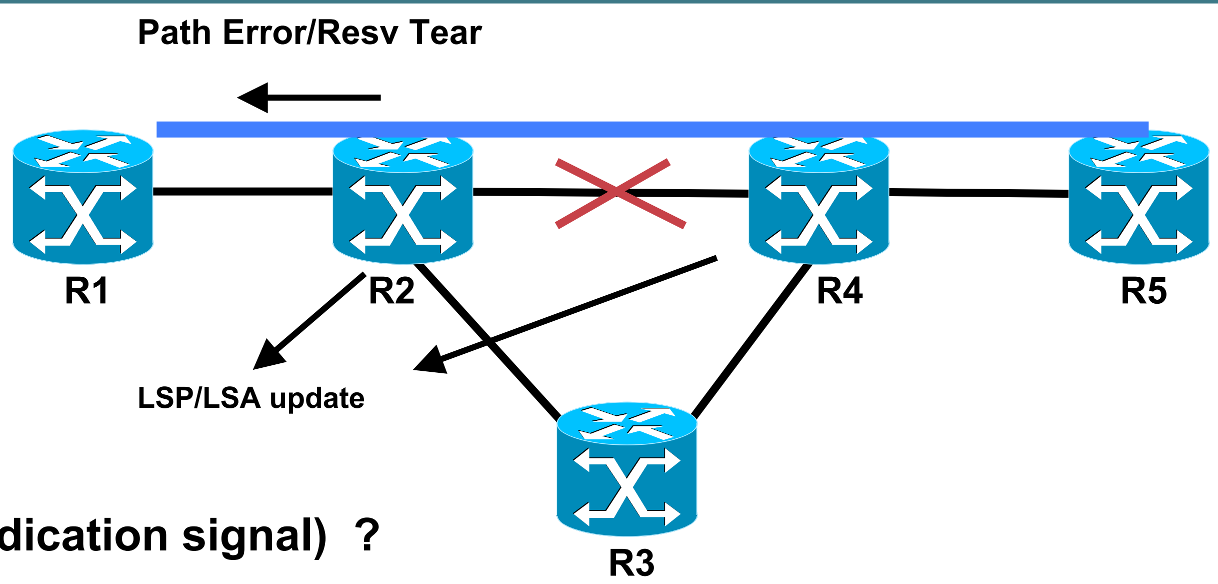
LSP rerouting

- **Controlled by the head-end of a trunk via the resilience attribute of the trunk**
- **Fallback to either (pre)configured or dynamically computed path. Preferably last path option should be dynamic**

```
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.0.1.102
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 3 3
 tunnel mpls traffic-eng bandwidth 10000
 tunnel mpls traffic-eng path-option 1 explicit name prim_path
 tunnel mpls traffic-eng path-option 2 dynamic
```

```
ip explicit-path name prim_path enable
 next-address 10.0.1.123
 next-address 10.0.1.100
```

MPLS TE rerouting



- **The FIS (failure indication signal) ?**

  *** R1 may receive a Path Error from R2 and a Resv Tear OR**

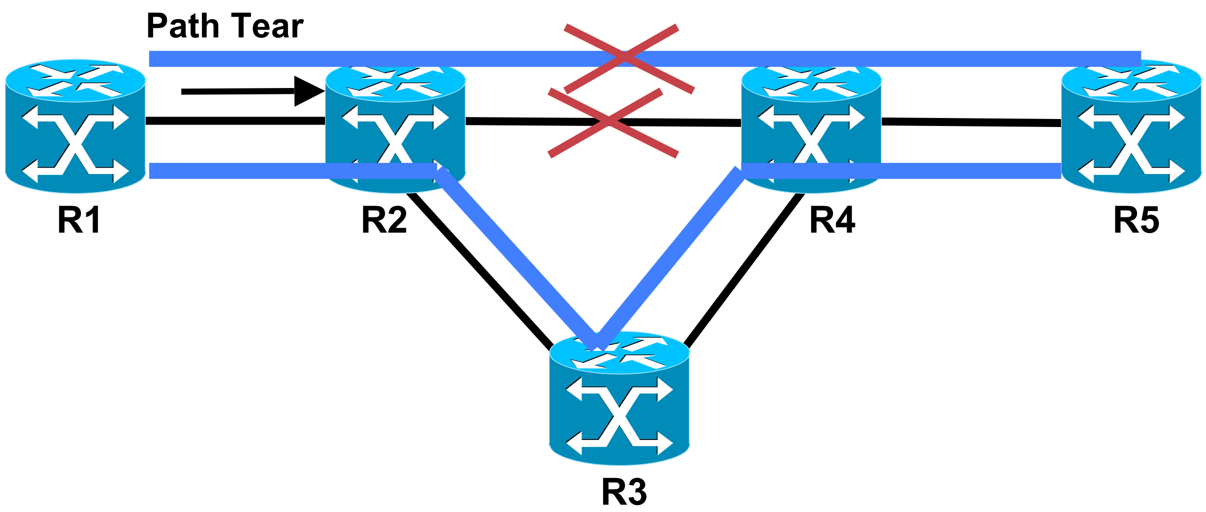
  *** R1 will receive a new LSA/LSP indicating the R2-R4 is down and will conclude the LSP has failed**

  Which one on those two events will happen first ? It depends of the failure type and IGP tuning

- **Receipt of Path Error allows to remove the failed link from the TE database to prevent to retry the same failed link (if the IGP update has not been received yet)**

MPLS TE rerouting

- R1 is now informed that the LSP has suffered a failure



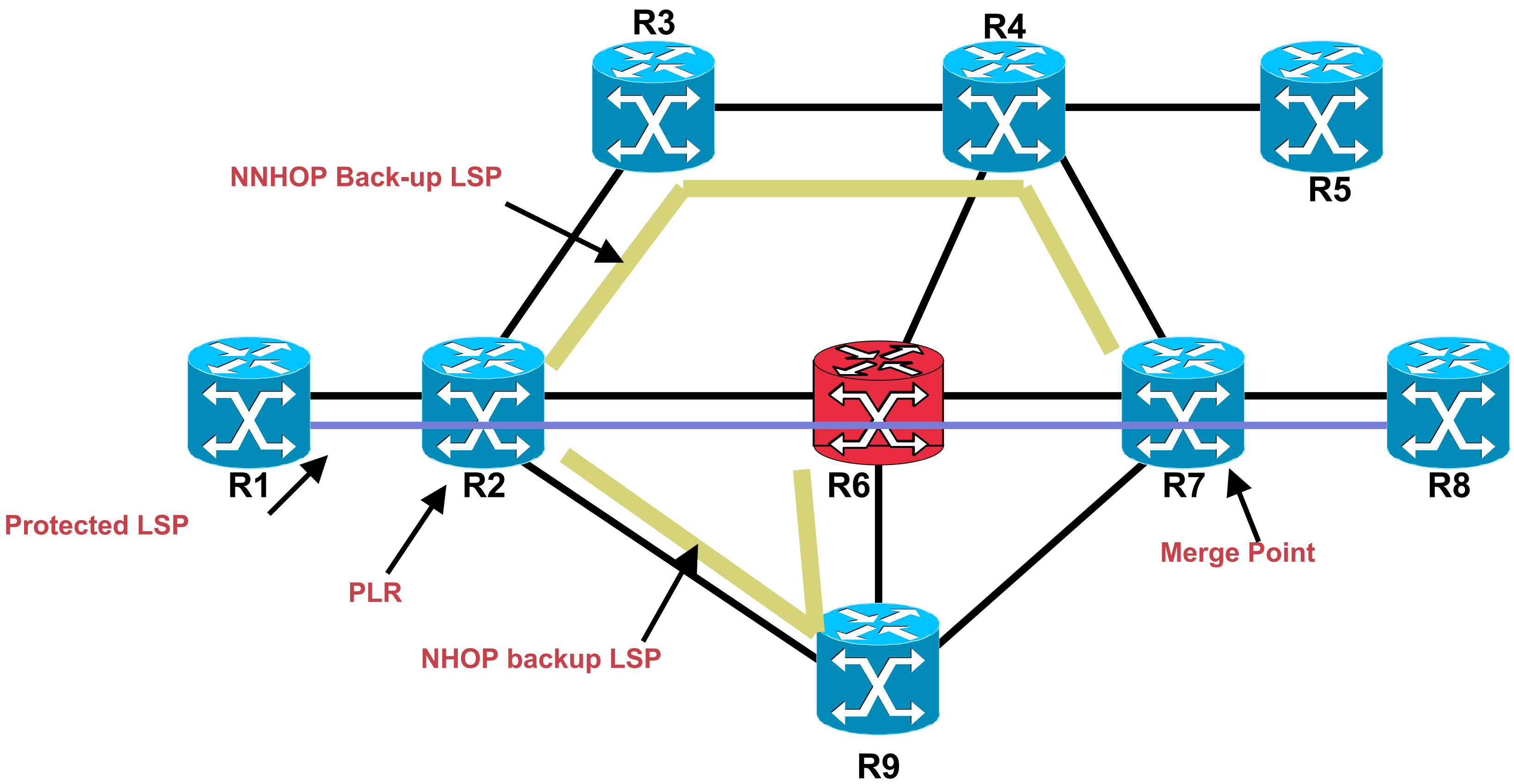
- R1 clear the Path state with an RSVP Path Tear message
- R1 recalculates a new Path for the Tunnel and will signal the new tunnel. If no Path available, R1 will continuously retry to find a new path (local process)

Convergence = O(secs)

Fast ReRoute

- **FRR builds a path to be used in case of a failure in the network**
- **Minimize packet loss by taking a quick local decision to reroute at the failure point**

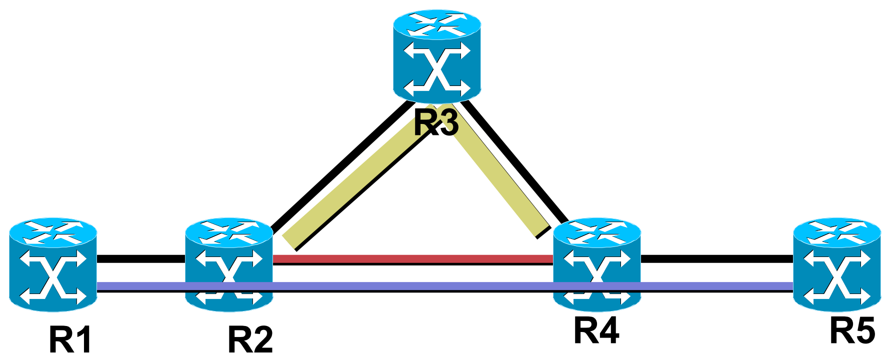
Terminology

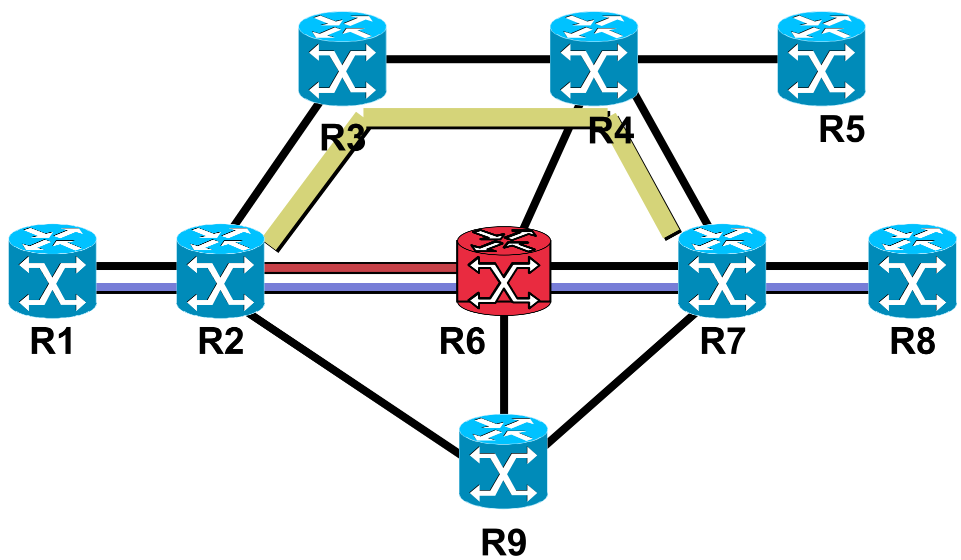


Fast ReRoute

MPLS Fast Reroute Local Repair

- **Link protection**: the backup tunnel tail-end (MP) is one hop away from the PLR





- **Node protection**: the backup tunnel tail-end (MP) is two hops away from the PLR

IP Failure Recovery

For IP to Recover From a Failure, Several Things Need to Happen:

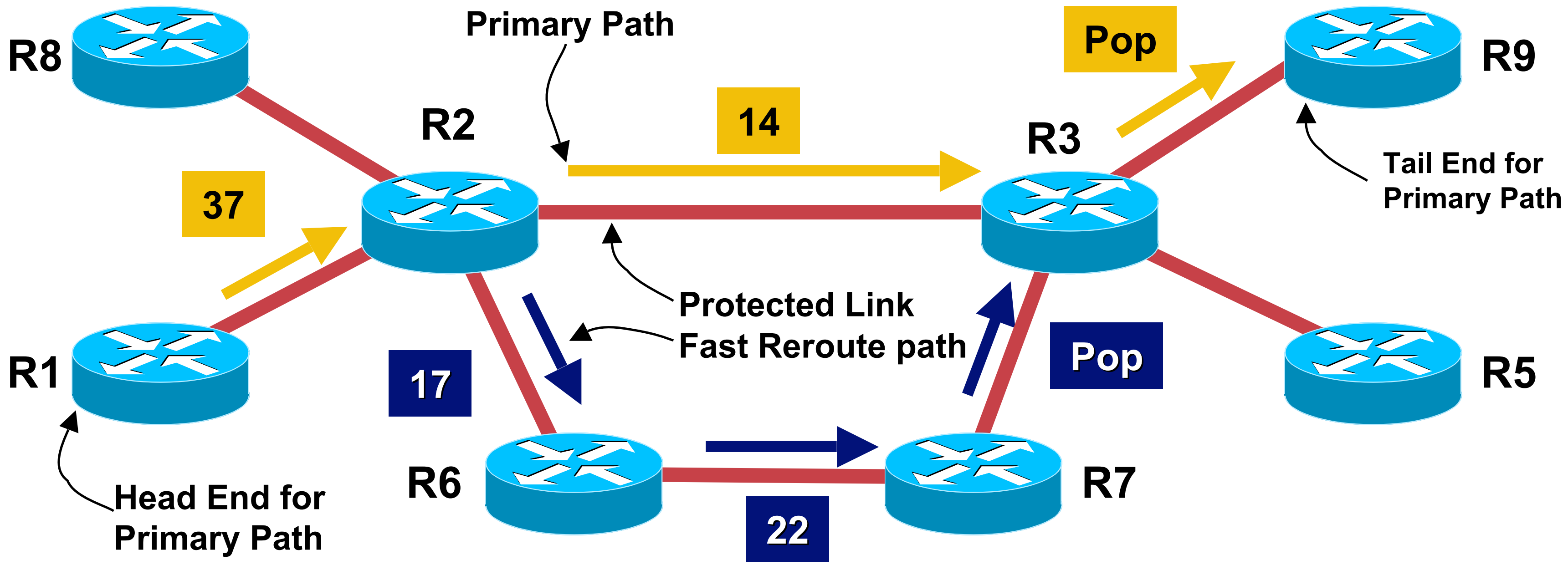
Thing	Time
Link Failure Detection	usec–msec
Failure Propagation+SPF	• hundreds of msec with aggressive tuning (400ms for 500 pfx) • sec (5-10) with defaults
Local forwarding rewrite	<100ms
TOTAL:	~500ms–10sec

FRR Failure Recovery

Since FRR is a Local Decision, No Propagation Needs to Take Place

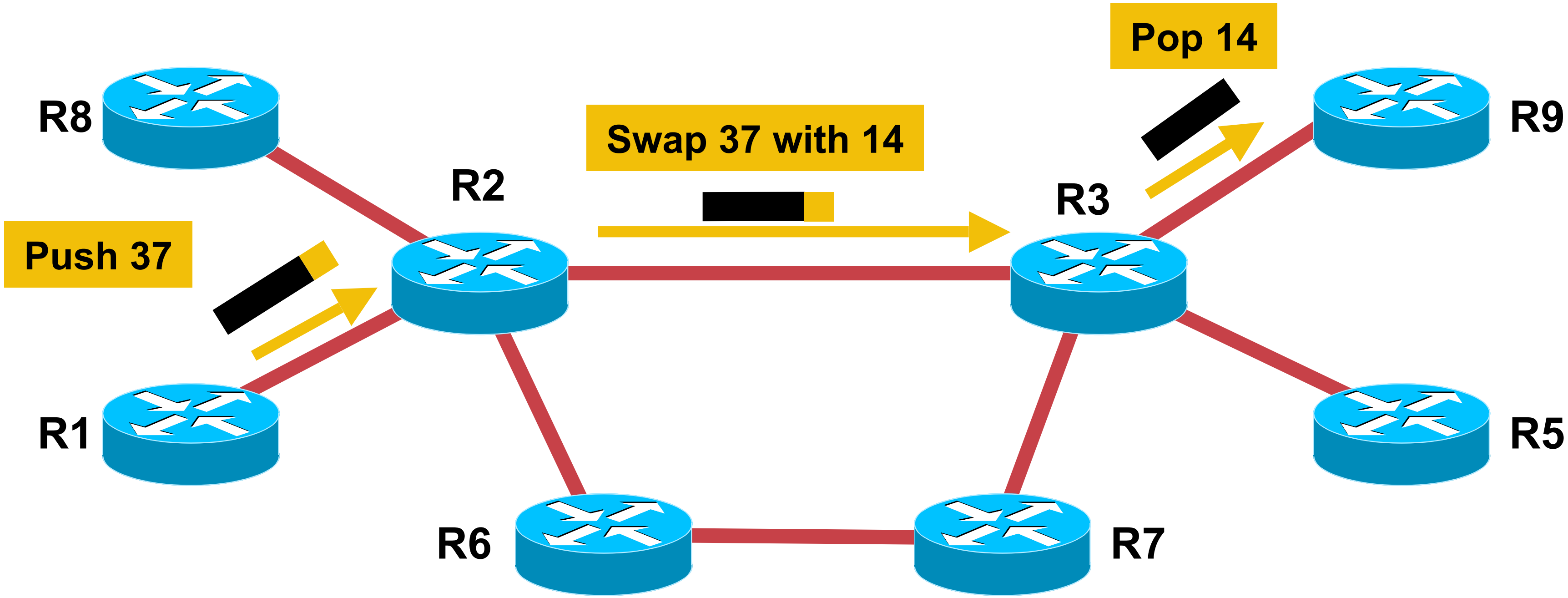
Thing	Time
Link Failure Detection	usec–msec
Failure Propagation+SPF	0
Local forwarding rewrite	<100ms
TOTAL:	<100ms (often <50ms, <10ms with properly greased skateboard)

Link Protection Example

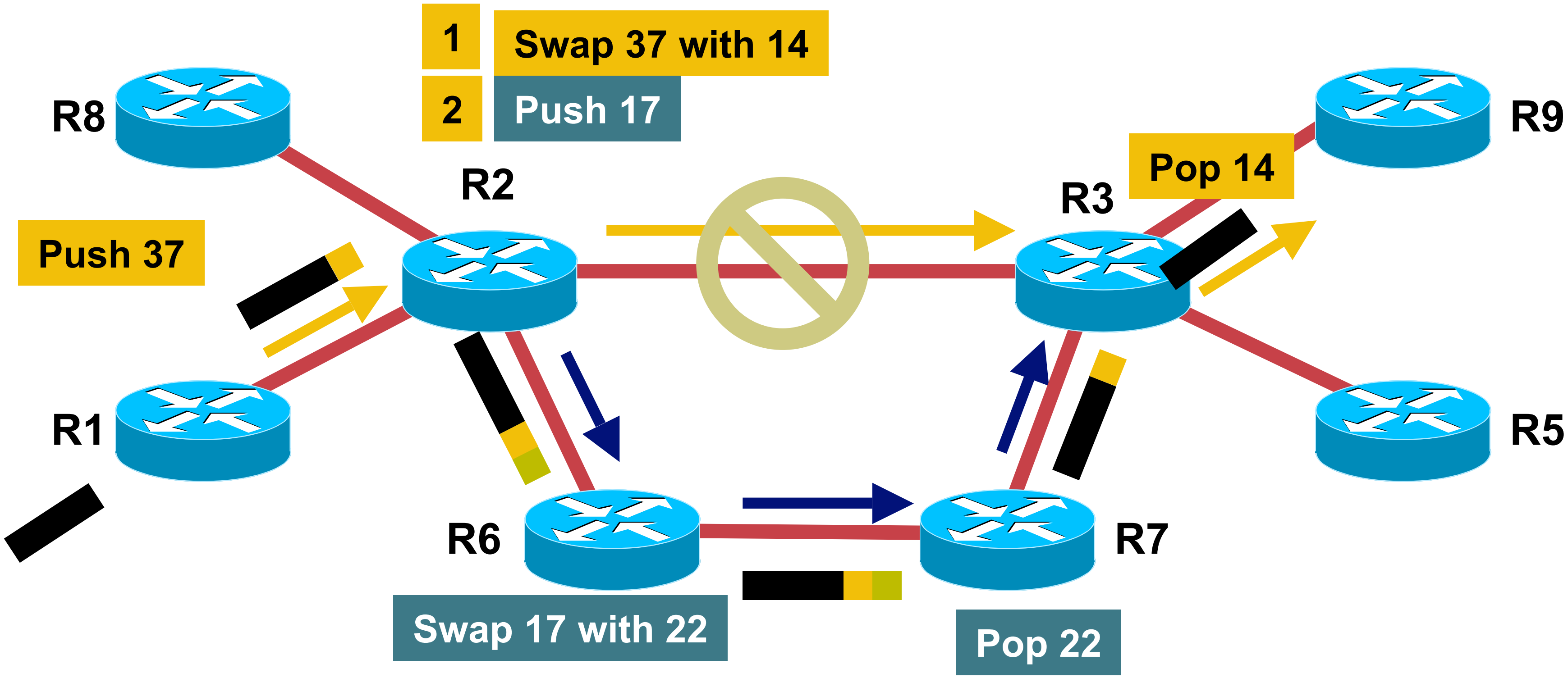


Primary path: R1 → R2 → R3 → R9
Fast Reroute path: R2 → R6 → R7 → R3

Normal TE Operation



Fast Reroute Link Failure

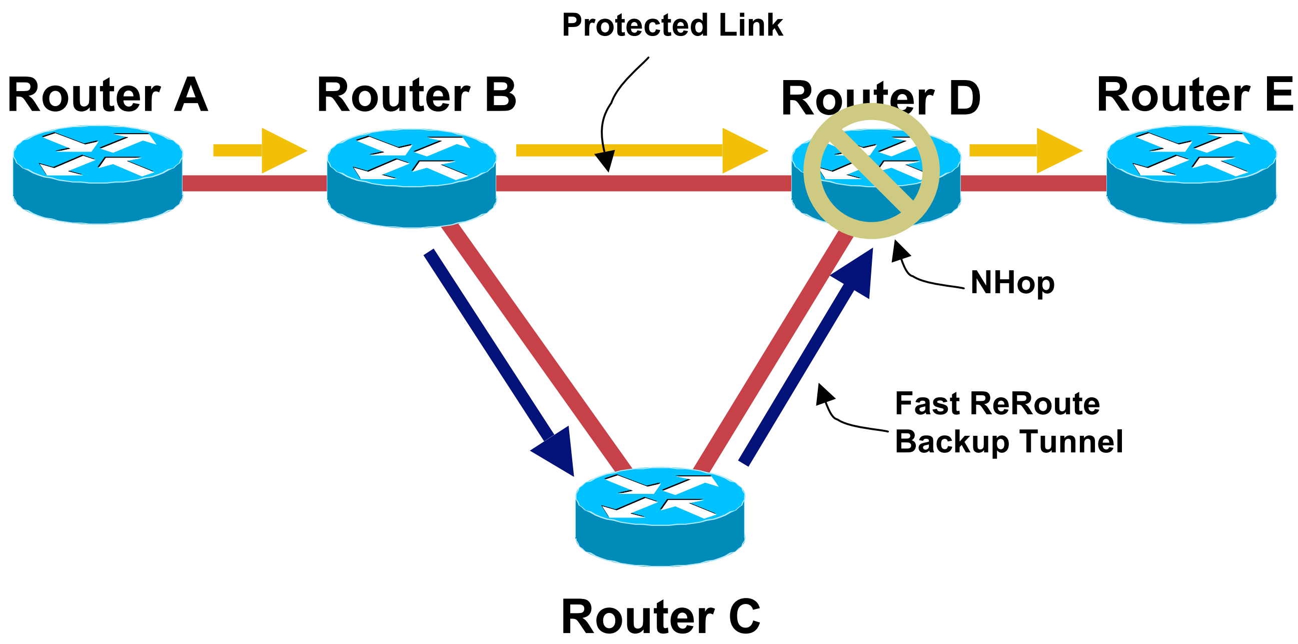


FRR Procedures

1. **Pre-establish backup paths**
2. **Failure happens, protected traffic is switched onto backup paths**
3. **After local repair, tunnel headends are signaled to recover if they want; no time pressure here, failure is being protected against**
4. **Protection is in place for hopefully ~10-30+ seconds; during that time, data gets through**

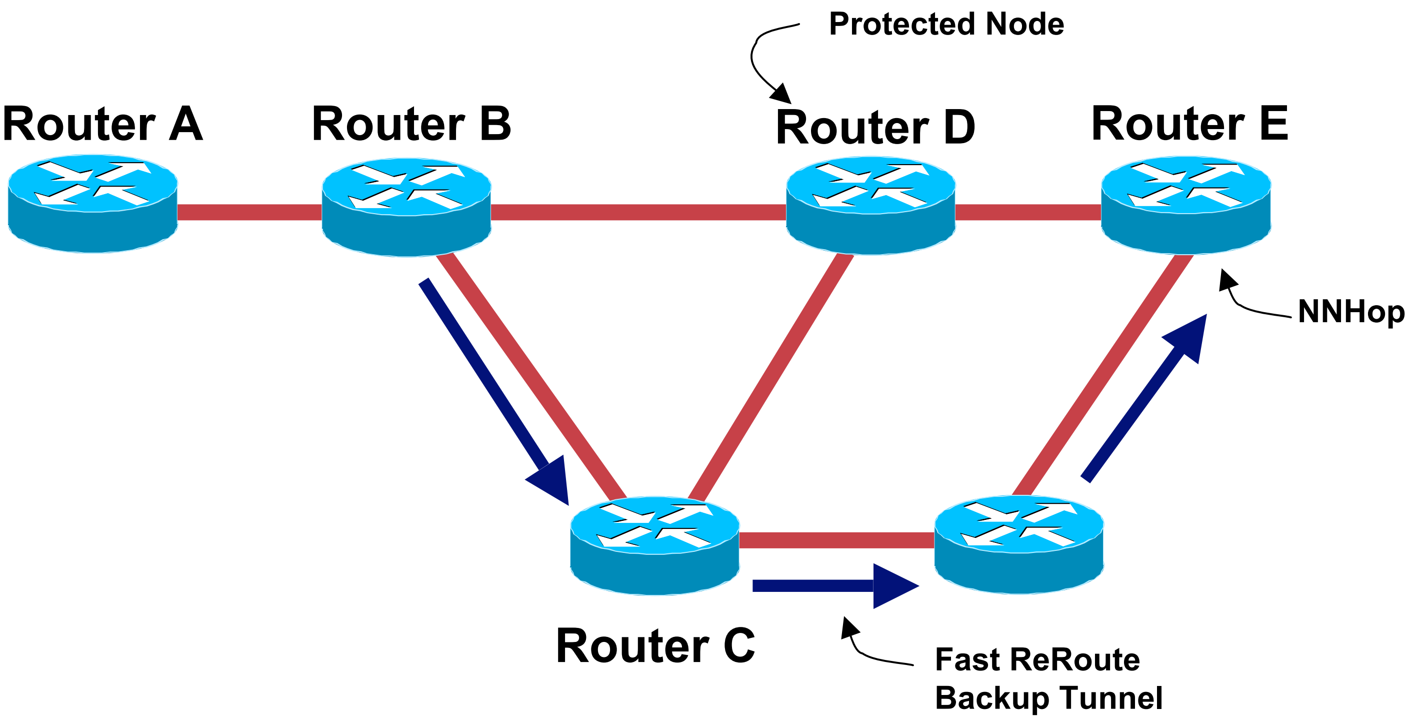
Node Protection

- **What if Router D failed?**
- **Link protection would not help as the backup tunnel terminates on Router D (which is the NHop of the protected link)**



Node Protection

- **SOLUTION: NODE PROTECTION** (If network topology allows)
- Protect tunnel to the next hop **PAST** the protected link (NNhop)



Node Protection

- Node protection still has the same convergence as link protection
- Deciding where to place your backup tunnels is a much harder problem to solve on a large-scale
- For small-scale protection, link may be better
- Configuration is identical to link protection, except where you terminate the backup tunnel (NNHop vs. NHop)

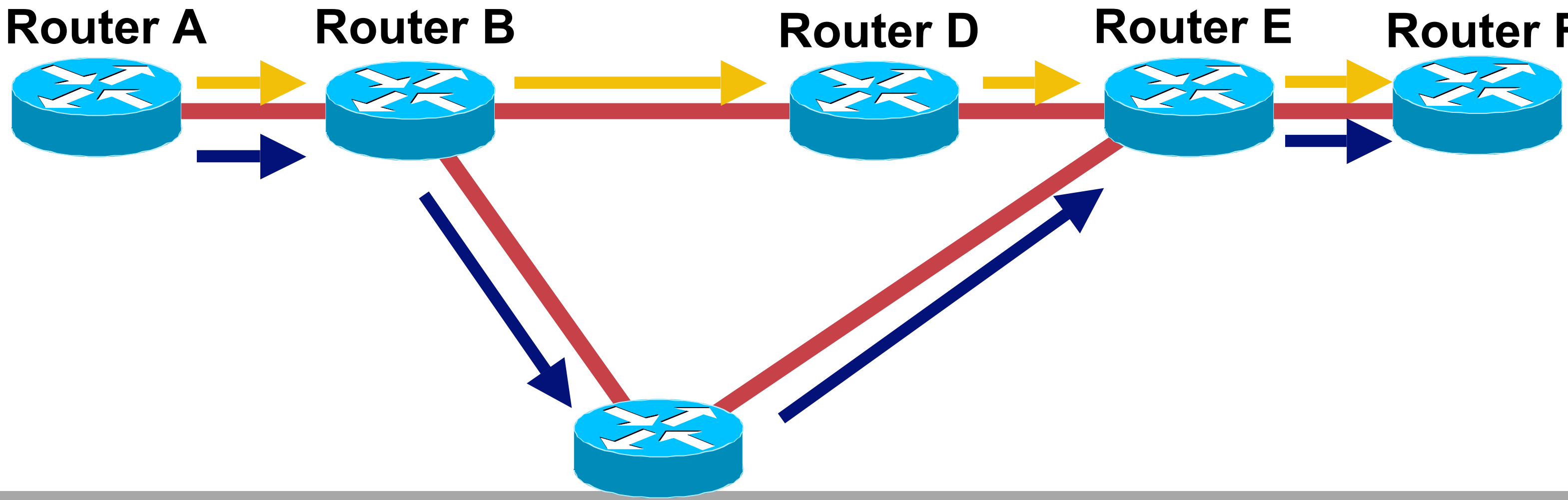
```
RouterB(config)# ip explicit-path name avoid-node
RouterB(cfg-ip-expl-path)# exclude-address <Router_D>

RouterB(config)# interface Tunnel2
RouterB(config-if)# tunnel mpls traffic-eng path-option explicit
avoid-node
```

Link and Node Protection Times

- Link and Node protection are very similar
- Protection times are commonly linear to number of protected items
- One provider gets ~35ms of loss

Path Protection

- Path protection: Multiple tunnels from TE head to tail, across diverse paths
- Backup tunnel pre-signalled. If primary tunnel goes down, tell headend to use backup



Path Protection

- **Least scalable, most resource-consuming, slowest convergence of all 3 protection schemes**
- **With no protection, worst-case packet loss is 3x path delay**
- **With path protection, worst-case packet loss is 1x path delay**
- **With link or node protection, packet loss is easily engineered to be subsecond (<100ms, <50ms, 4ms, all possible)**

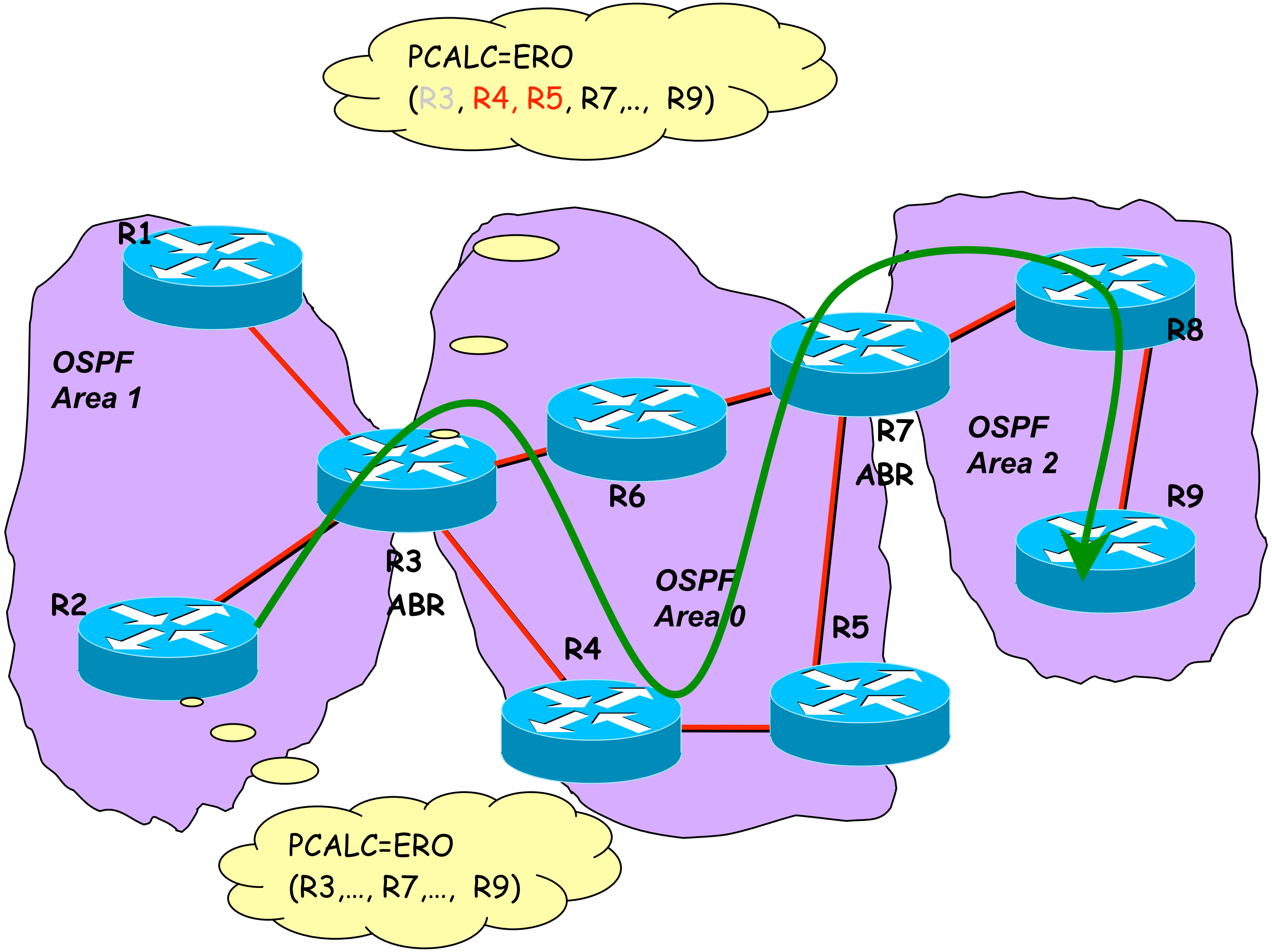
Inter-area TE

- Build LSPs across different OSPF areas
- OSPF uses Opaque LSA (type 10) within area to propagate TE information
- Use explicit path with "loose hop" option
- Each loose hop node is an ABR
- Each ABR will run CSPF to get to the next ABR in its area and inset the nodes in explicit path
- Inter-area tunnels can do reoptimization and FRR
- Autoroute is not supported for Inter-area, since you need to know the topology downstream to the tail

Enabling Inter-area TE

```
interface Tunnel1
tunnel mpls traffic-eng path-option 1 explicit name
  path-tunnel1
!
ip explicit-path name path-tunnel1
next-address loose <ABR1>
next-address loose <ABR2>
next-address loose <ABR3>
```

Inter-area TE



Configuring MPLS-TE Backup (if time ever permits)

Prerequisite Configuration (Global)

```
ip cef [distributed]

mpls traffic-eng tunnels
```

Information Distribution

- **OSPF**

```
mpls traffic-eng tunnels
mpls traffic-eng router-id loopback0
mpls traffic-eng area ospf-area
```

- **ISIS**

```
mpls traffic-eng tunnels
mpls traffic-eng router-id loopback0
mpls traffic-eng level-x
metric-style wide
```

Information Distribution

- **On each physical interface**

```
interface pos0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth Kbps (Optional)
  mpls traffic-eng attribute-flags attributes (Opt)
```

Build a Tunnel Interface (Headend)

```
interface Tunnel0
  ip unnumbered loopback0
  tunnel destination RID-of-tail
  tunnel mode mpls traffic-eng
  tunnel mpls traffic-eng bandwidth  10
```

Tunnel Attributes

```
interface Tunnel0
  tunnel mpls traffic-eng bandwidth Kbps
  tunnel mpls traffic-eng priority pri [hold-pri]
  tunnel mpls traffic-eng affinity properties [mask]
  tunnel mpls traffic-eng autoroute announce
```

Path Calculation

- **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]
```

Multiple Path Calculations

- **A tunnel interface can have several path options, to be tried successively**

```
Interface Tunnel 1
…….
tunnel mpls traffic-eng path-option 10 explicit name foo
tunnel mpls traffic-eng path-option 20 explicit name bar
tunnel mpls traffic-eng path-option 30 dynamic
```

Static and Policy Routing Down a Tunnel

- **Static routing**

```
ip route prefix mask Tunnel0
```

- **Policy routing (Global Table)**

```
access-list 101 permit tcp any any eq www

interface Serial0
  ip policy route-map foo

route-map foo
  match ip address 101
  set interface Tunnel0
```

Autoroute and Forwarding Adjacency

```
interface Tunnel0
  tunnel mpls traffic-eng autoroute announce
```

OR

```
  tunnel mpls traffic-eng forwarding-adjacency
  isis metric x level-y (ISIS)
  ip ospf cost ospf-cost (OSPF)
```

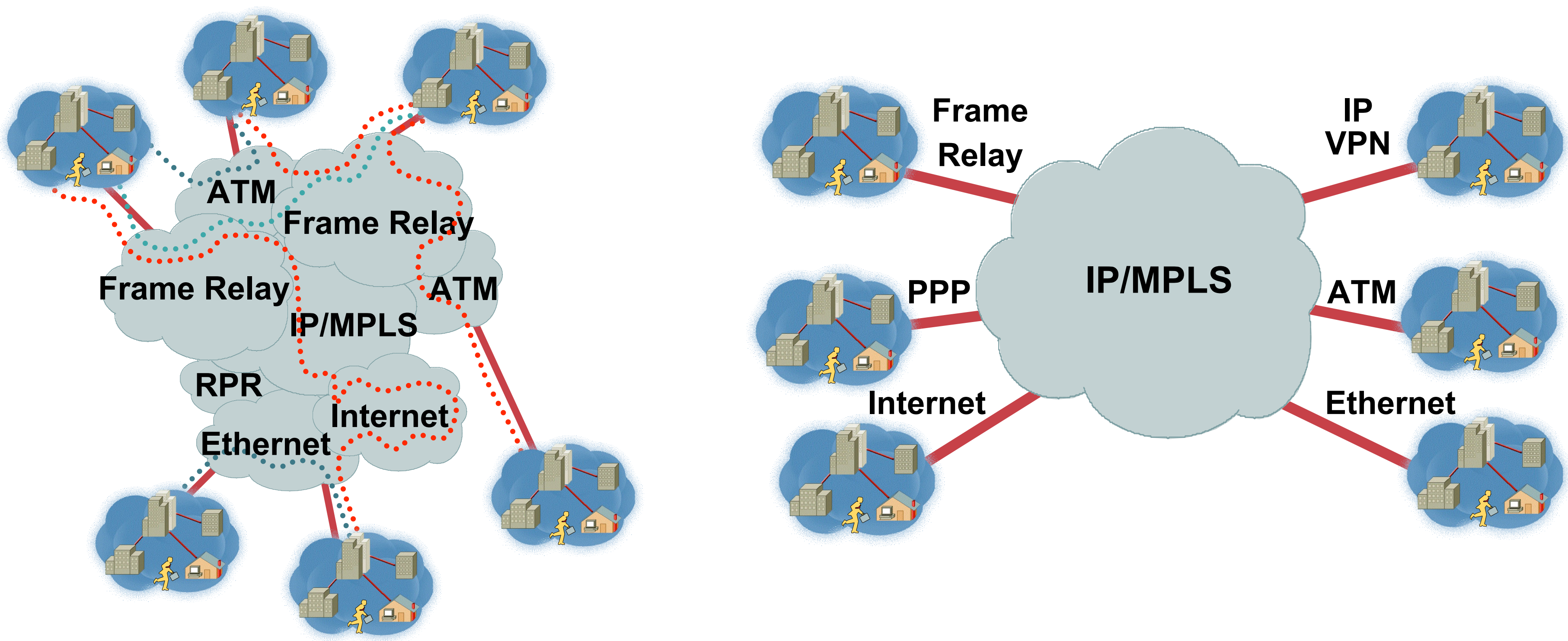
L2VPN Concepts

Agenda

- Introduction to L2VPN
- PWE3 Signaling Concepts
- Virtual Private Wire Service (VPWS) Transports
- VPWS Service Interworking
- Virtual Private LAN Service (VPLS)

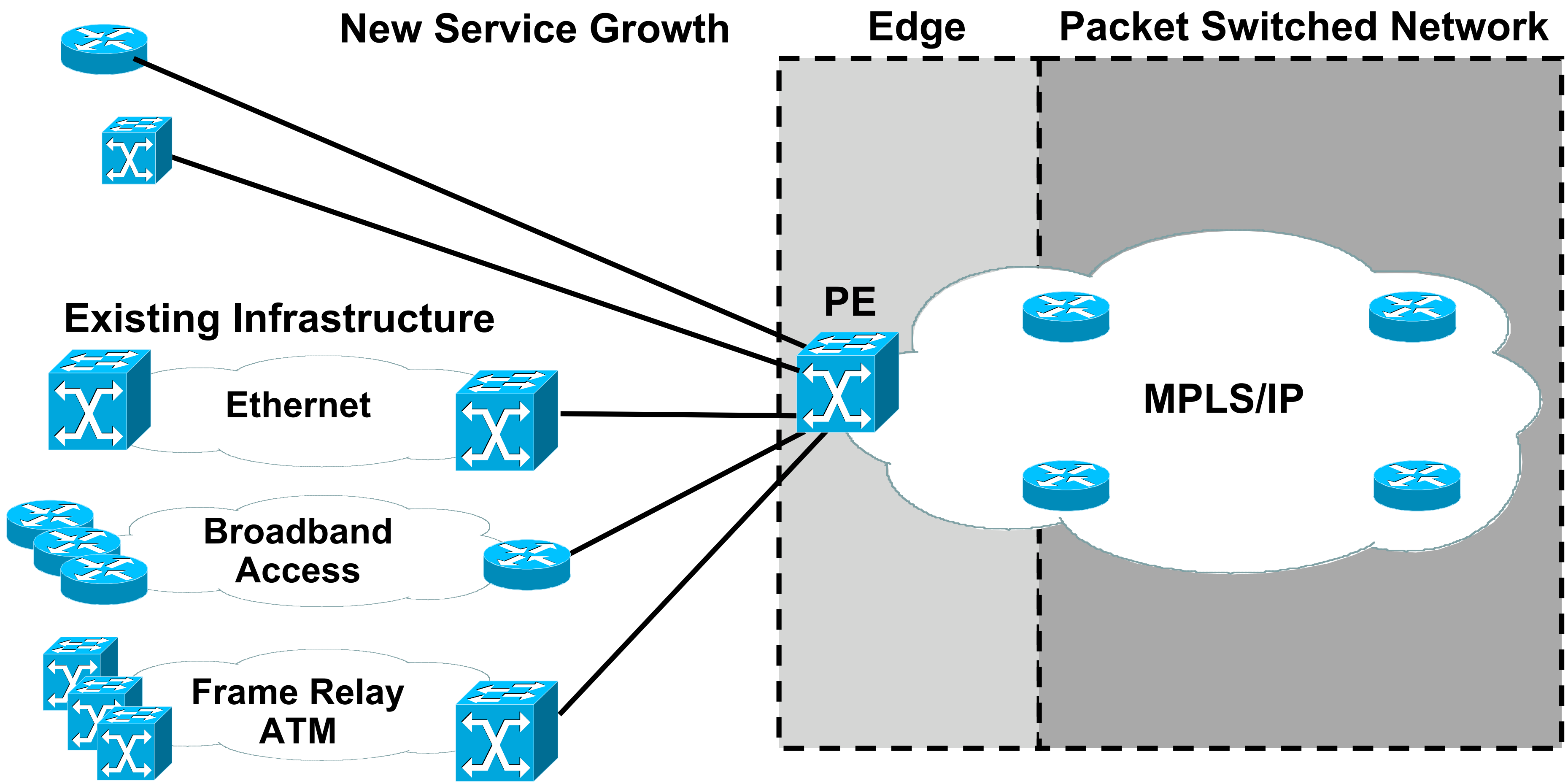
Introduction to L2VPN

Multiple Services over a Converged Infrastructure



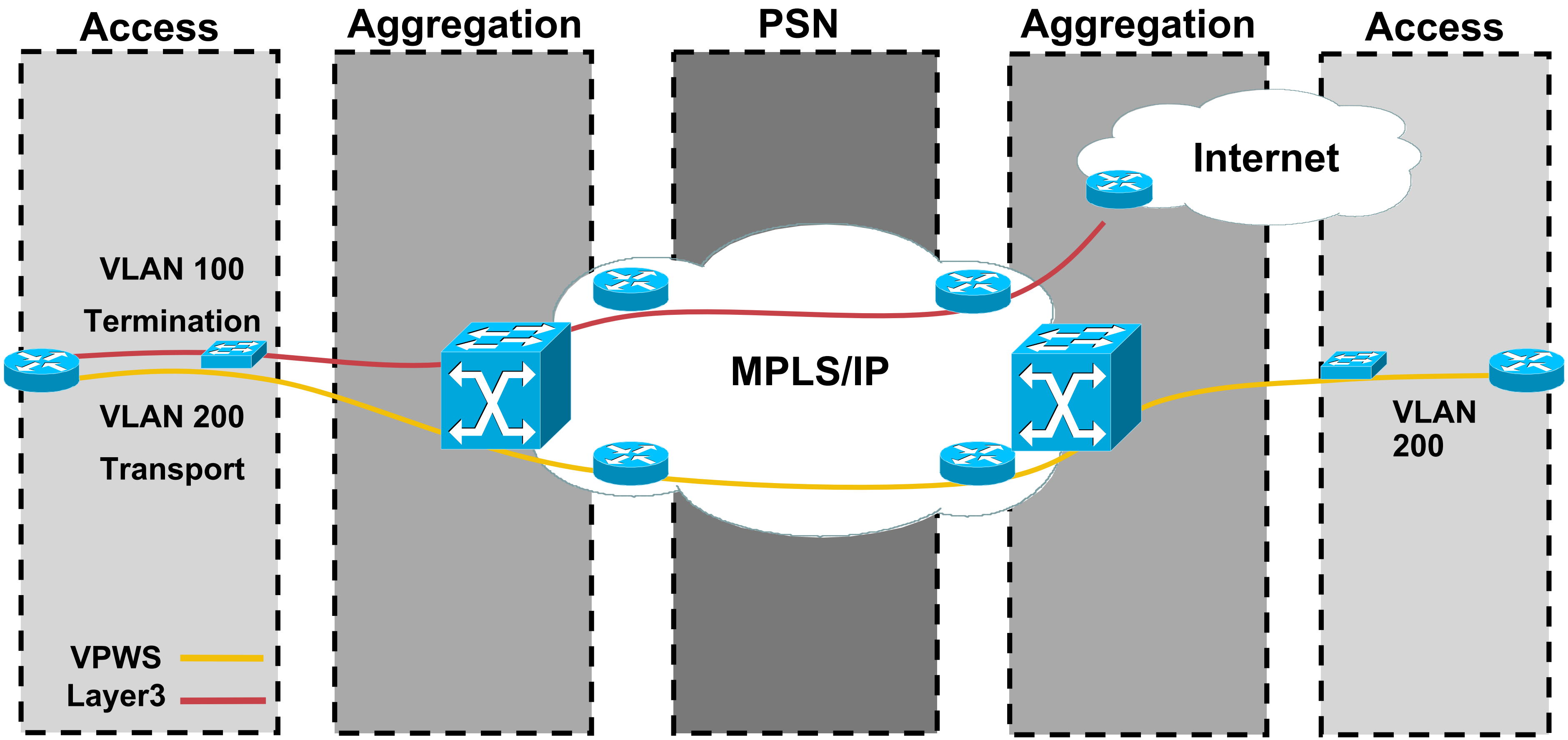
Many Services, Many Networks → **Many Services, One Network**

Motivation for L2VPNs:
Converged Networks Support



- Reduce overlapping core expense; consolidate trunk lines
- Offer multiservice/common interface (i.e. Ethernet MUX = L2, L3 and Internet)
- Maintain existing revenues from legacy services

Motivation for L2VPNs:
The Ever Expanding Applications of Ethernet

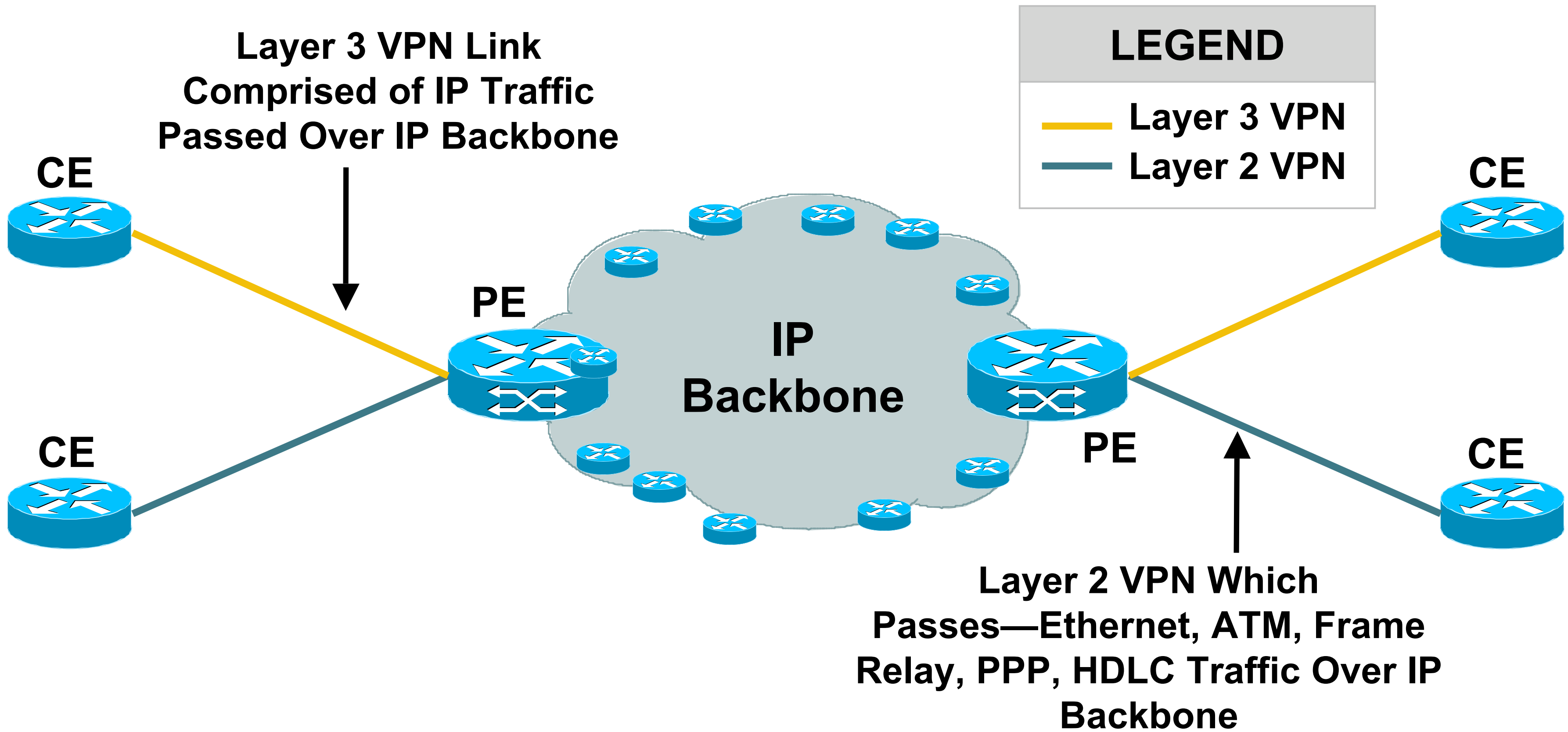


- Fast becoming the access technology of choice
- Layer 2, Layer 3 and Internet Services on a common port
- Extends the reach of Metro Area Ethernet Networks

Why is L2VPN needed?

- Allows SP to have a single infrastructure for both IP and legacy services
  - Migration
  - Provisioning is incremental
  - Network Consolidation
  - Capital and Operational savings
- Customer can have their own routing, qos policies, security mechanisms, etc
  - Layer 3 (IPv4, IPX, OSPF, BGP, etc …) on CE routers is transparent to MPLS core
  - CE1 router sees CE2 router as next-hop
  - No routing involved with MPLS core
- open architecture and vendor interoperability

Introduction to Layer 2 and Layer 3 VPN Services



- **Layer 2 and Layer 3 VPN Services are offered from the edge of a network**

Layer 3 and Layer 2 VPN Characteristics

Layer 3 VPNs

1. Packet-based forwarding e.g. IP
2. SP is involved
3. IP specific
4. Example: RFC 2547bis VPNs (L3 MPLS-VPN)

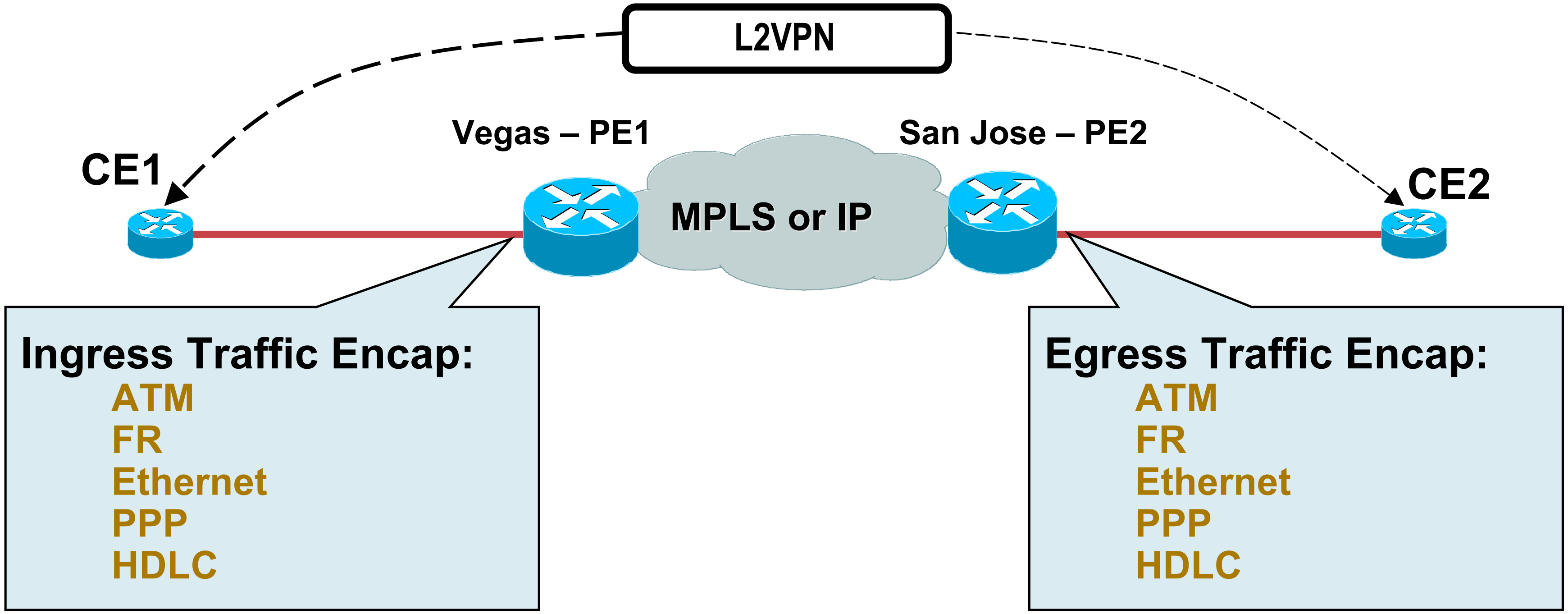
Layer 2 VPNs

1. Frame-based forwarding e.g. DLCI, VLAN, VPI/VCI
2. No SP involvement
3. Multiprotocol support
4. Example: FR—ATM—Ethernet

The Choice of L2VPN over L3VPN Will Depend on How Much Control the Enterprise Wants to Retain

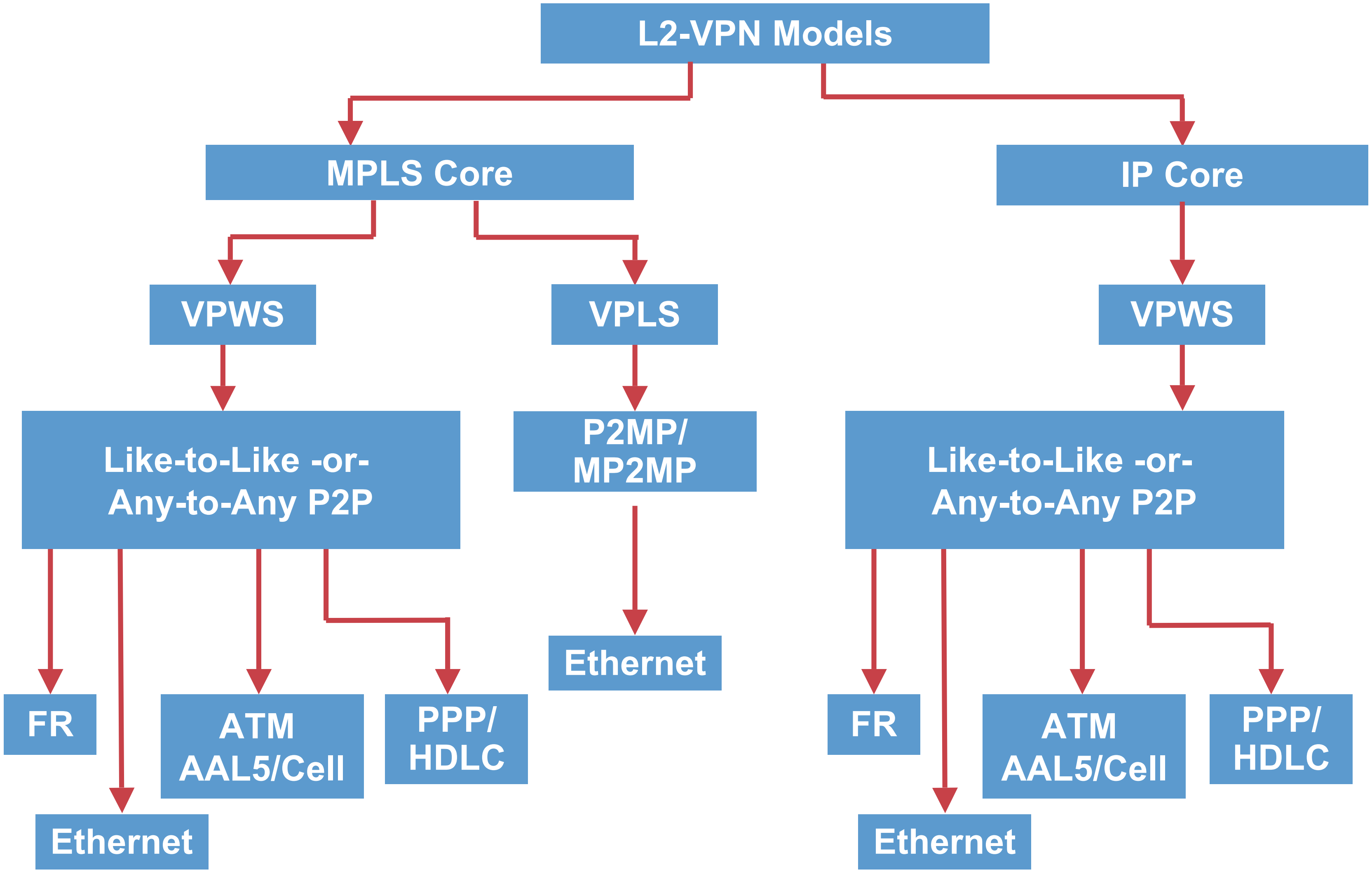
L2 VPN Services Are Complementary to L3 VPN Services

L2VPN - Simple definition



L2VPN provides an end-to-end layer 2 connection to an enterprise office in Vegas and San Jose over a SP's MPLS or IP core

L2VPN Models



Pseudowire—
IETF Technology Adoption

- Virtual private wire service (VPWS) P2P

    RFC3916 Pseudo Wire Emulation Edge-to-Edge (PWE3) Requirements

    RFC3985 Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture

    RFC 4447 Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)

    RFC4385 Pseudo wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN

    RFC 4448 Encapsulation Methods for Transport of Ethernet over MPLS Networks

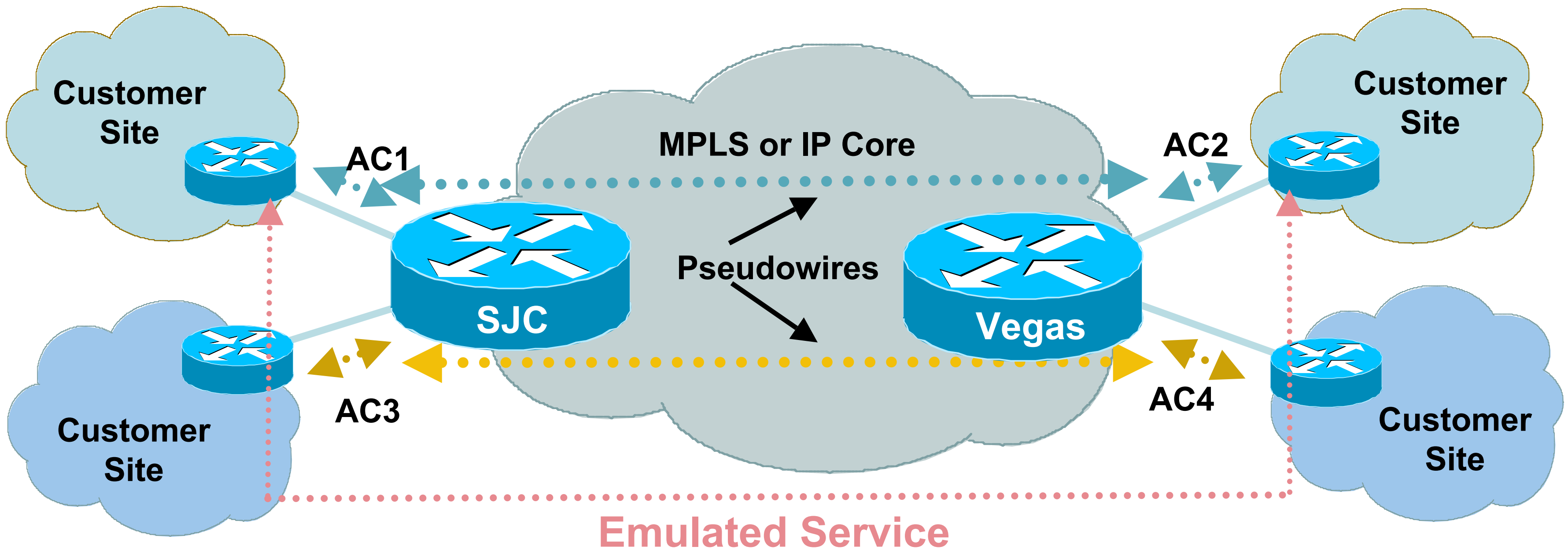
    draft-ietf-pwe3-[atm, frame-relay etc.]

- Virtual private LAN services (VPLS) P2M

    draft-ietf-l2vpn-vpls-ldp-xx
    draft-ietf-l2vpn-vpls-bgp-xx

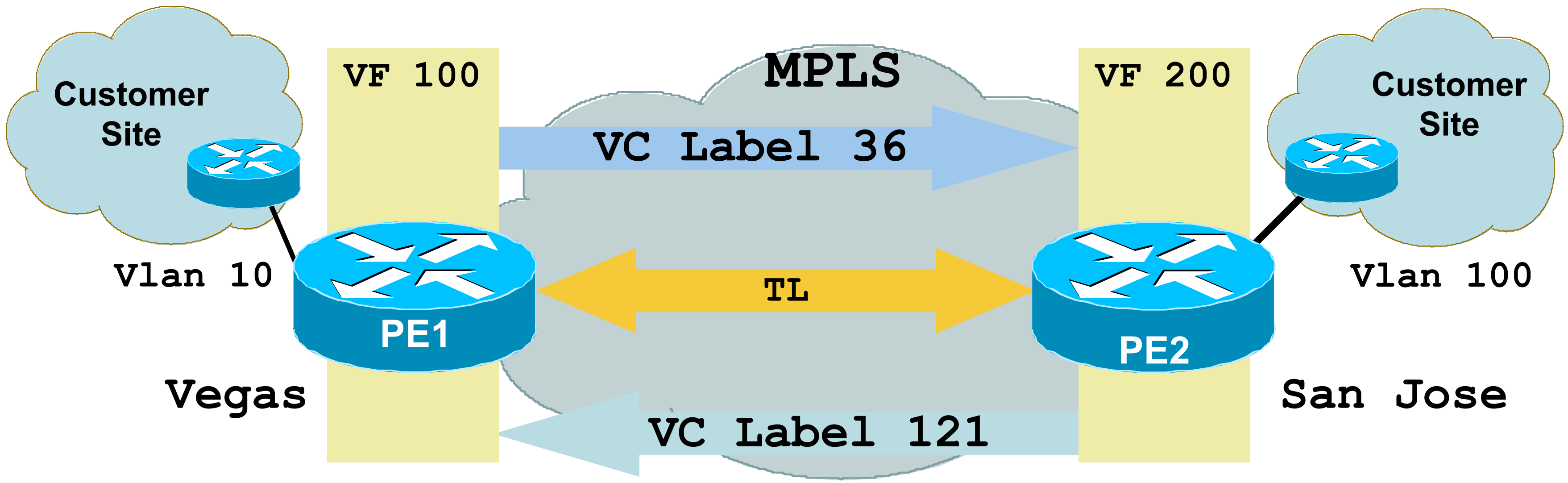
- **Layer 2 Transport (VPWS)**
  - **L2TPv3**
    - **draft-ietf-l2tpext-l2tp-base-xx**
    - **draft-ietf-l2tpext-l2tpmib-base-xx**

VPWS—Pseudowire Reference Model

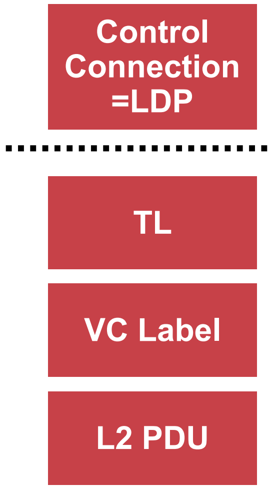


A Pseudowire (PW) Is a Connection Between Two Provider Edge (PE) Devices Which Connects Two Attachment Circuits (ACs)

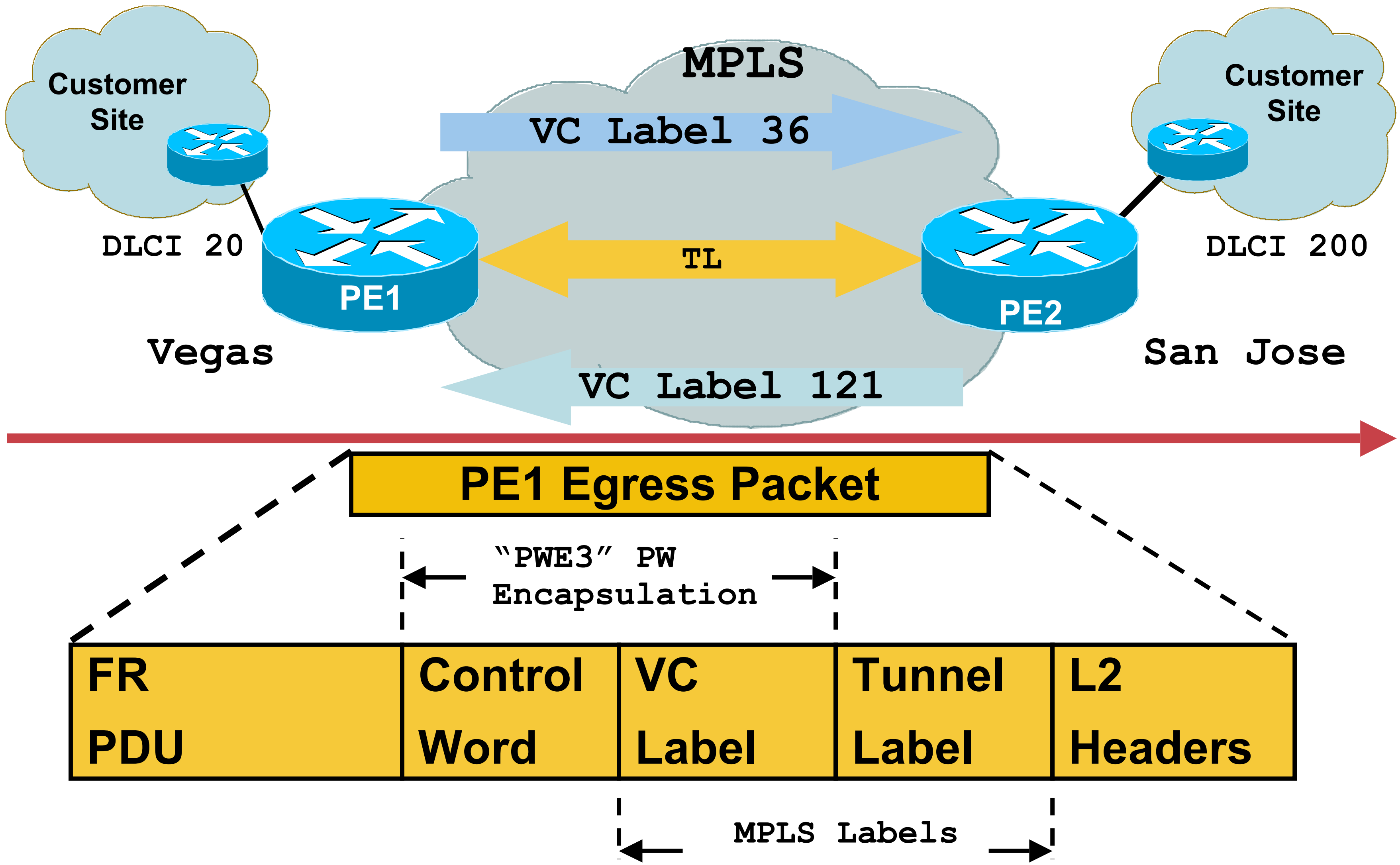
Building Blocks for L2VPNs— Data Plan Components—MPLS Core



- Virtual Forwarders (VF)—Subsystem that associates AC to PW
- Tunnel Label (TL)—Path between PE1 and PE2
- Pseudowire (PW)—Paths between VFs, a pair of unidirectional LSPs—VC label
- Attachment Circuits (AC)—L2 connection between CE and PE, i.e. VLAN, DLCI, ATM, etc.

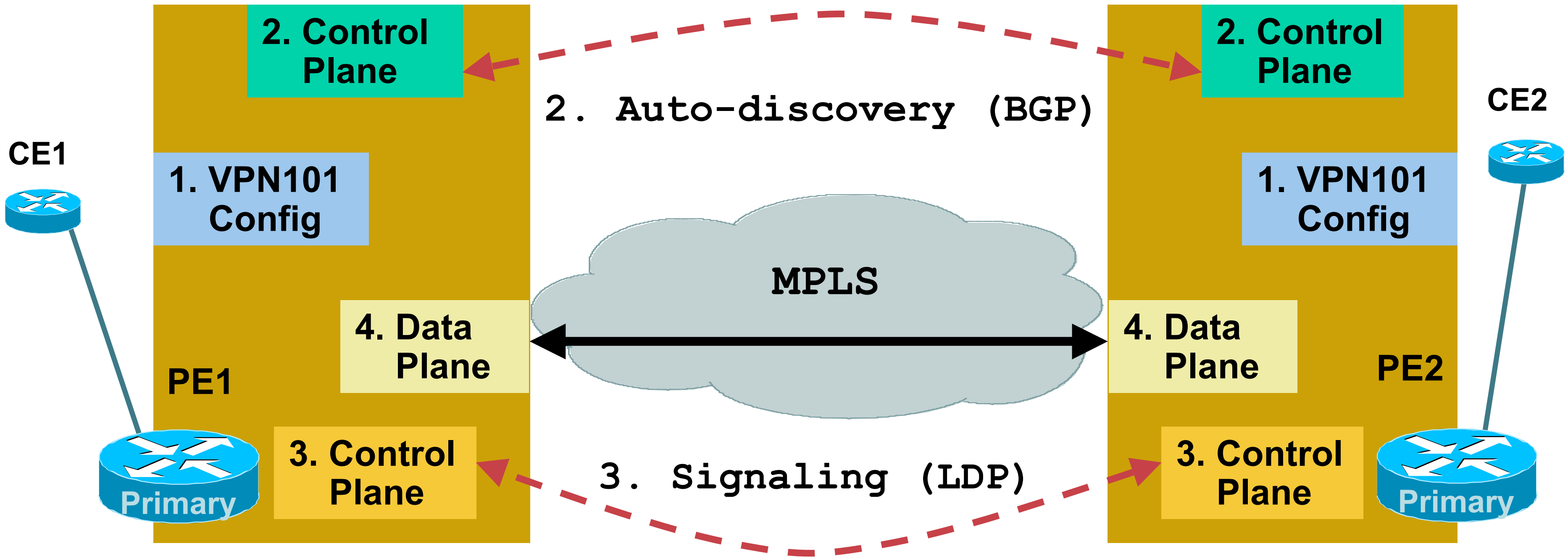


Building Blocks for L2VPNs—Data Plan Components—FR Example



PWE3 Signaling Concepts

Building Blocks for L2VPNs—Control Plane



1.	Provision	Config VPN
2.	Auto-discovery	Advertise loopback and VPN members
3.	Signaling	Setup pseudowire
4.	Data Plane	Packet forwarding

LDP Signaling Overview

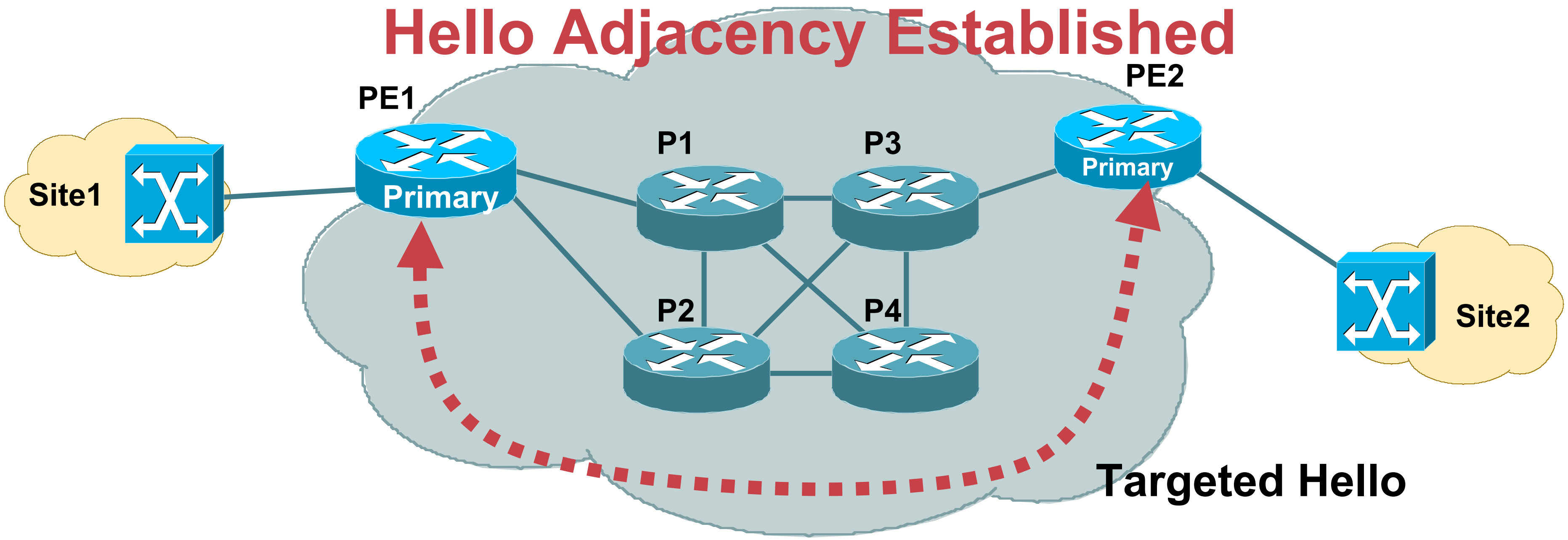
Four Classes of LDP Messages:

1. **Peer discovery**
   - LDP link hello message
   - Targeted hello message
2. **LDP session**
   - LDP initialization and keepalive
   - Setup, maintain and disconnect LDP session
3. **Label advertisement**
   - Create, update and delete label mappings
4. **LDP notification**
   - Signal error or status info

Peer discovery: **UDP**

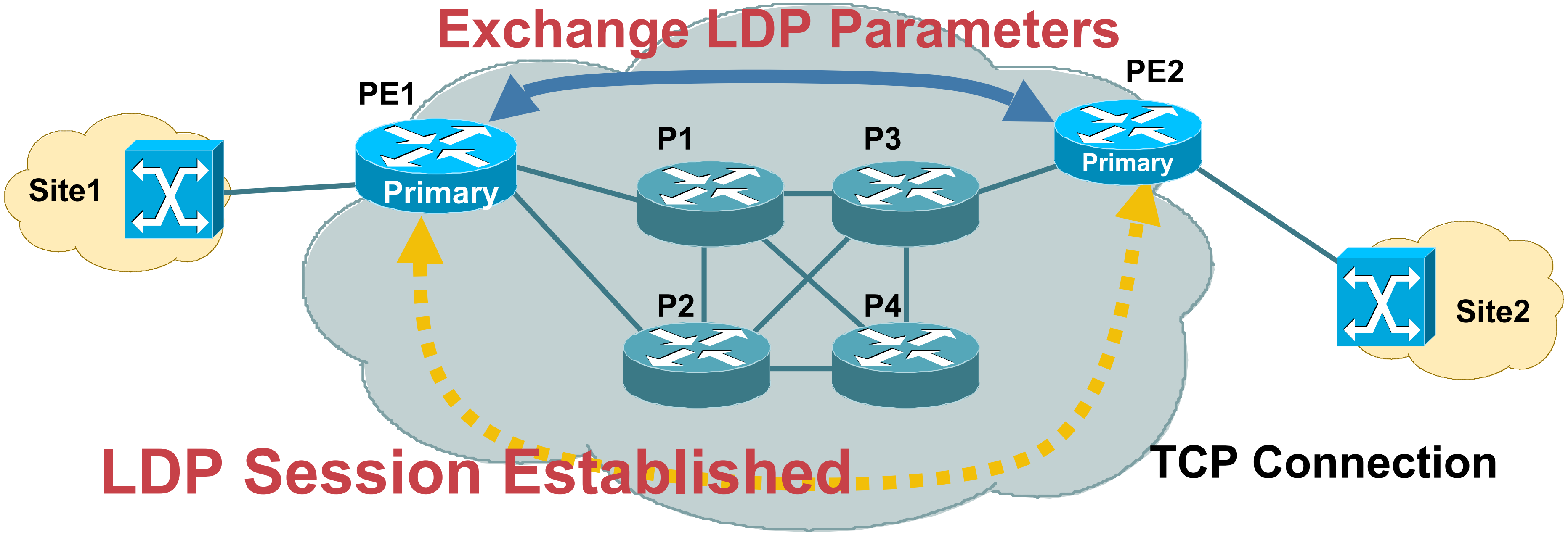
LDP session, Label advertisement, LDP notification: **TCP**

L2VPN LDP Extended Discovery



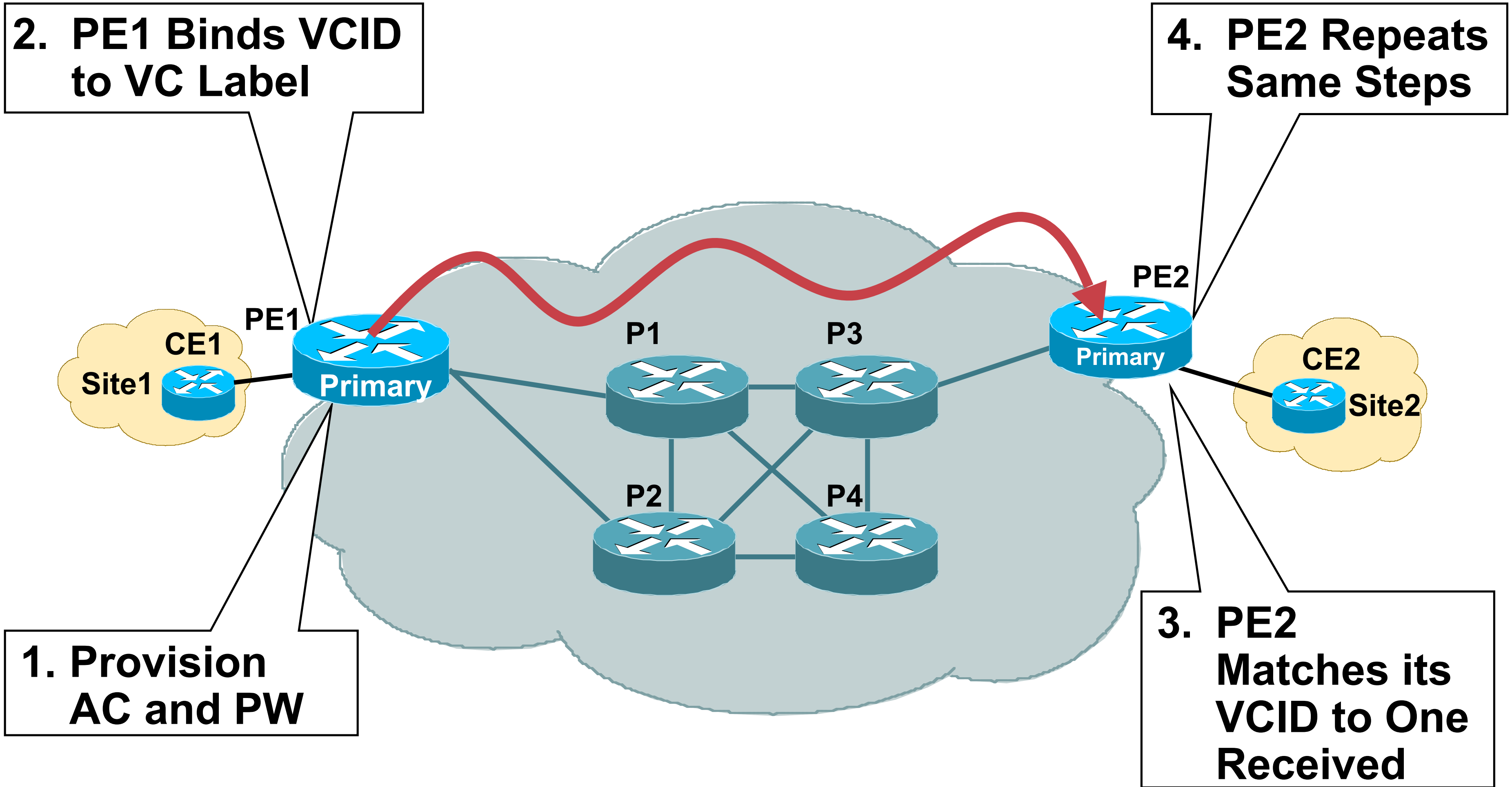
- **Targeted Hello Messages Are Exchanged as UDP Packets on Port 646 Consisting of router-id and label space**

L2VPN LDP Session Establishment



- Active role PE—establishes TCP connection using port 646
- LDP peers exchange and negotiate session parameters such as the protocol version, label distribution methods, timer values, label ranges, and so on
- LDP session is operational

L2VPN—Pseudowire Label Binding



Uni-Directional PW LSP Established

New VC FEC Element

VC TLV	C	VC Type	VC Info Length
Group ID			
VC ID			
Interface Parameters			

Virtual Circuit FEC Element

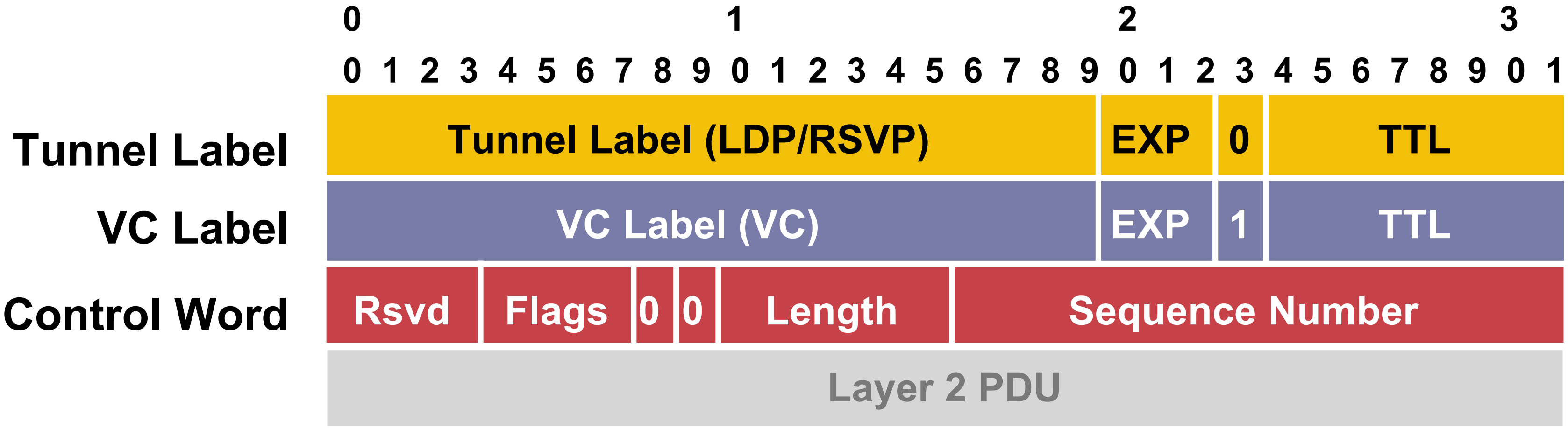
- C—control word present
- VC Type—ATM, FR, Ethernet, HDLC, PPP, etc.
- VC Info Length—length of VCID
- Group ID—group of VCs referenced by index (user configured)
- VC ID—used to identify Virtual Circuit
- Interface Parameters—MTU, etc.

Pseudowire VC Type

Some Widely Deployed VC Types

PW Type	Description
0x0001	Frame Relay DLCI
0x0002	ATM AAL5 SDU VCC transport
0x0003	ATM transparent cell transport
0x0004	Ethernet Tagged Mode (VLAN)
0x0005	Ethernet
0x0006	HDLC
0x0007	PPP

L2VPNs—Label Stacking



Three Layers of Encapsulation

- Tunnel label—determines path through network
- VC label—identifies VC at endpoint
- Control word—contains attributes of L2 payload (optional)

Generic Control Word—VC Information Fields

Control Word

bits	4	4	8	16
	Rsvd	Flags	Length	Sequence Number

- Use of control word is optional
- Flags—carries “flag” bits depending on encapsulation

  (FR;FECN, BECN, C/R, DE, ATM;CLP, EFCI, C/R, etc)
- Length—required for padding small frames when < interface MTU
- Sequence number—used to detect out of order delivery of frames

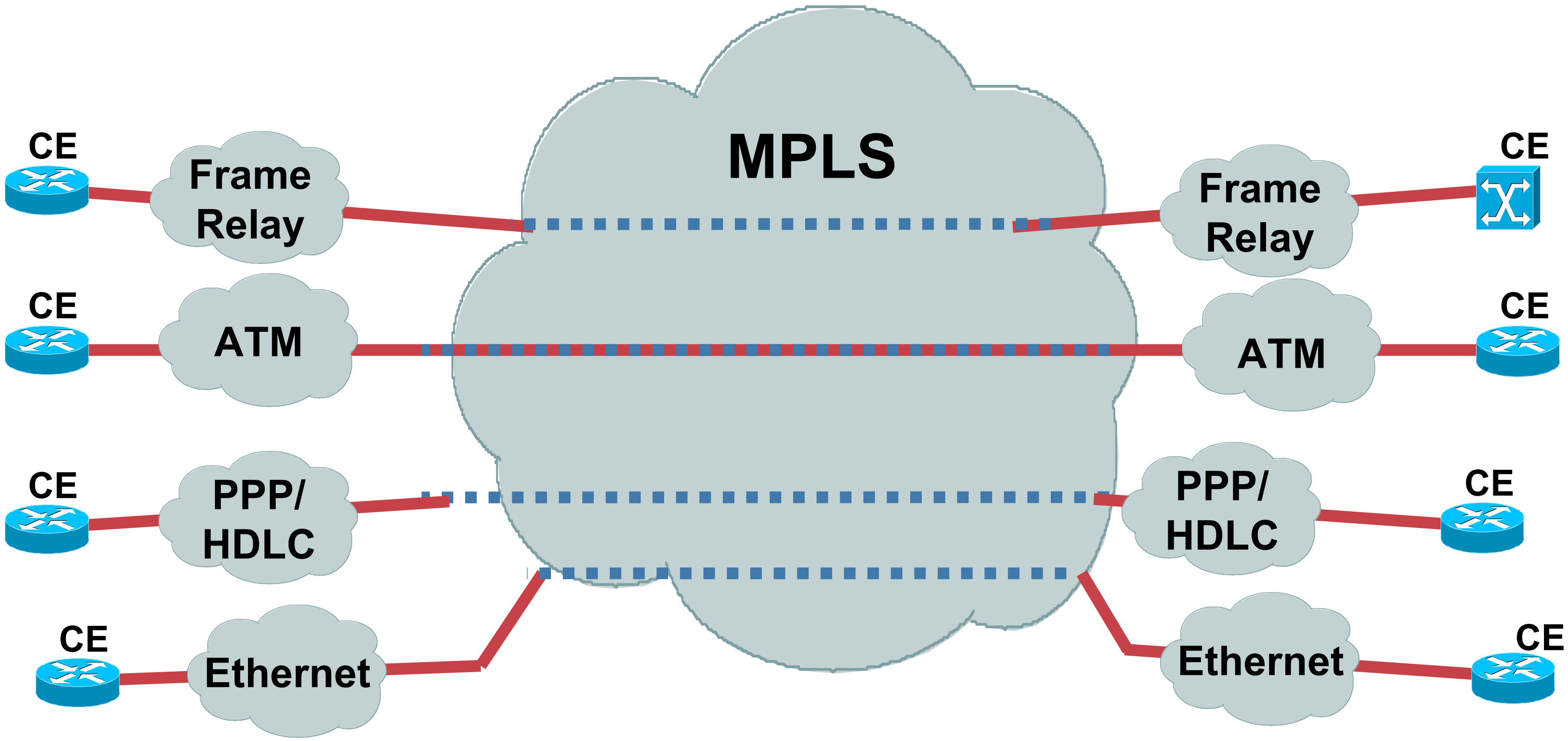
Control Word	
Encap.	**Required**
CR	No
AAL5	Yes
Eth	No
FR	Yes
HDLC	No
PPP	No

VPWS Transport

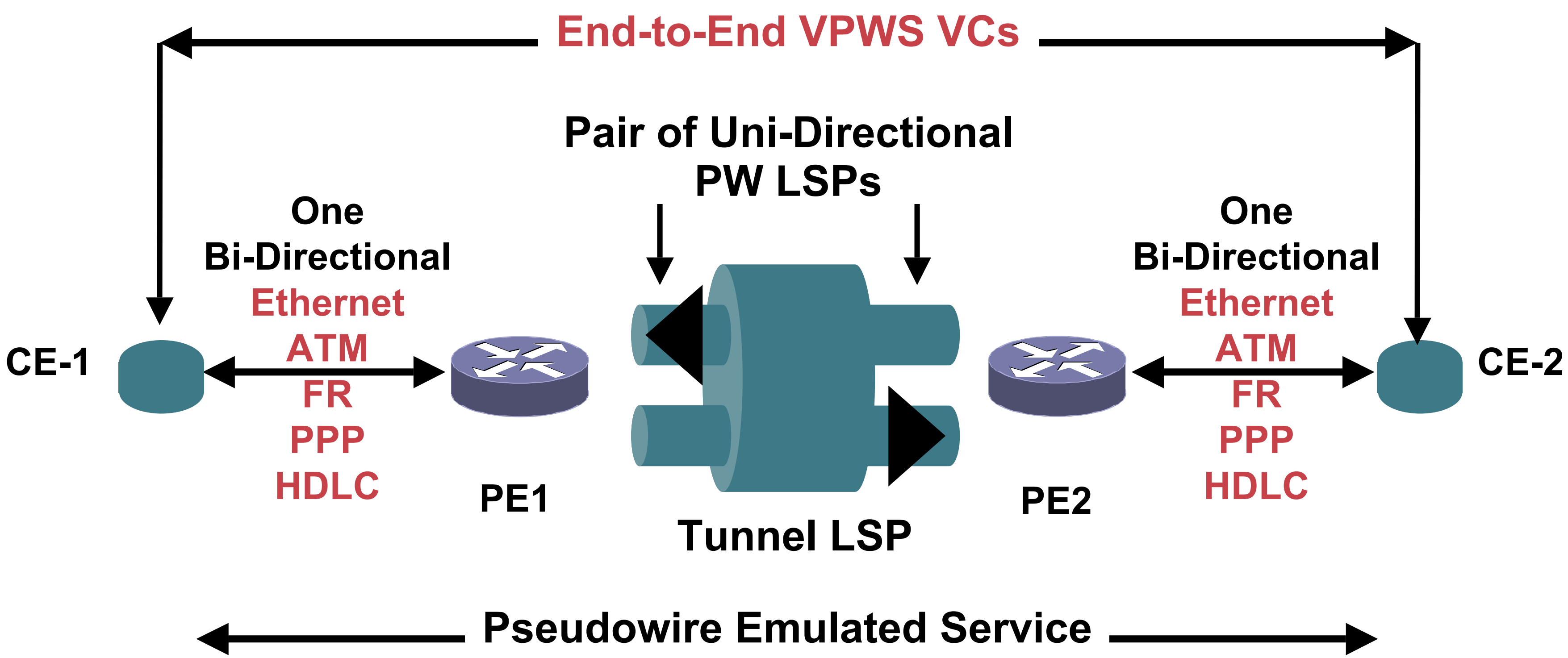
VPWS Transports—Encapsulations

- **Ethernet/802.1Q VLAN (EoMPLS)**
  - RFC 4448 Encapsulation Methods for Transport of Ethernet over MPLS Networks
- **Frame Relay (FRoMPLS)**
  - draft-ietf-pwe3-frame-relay-encap-xx.txt
- **ATM AAL5 and ATM Cell (ATMoMPLS)**
  - draft-ietf-pwe3-atm-encap-xx.txt
- **PPP/HDLC (PPPoMPLS/HDLCoMPLS)**
  - draft-ietf-pwe3-hdlc-ppp-encap-mpls-xx.txt

VPWS Transports

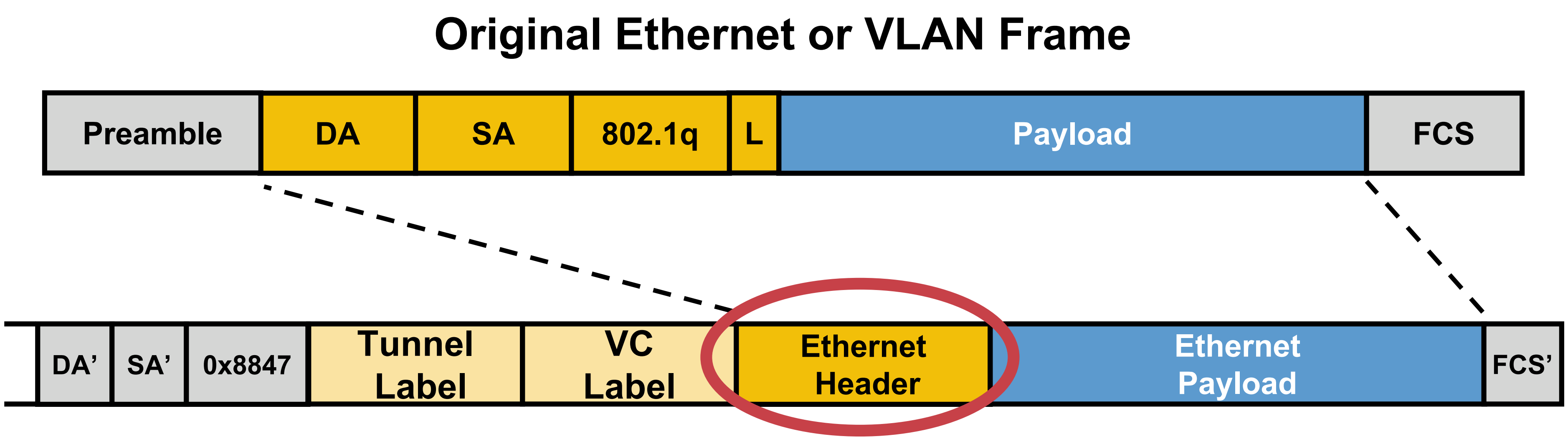


VPWS Transports Service—Reference Model



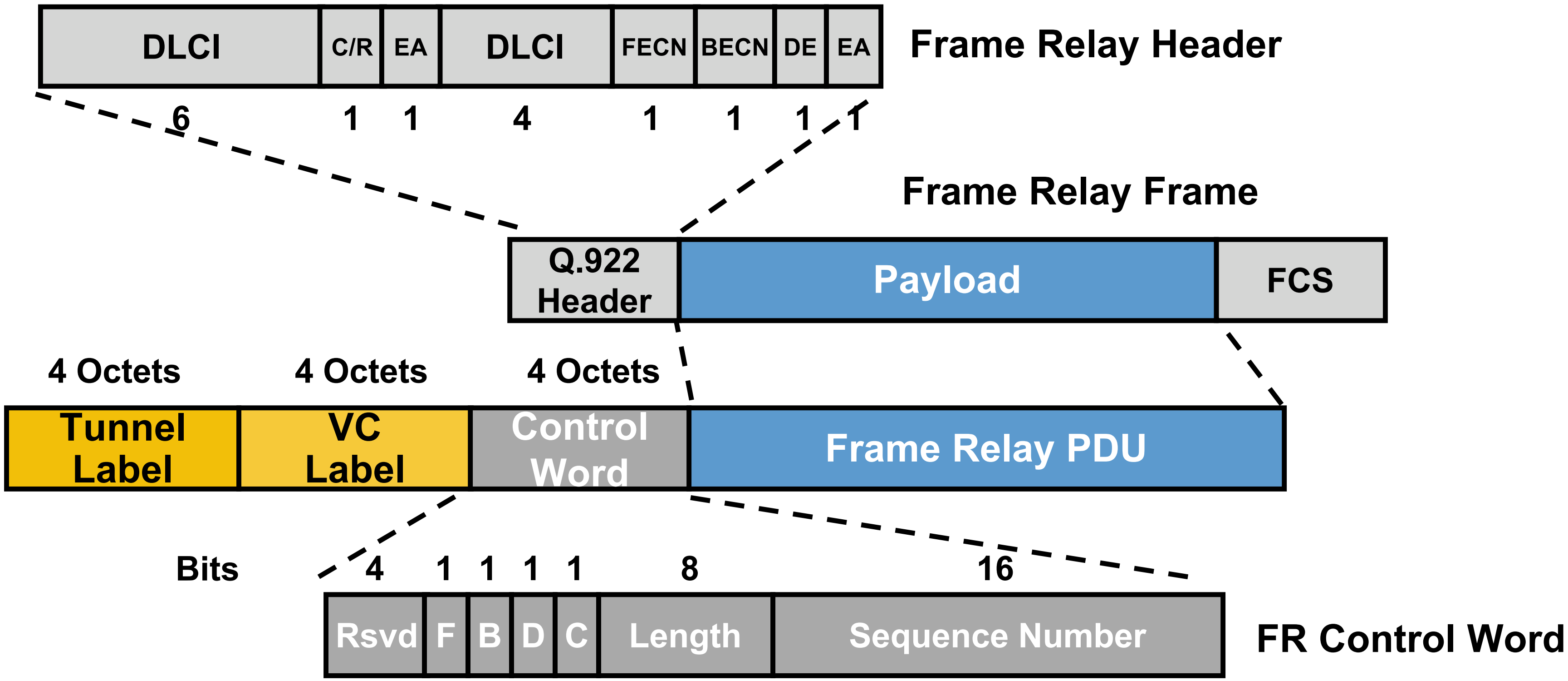
- **Pseudowire transport (across PEs) applications**
- **Local switching (within a PE) applications**

VPWS EoMPLS—
RFC 4448



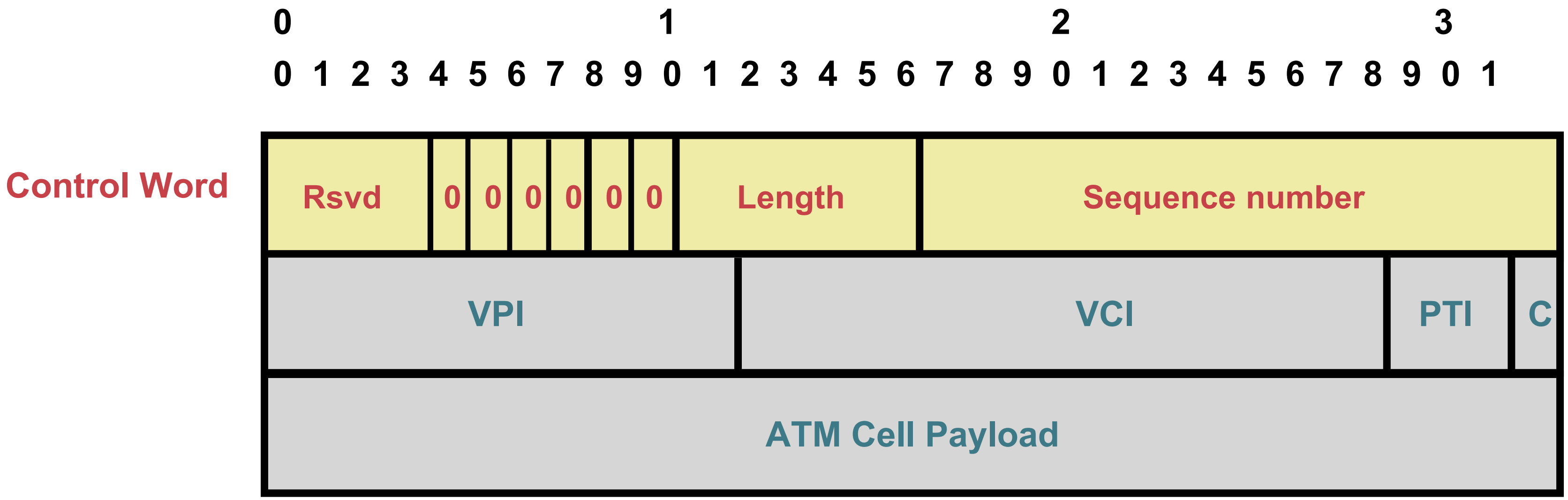
- VC type-0x0004 is used for VLAN over MPLS application
- VC type-0x0005 is used for Ethernet port tunneling application (port transparency)

VPWS FRoMPLS—draft-ietf-pwe3-frame-relay-encap-xx.txt



- F = FECN (Forward Explicit Congestion Notification)
- B = BECN (Backward Explicit Congestion Notification)
- D = DE (Discard Eligibility Indicator)
- C = C/R (Command/Response Field)

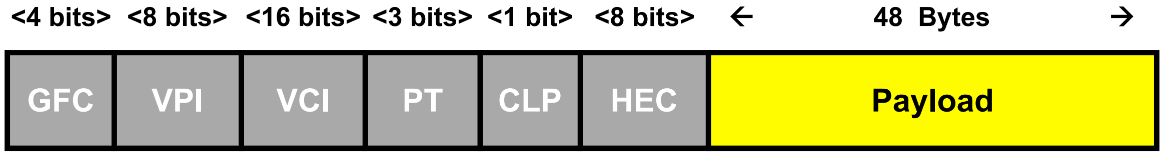
VPWS CRoMPLS—
draft-ietf-pwe3-atm-encap-xx.txt



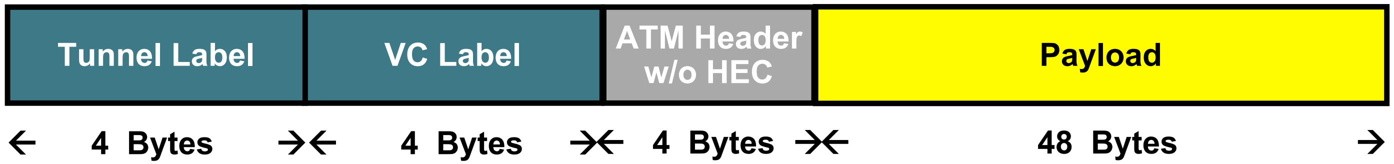
- This is cell relay over MPLS (VC/VP/port mode)
- Single cell is encapsulated; no HEC (52 bytes only)
- Control word is optional
- Control word flags should be set to zero and ignored

VPWS CPKoMPLS—Encapsulation
draft-ietf-pwe3-atm-encap-xx.txt

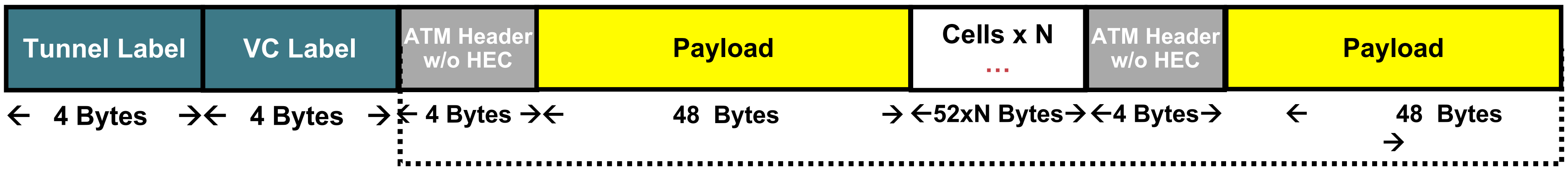
ATM Cell



Single Cell Relay



Packed Cell Relay



Packed Cells Max 28

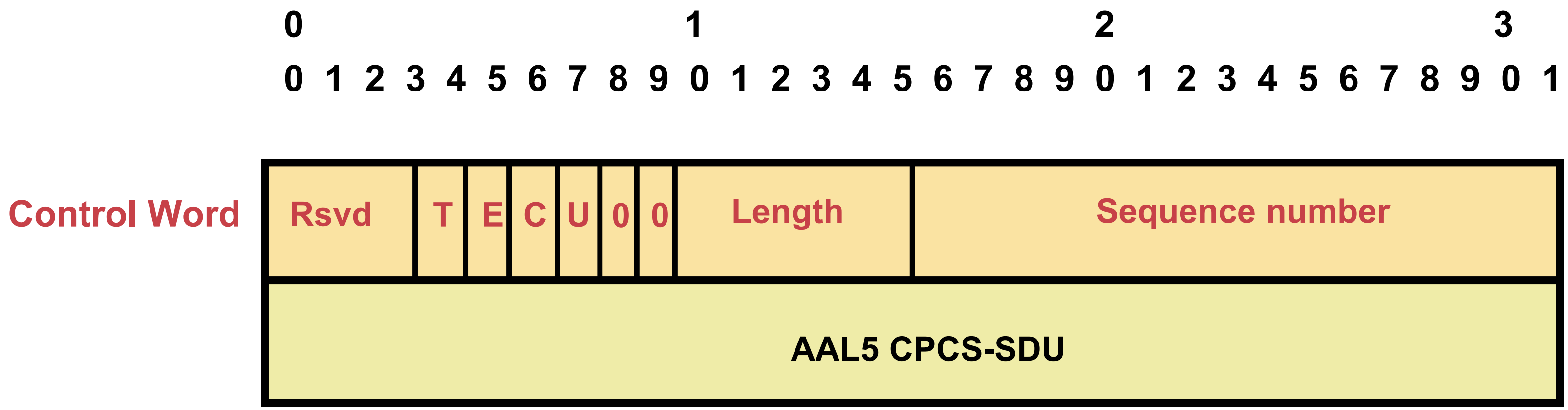
28*52=1456 Bytes

VPWS CPKoMPLS—
draft-ietf-pwe3-atm-encap-xx.txt

CPKoMPLS = Cell Packing over MPLS

- Used to mitigate cell to MPLS packet MTU inefficiencies
- Concatenated ATM cell (52 bytes); no HEC
- Maximum 28 cells per MPLS frame (<1500 byte MTU)
- VC/VP/port mode support
- Cell Packing operation:
    - -Maximum Number of Cells to Pack (MNCP)
    - -Minimum Cell Packing Timer (MCPT)

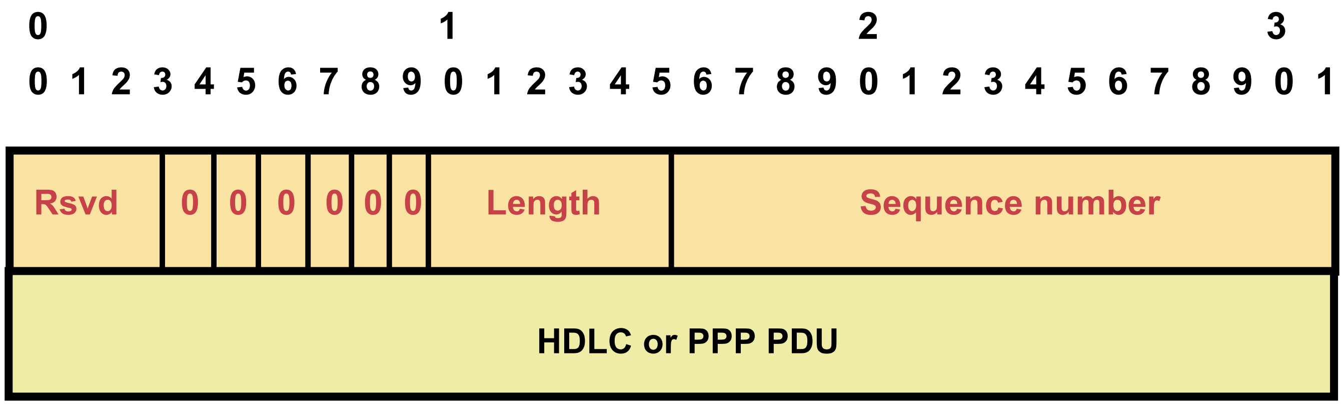
VPWS AAL5oMPLS—
draft-ietf-pwe3-atm-encap-xx.txt



- **AAL5 SDU is encapsulated**
- **Control word is required**
- **Service allows transport of OAM and Resource Management cells**
- **Control word flags encapsulate transport type, EFCI, CLP, C/R bit**

VPWS PPPoMPLS/HDLCoMPLS—
draft-ietf-pwe3-hdlc-ppp-encap-xx.txt

- **Cisco HDLC and PPP PDUs are transported without flags or FCS**
  - **PPP frames also do not carry HDLC address and control information**
- **The control word is optional**



Optional

Frame Format CE — LER

Original Ethernet Frame

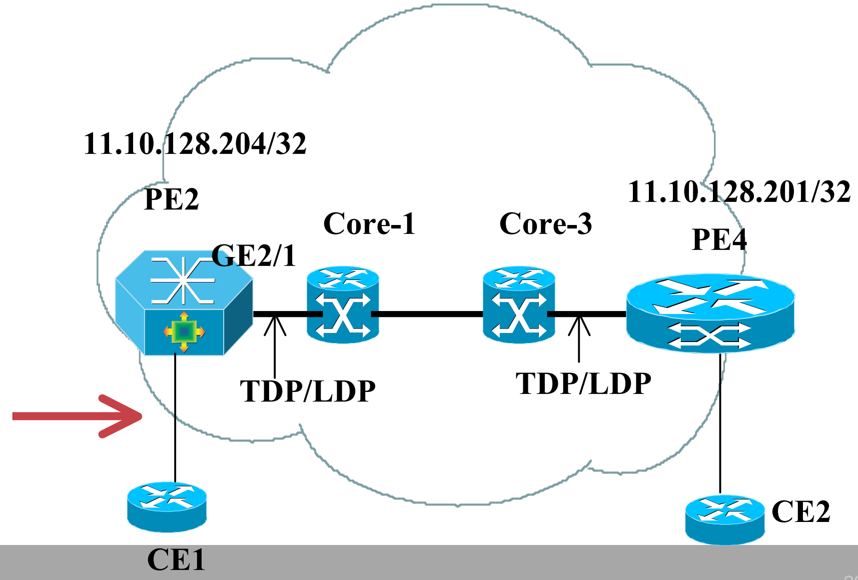
DA SA	8000 V HL TOS …

VLAN Encapsulated Frame

DA SA	8100 Pbits Cbit VLAN ID	Ethernet Frame

4 Byte 802.1q Header

- **2 Byte EtherType Field (8100)**
- **3 P bits**
- **C bit**
- **12 bit VID**



Frame Format LER—LSR

VLAN Encapsulated Frame

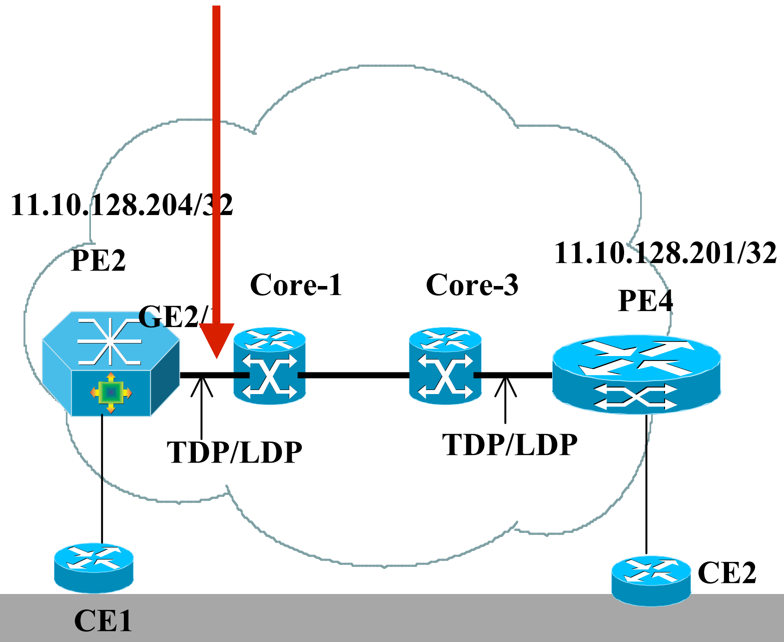
DA SA	8100 Pbits Cbit VLAN ID	Ethernet Frame

MPLS Labeled Packet

DA SA 8847 MPLS LSEs	DA SA	8100 Pbits Cbit VLAN ID	Ethernet Frame

LSE (Label Stack Entries)

- **20 Bit Label**
- **3 Bit Experimental Field (Exp)**
- **1 Bit Bottom of Stack Indicator (S)**
- **1 Byte TTL**

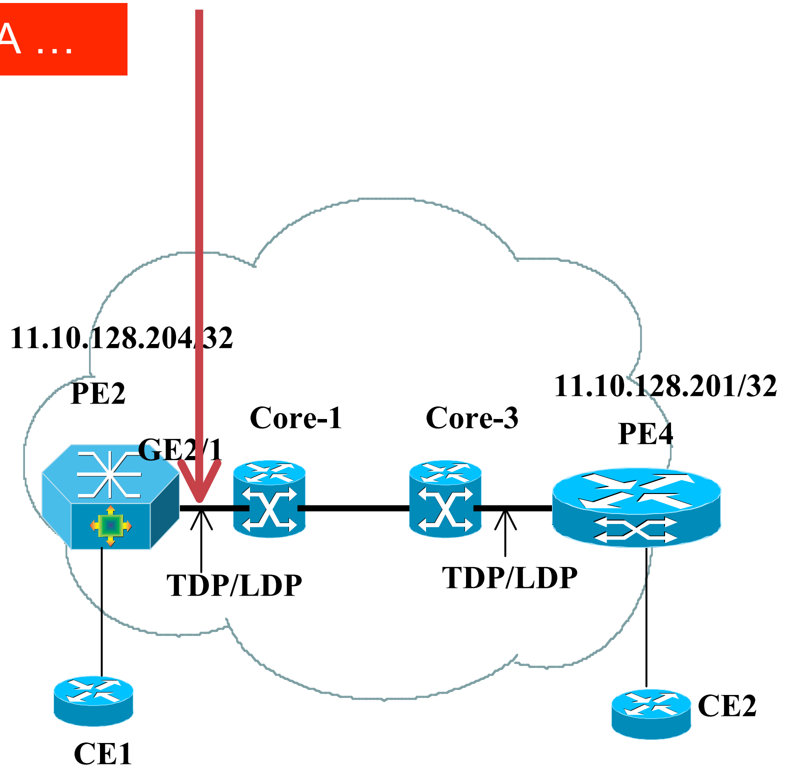


Frame Format LER—LSR (Cont.)

MPLS Labeled Packet

DA SA 8847 00037 0 FE 00012 1 02	DA SA …

- **Tunnel Label Entry**
  - Label 55 (37)
  - Exp = 0
  - S = 0
  - TTL = FE
- **VC Label**
  - Label 18 (12)
  - Exp = 0
  - S=1
  - TTL = 02



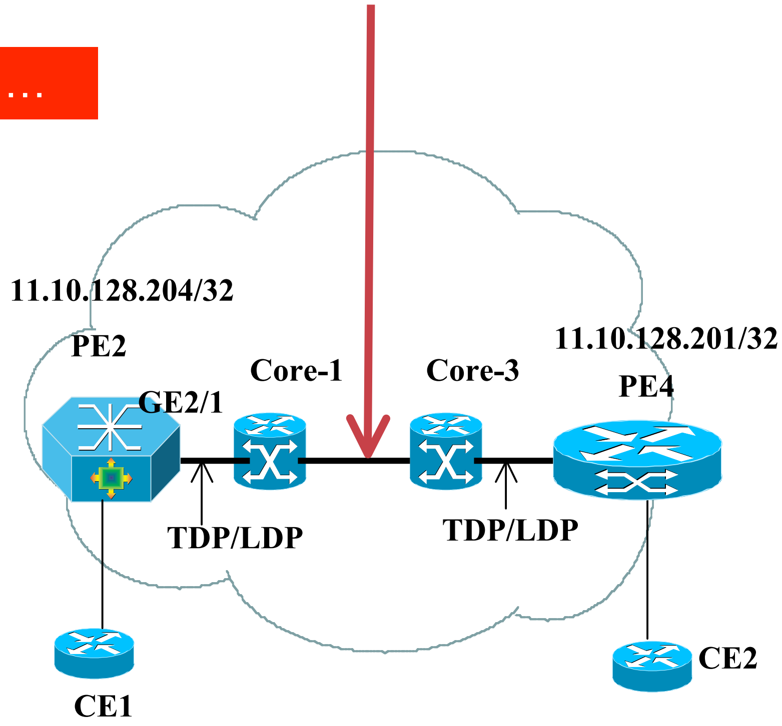
Detaled packet header explanation at: http://www-tac.cisco.com/Teams/NSA/MPLS/EOMPLS/pac1.htm

Frame Format LSR—LSR

MPLS Labeled Packet

DA SA 8847 00088 0 FD 00012 1 02	DA SA …

- **Tunnel Label Entry**
  - **Label 136 (88)**
  - **Exp/S = 0**
  - **TTL = FD**
- **VC Label**
  - **Label 18 (12)**
  - **Exp/S = 1**
  - **TTL = 02**

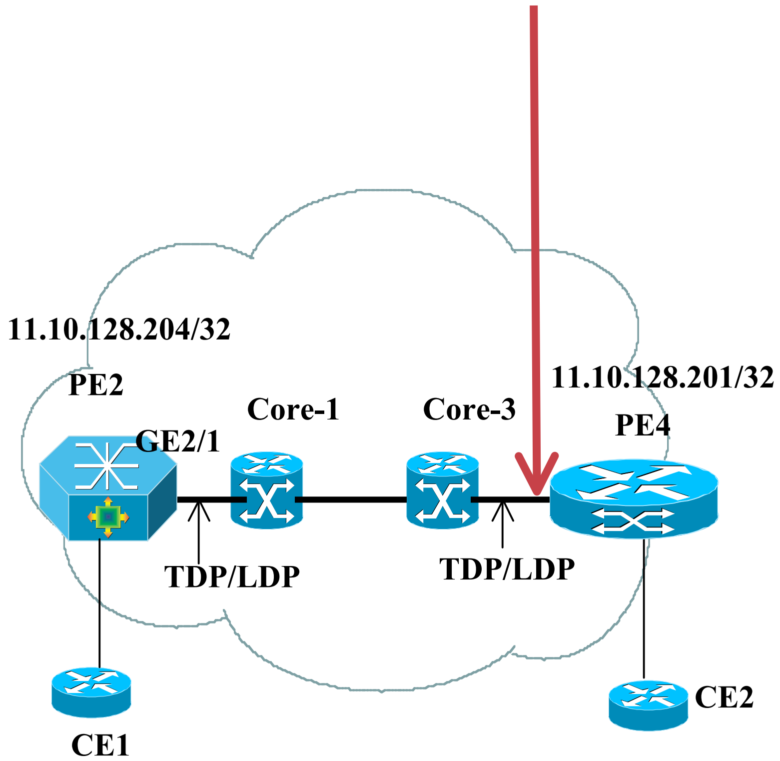


Frame Format LSR—LER

MPLS Labeled Packet

DA SA 8847 00012 1 01	DA SA …

- •VC Label
  - Label 18 (12)
  - Exp/S = 1
  - TTL = 01

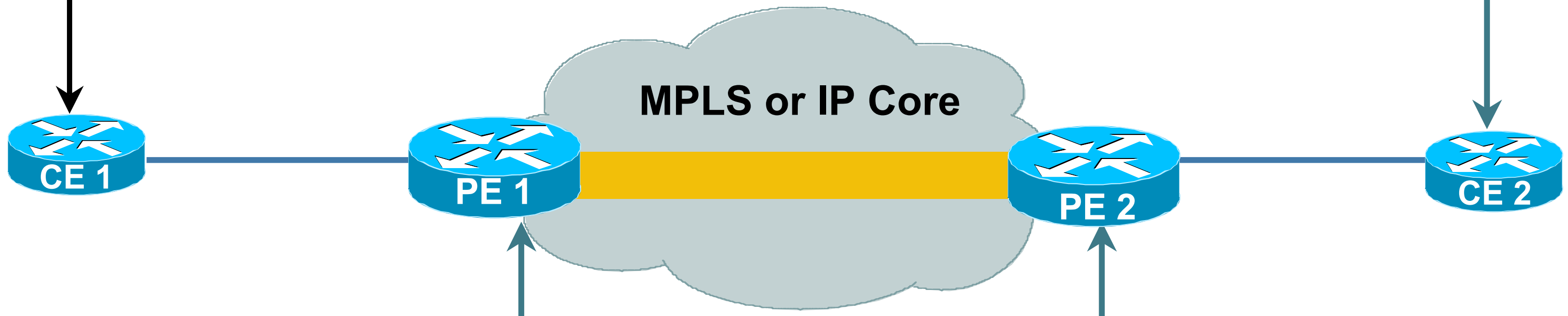


Example: VPWS

Point-to-Point
VLAN over MPLS

interface GigabitEthernet0/0/0/2.1
encapsulation dot1Q 1
ip address 192.168.1.1 255.255.255.0

interface GigabitEthernet1/2.1
encapsulation dot1Q 1
ip address 192.168.1.2 255.255.255.0



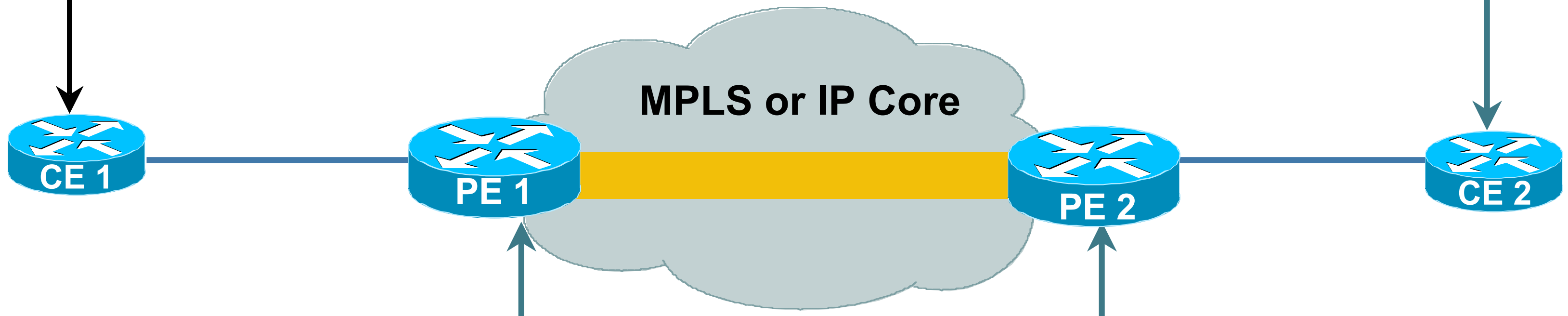
interface GigabitEthernet8/2.1
encapsulation dot1Q 1
xconnect 22.22.22.22 1 encapsulation mpls

interface GigabitEthernet2/2.1
encapsulation dot1Q 1
xconnect 5.5.5.5 1 encapsulation mpls

Point-to-Point
Cell Relay over MPLS

```
interface ATM0/2/0/2.1 point-to-point
 pvc 0/32
 ipv4 address 192.168.2.1 255.255.255.0
```

```
interface ATM3/0.1 point-to-point
pvc 0/32
ip address 192.168.2.2 255.255.255.0
```

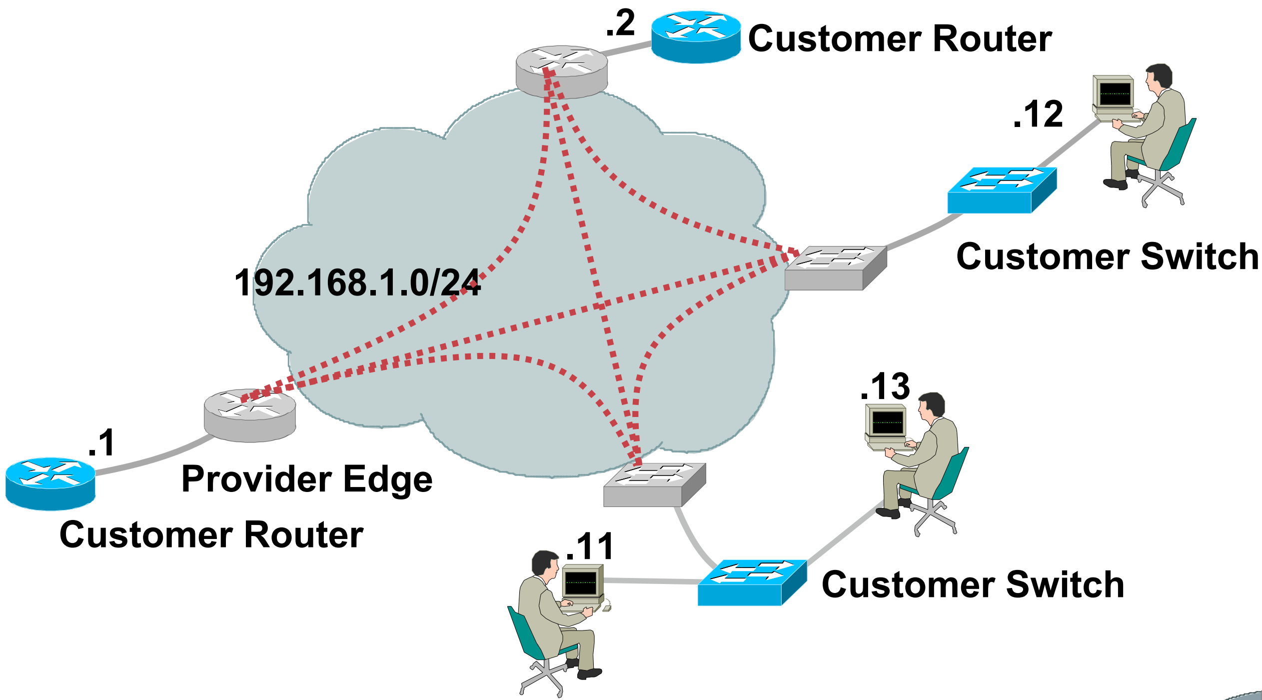



```
interface ATM2/2
pvc 0/32 l2transport
  encapsulation aal0
  xconnect 22.22.22.22 2 encapsulation mpls
```

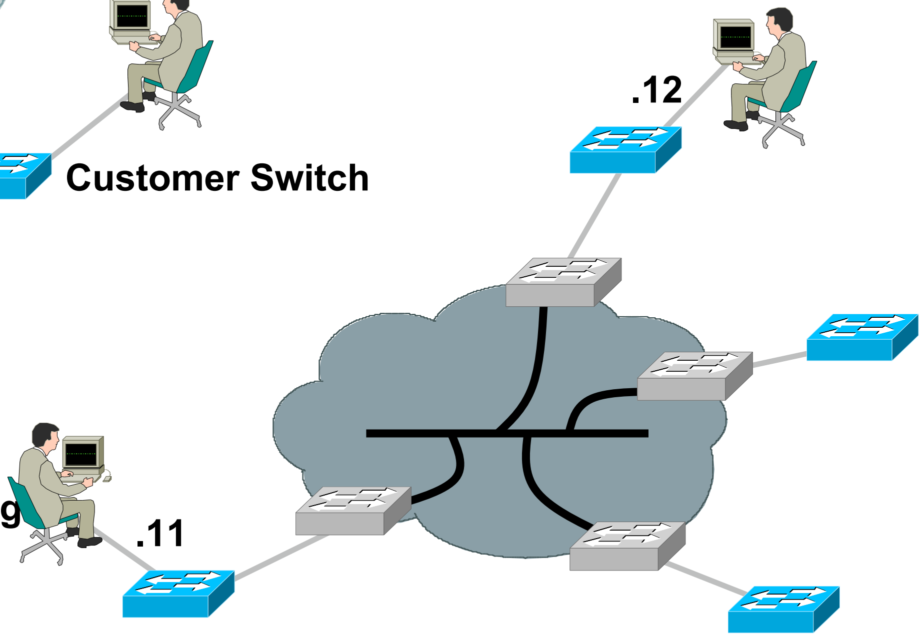
```
interface ATM3/0
 pvc 0/32 l2transport
  encapsulation aal0
  xconnect 5.5.5.5 2 encapsulation mpls
```

Virtual Private LAN Service (VPLS)

VPLS: Customer View



- Single bridge domain (one VLAN)
- Single subnet
- Single SLA
- MAC address learning and forwarding



VPLS—Overview

- **Architecture**

  It is an end-to-end architecture that allows IP/MPLS networks to provide Layer 2 multipoint Ethernet services while using LDP as signaling protocol

- **Bridge emulation**

  Emulates an Ethernet bridge

- **Bridge functions**

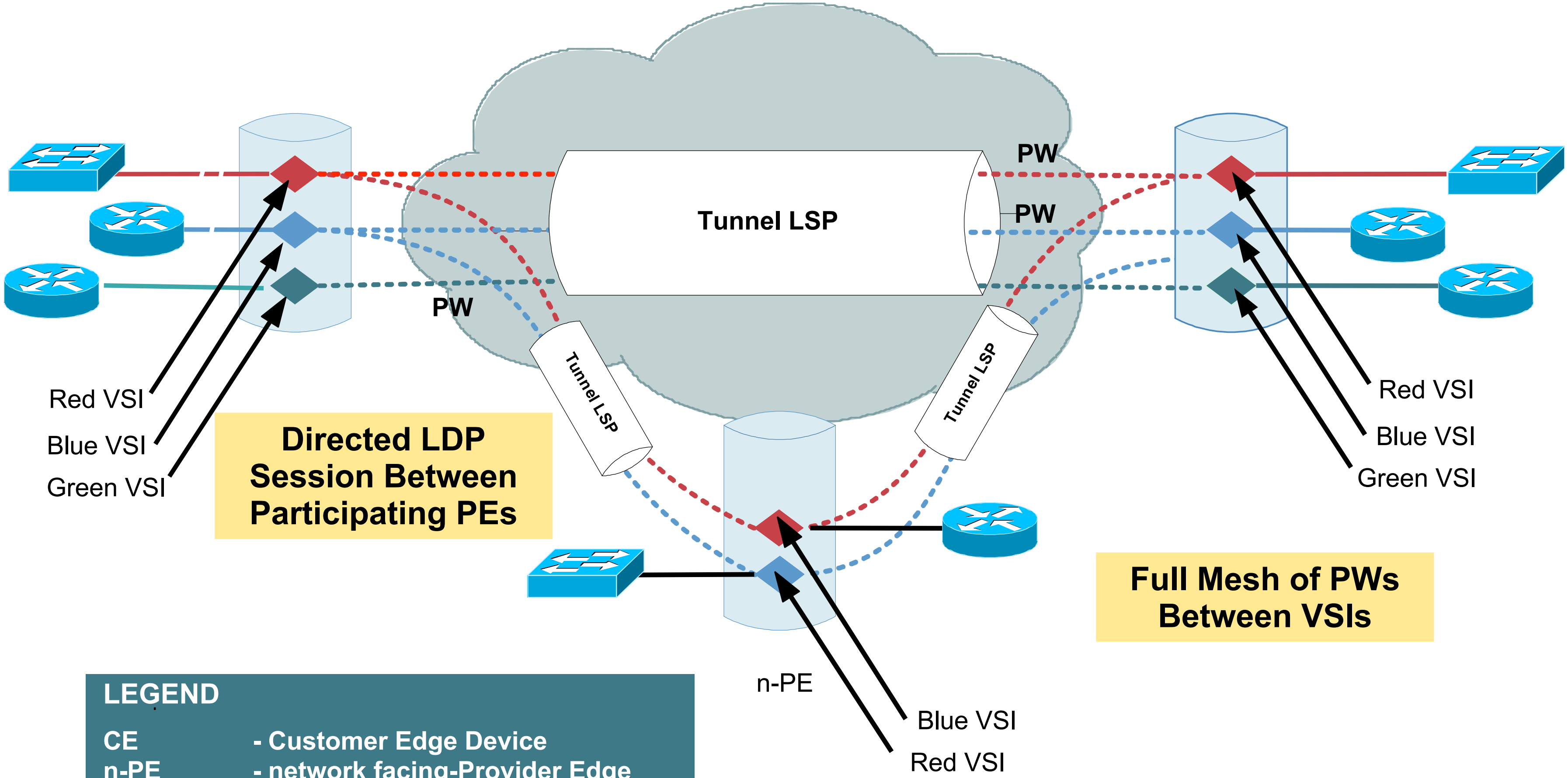
  Operation is the same as for an Ethernet bridge, i.e. forwards using the destination MAC address, learns source addresses and floods broad-/multicast and unknown frames

- **Several drafts in existence**

  draft-ietf-l2vpn-vpls-ldp-xx.txt

  draft-ietf-l2vpn-vpls-bgp-xx-txt

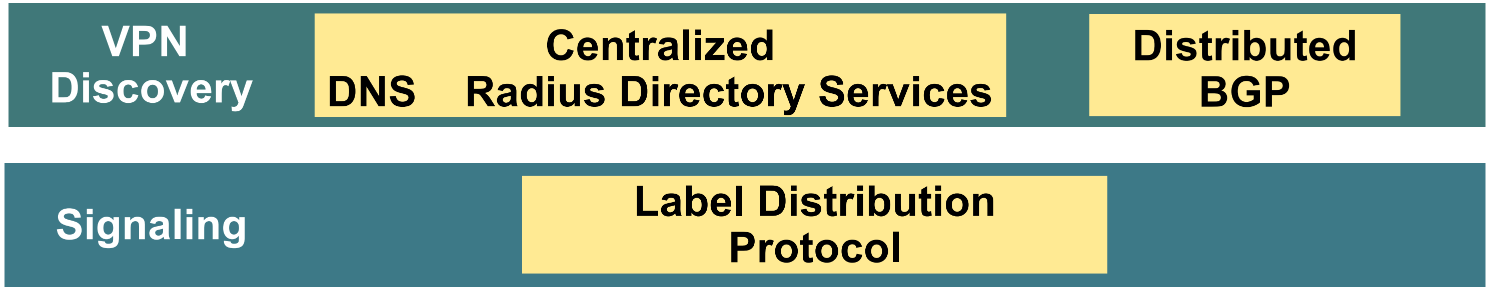
VPLS Components



LEGEND

CE - Customer Edge Device
n-PE - network facing-Provider Edge
VSI - Virtual Switch Instance
PW - Pseudo-Wire
Tunnel LSP - Tunnel Label Switch Path that provides PW transport

VPLS Auto-Discovery and Signaling



- **Draft-ietf-l2vpn-vpls-ldp-01 does not mandate an auto-discovery protocol**
  - **Can be BGP, RADIUS, DNS based**
- **Draft-ietf-l2vpn-vpls-ldp-01 describes using Targeted LDP for Label exchange and PW signaling**
  - **PWs signal other information such as attachment circuit state, sequencing information, etc.**

VPLS: Layer 2 Forwarding Instance Requirements

A Virtual Switch **Must** Operate Like a Conventional L2 Switch!

Flooding/Forwarding:

- **MAC table instances per customer and per customer VLAN (L2-VRF idea) for each PE**
- **VSI will participate in learning, forwarding process**
- **Uses Ethernet VC-Type defined in pwe3-control-protocol-xx**

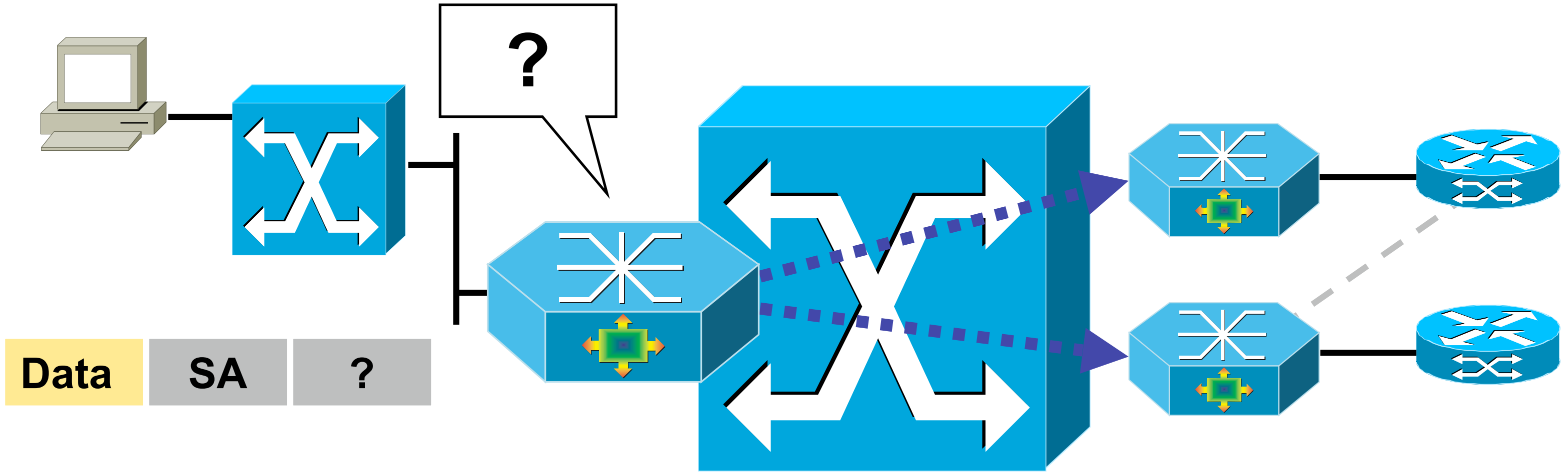
Address Learning/Aging:

- **Self-learn source MAC to port associations**
- **Refresh MAC timers with incoming frames**
- **New additional MAC TLV to LDP**

Loop Prevention:

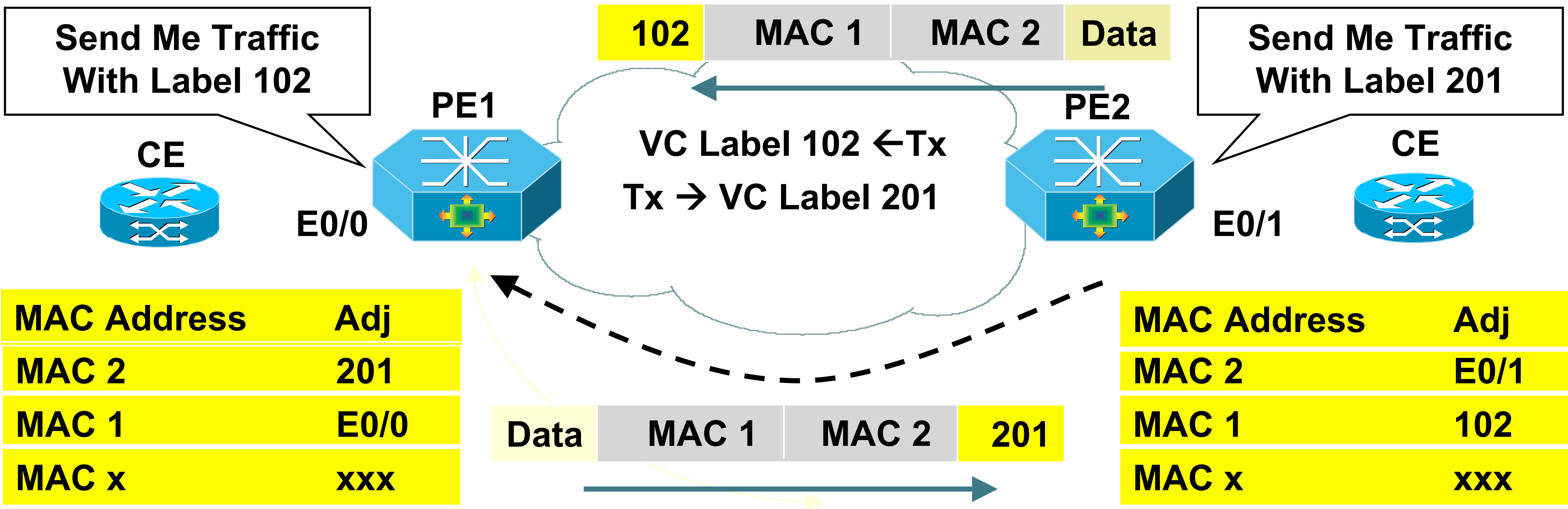
- **Create partial or full-mesh of EoMPLS VCs per VPLS**
- **Use "split horizon" concepts to prevent loops**

VPLS Overview:
Flooding and Forwarding



- **Flooding (Broadcast, Multicast, Unknown Unicast)**
- **Dynamic learning of MAC addresses on PHY and VCs**
- **Forwarding**
  - **Physical port**
  - **Virtual circuit**

VPLS Overview:
MAC Address Learning

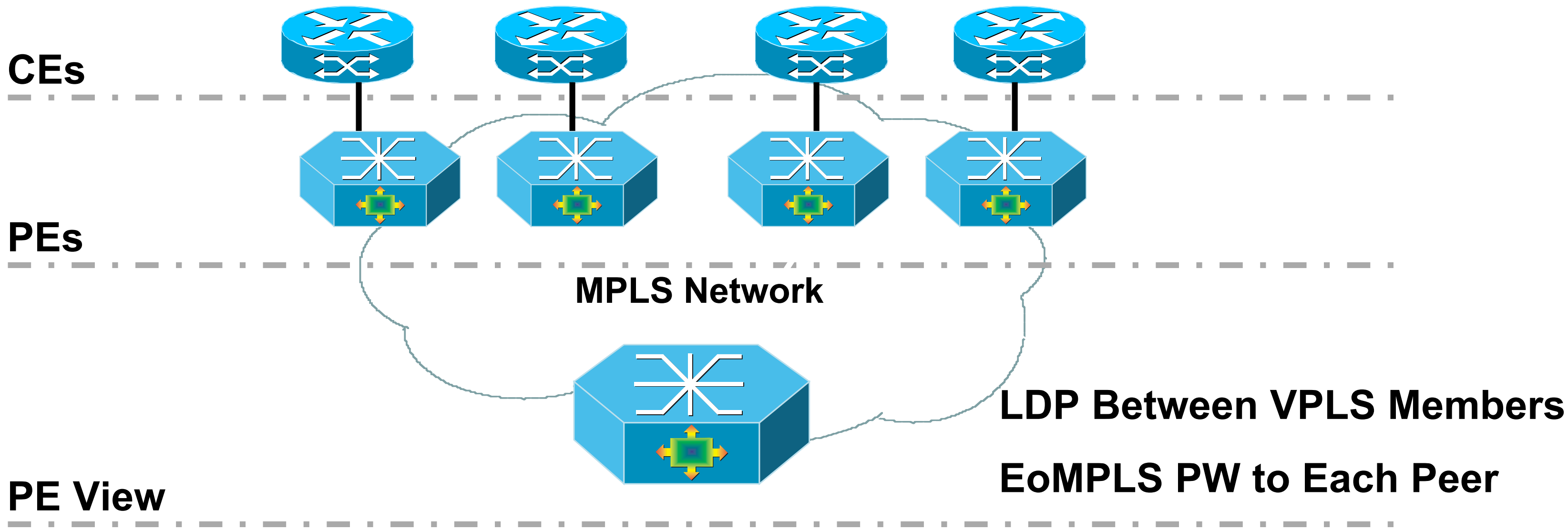


MAC Address	Adj
MAC 2	201
MAC 1	E0/0
MAC x	xxx

MAC Address	Adj
MAC 2	E0/1
MAC 1	102
MAC x	xxx

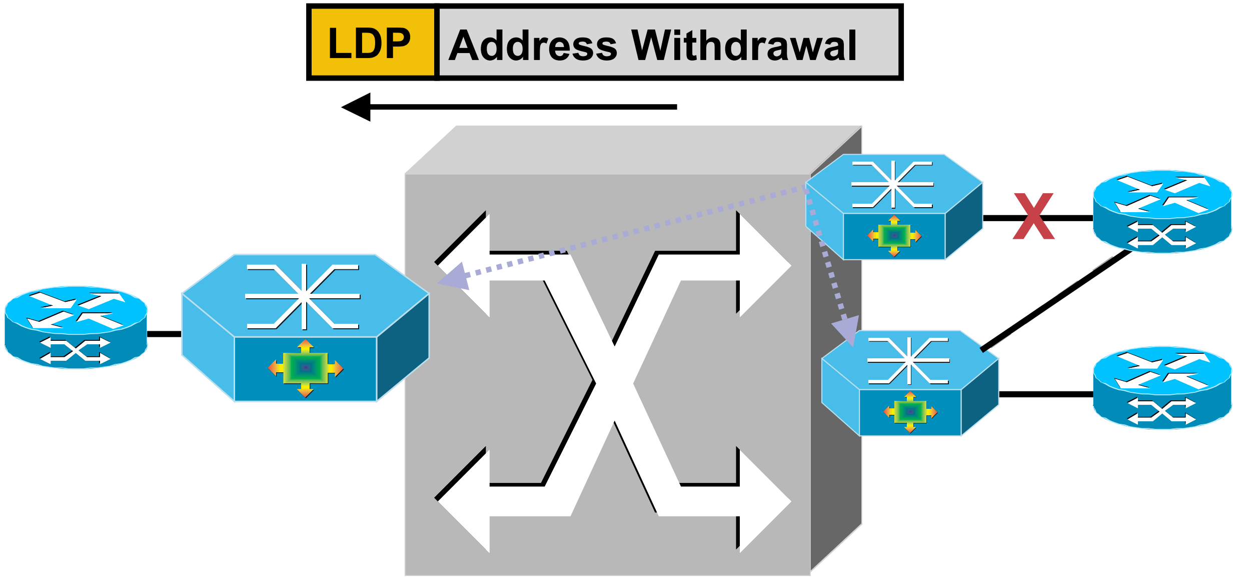
- Broadcast, multicast, and unknown unicast are learned via the received label associations
- Two LSPs associated with an VC (Tx and Rx)
- If inbound or outbound LSP is down, then the entire circuit is considered down

VPLS Overview:
VPLS Loop Prevention



- **Each PE has a P2MP view of all other PEs it sees it self as a root bridge, split horizon loop protection**
- **Full mesh topology obviates STP requirements in the service provider network**
- **Customer STP is transparent to the SP/customer BPDUs are forwarded transparently**
- **Traffic received from the network will not be forwarded back to the network**

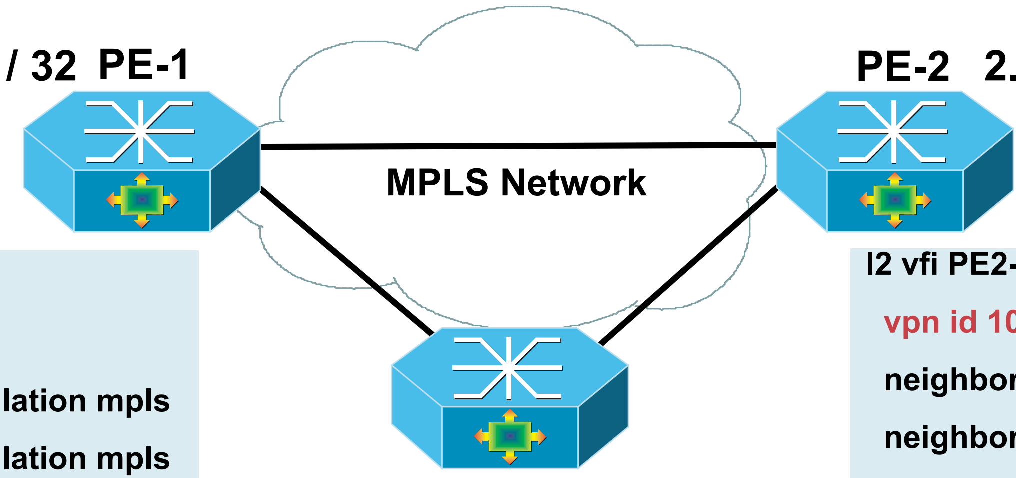
VPLS Overview: MAC Address Withdrawal



- **Primary link failure triggers notification message**
- **PE removes any locally learned MAC addresses and sends LDP address withdrawal (RFC3036) to remote PEs in VPLS**
- **New MAC TLV is used**

VPLS: Configuration Example
PE → PE

Create a L2 VFI with a Full Mesh of Participating VPLS PE Nodes



PE-1:

```
l2 vfi PE1-VPLS-A manual
 vpn id 100
 neighbor 2.2.2.2 encapsulation mpls
 neighbor 3.3.3.3 encapsulation mpls
!
Interface loopback 0
 ip address 1.1.1.1 255.255.255.255
```

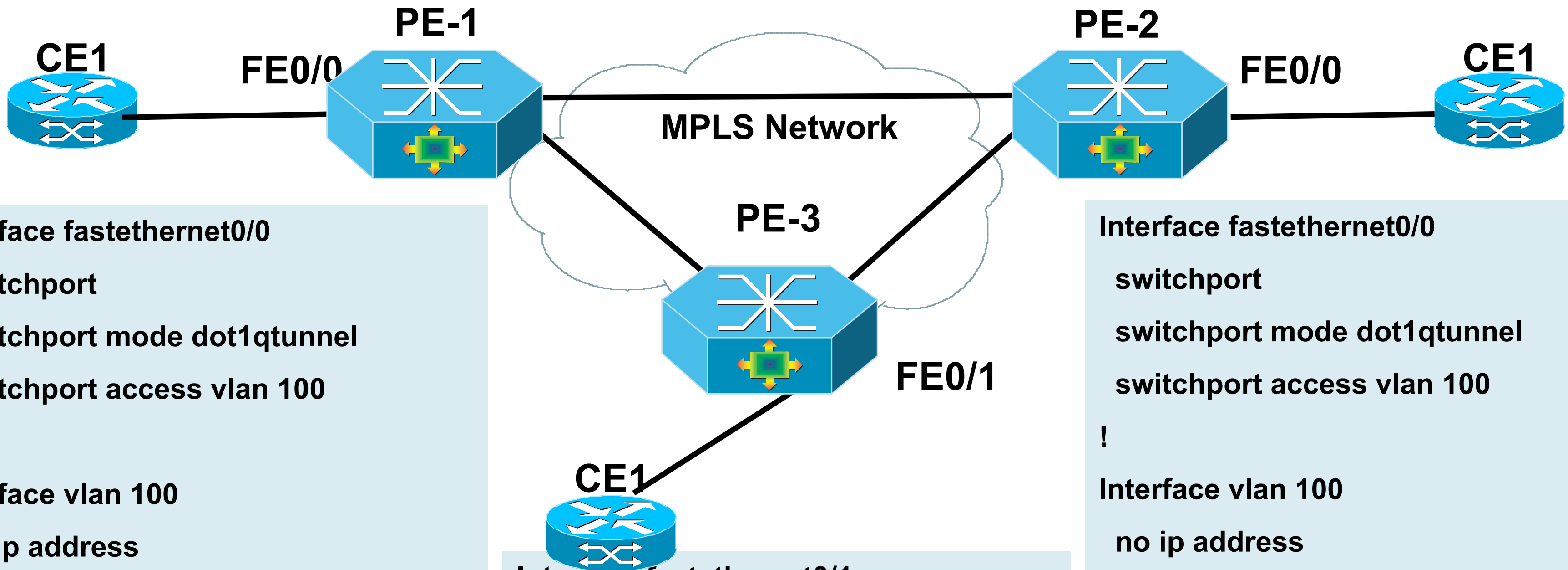
PE-2:

```
l2 vfi PE2-VPLS-A manual
 vpn id 100
 neighbor 1.1.1.1 encapsulation mpls
 neighbor 3.3.3.3 encapsulation mpls
!
Interface loopback 0
 ip address 2.2.2.2 255.255.255.255
```

PE-3:

```
l2 vfi PE3-VPLS-A manual
 vpn id 100
 neighbor 1.1.1.1 encapsulation mpls
 neighbor 2.2.2.2 encapsulation mpls
!
Interface loopback 0
 ip address 3.3.3.3 255.255.255.255
```

VPLS: Configuration Example
PE → CE

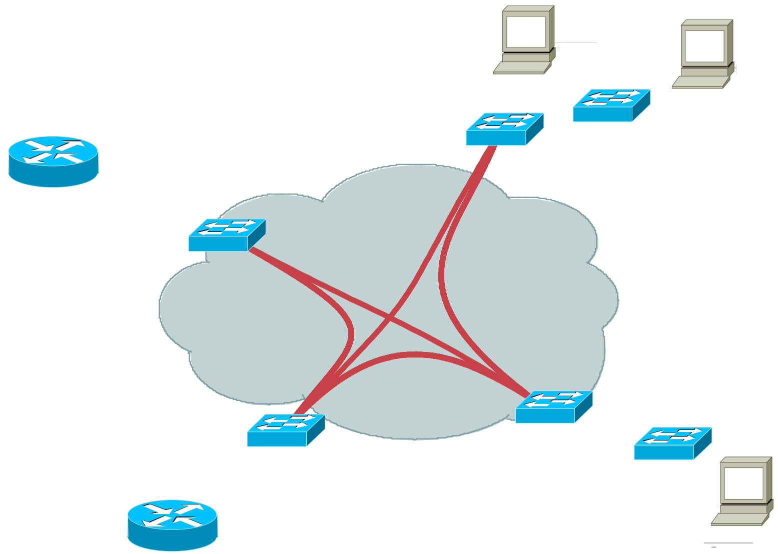



```
Interface fastethernet0/0
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
!
Interface vlan 100
  no ip address
 xconnect vfi PE1-VPLS-A
!
vlan 100
  state active
```

```
Interface fastethernet0/1
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
!
Interface vlan 100
  no ip address
 xconnect vfi PE3-VPLS-A …etc.
```

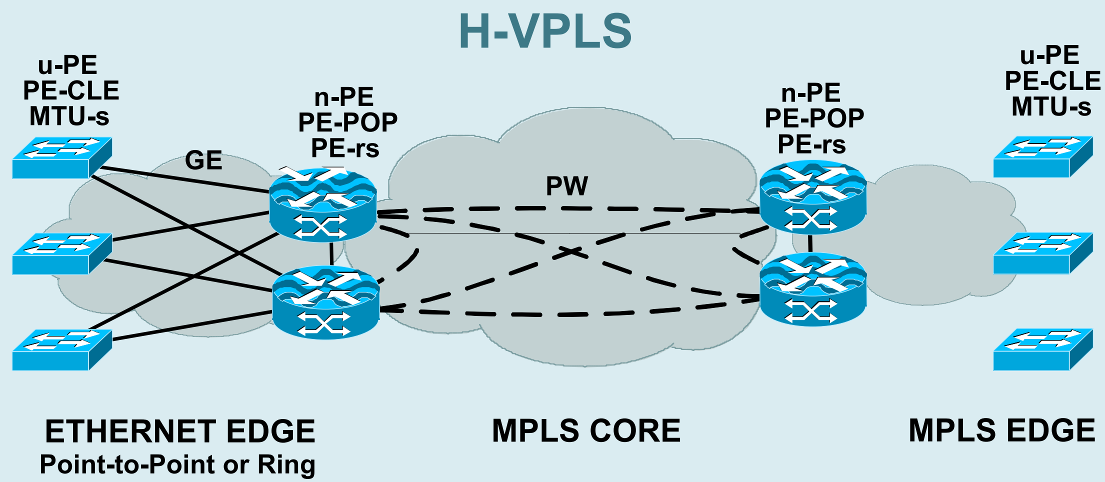
```
Interface fastethernet0/0
  switchport
  switchport mode dot1qtunnel
  switchport access vlan 100
!
Interface vlan 100
  no ip address
 xconnect vfi PE2-VPLS-A
!
vlan 100
  state active
```

VPLS and H-VPLS



- **VPLS**
  - **Single flat hierarchy**
  - **MPLS to the EDGE**

- **H-VPLS**
  - **Two Tier Hierarchy**
  - **MPLS or Ethernet Edge**
  - **MPLS Core**



QOS IN MPLS NETWORKS

Prerequisites

- **Basic understanding of MPLS (L3VPN, L2VPN, TE)**
- **Basic understanding of QoS (DiffServ)**

Agenda

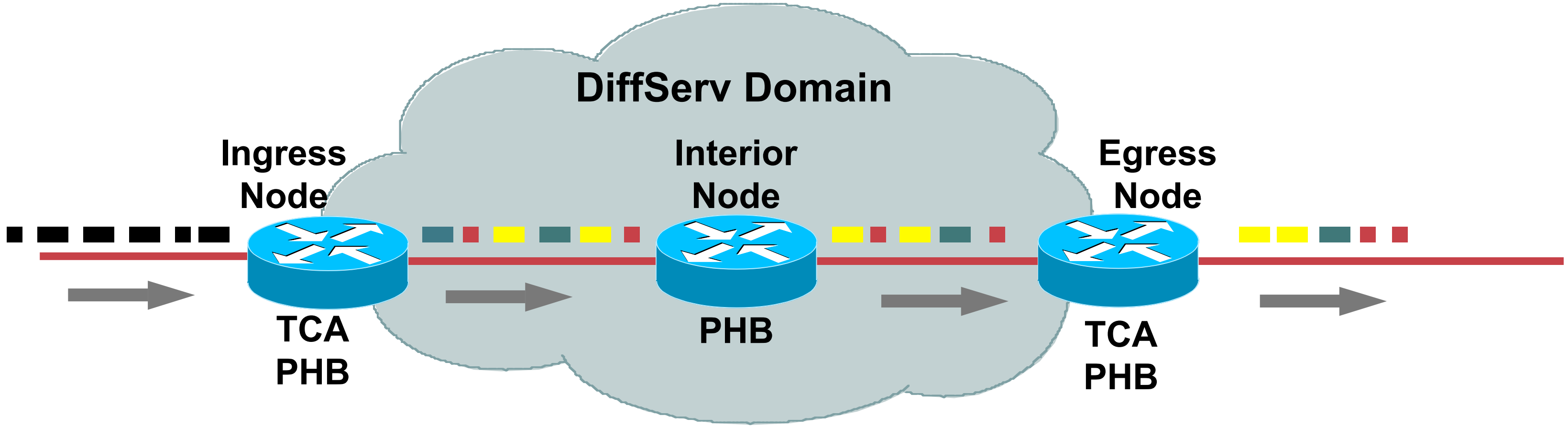
- Technology Overview
- Backbone Infrastructure
- IP Services
- Layer-2 Services
- Interprovider QoS
- Management

MPLS QOS
TECHNOLOGY OVERVIEW

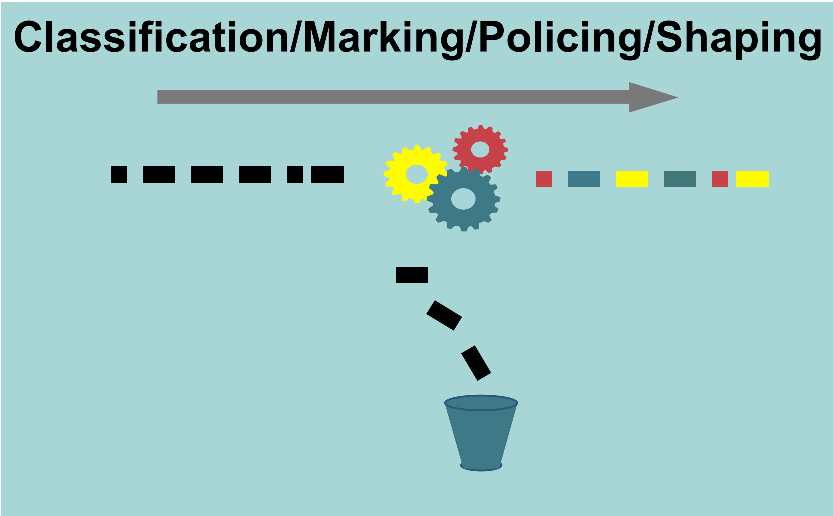
MPLS QoS Architectures

- MPLS does NOT define new QoS architectures
- MPLS QoS uses Differentiated Services (DiffServ) architecture defined for IP QoS
- DiffServ architecture defined in RFC2475
- MPLS support for DiffServ defined in RFC3270

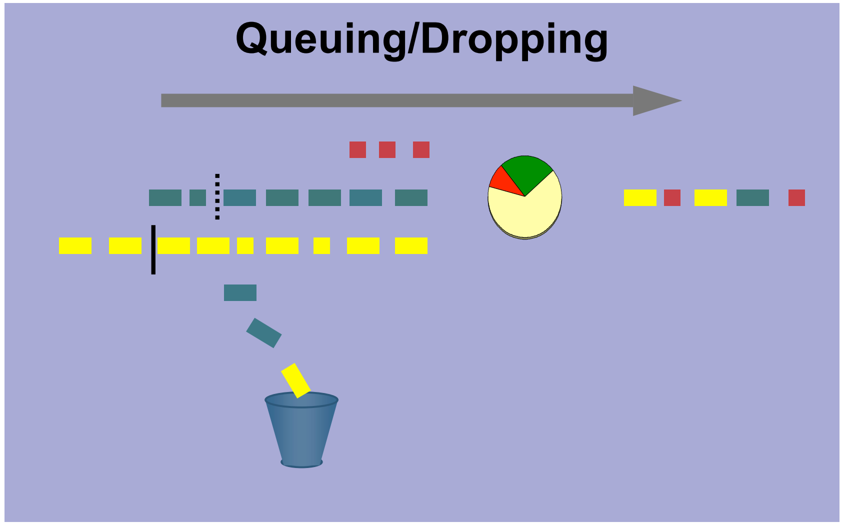
Differentiated Services Architecture



Traffic Conditioning Agreement (TCA)



Per-Hop Behavior (PHB)



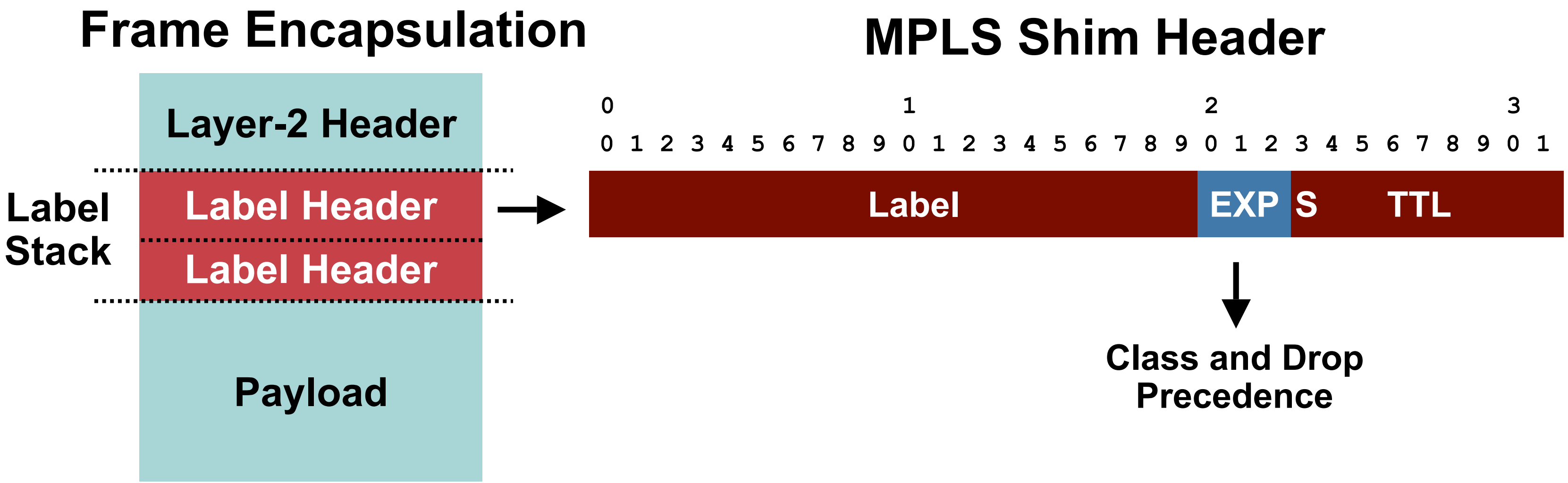
What's Unchanged in MPLS Support of DiffServ

- **Functional components (TCA/PHB) and where they are used**
  - Classification, marking, policing, and shaping at network boundaries
  - Buffer management and packet scheduling mechanisms used to implement PHB
- **PHB definitions**
  - Expedited Forwarding (EF): low delay/jitter/loss
  - Assured Forwarding (AF): low loss
  - Default (DF): No guarantees (best effort)

What's New in MPLS Support of DiffServ

- **How aggregate packet classification is conveyed (E-LSP vs. L-LSP)**
- **Interaction between MPLS DiffServ info and encapsulated DiffServ info (e.g. IP DSCP)**

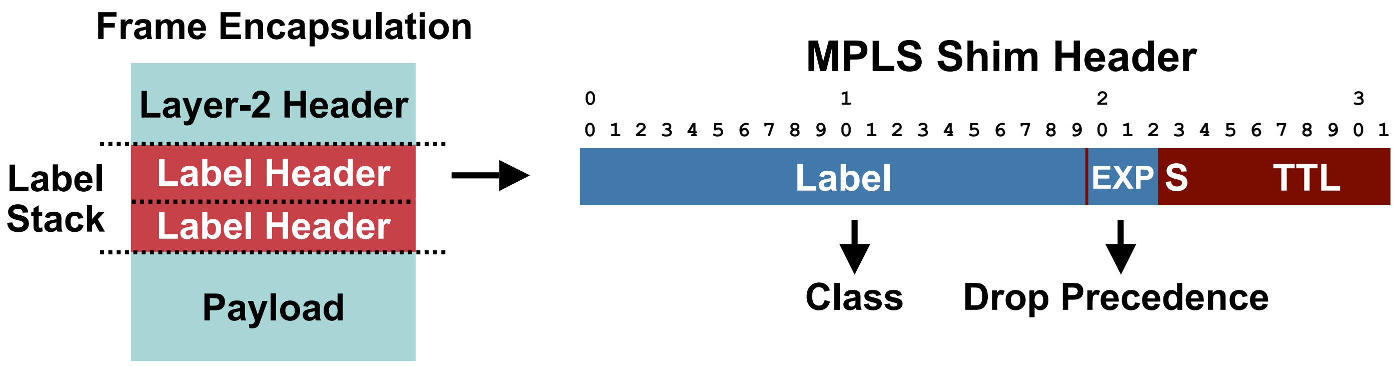
EXP-Inferred-PSC* LSP (E-LSP)

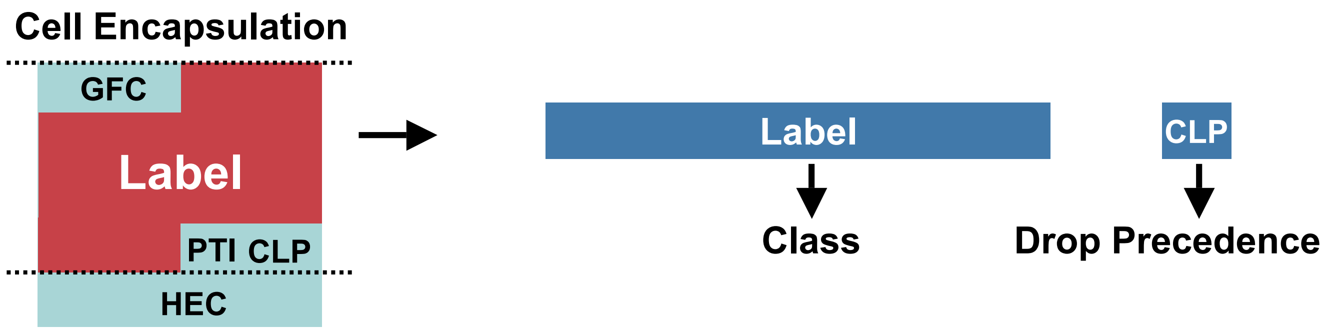


- Packet Class and drop precedence inferred from EXP (3-bit) field
- RFC3270 does not recommend specific EXP values for DiffServ PHB (EF/AF/DF)
- Used for frame-based MPLS

*Per-Hop Behavior Scheduling Class

Label-Only-Inferred-PSC* LSP (L-LSP)





- Packet class inferred from label
- Drop precedence inferred from EXP or ATM CLP
- Can be used for frame-based and cell-based MPLS

*Per-Hop Behavior Scheduling Class

E-LSP vs. L-LSP

- **An E-LSP may carry multiple classes (max eight, in real life less than that)**
- **An L-LSP carries one class**
- **Both E-LSP and L-LSP can use LDP or RSVP for label distribution**
- **Cisco products currently support E-LSP for frame-mode MPLS**
- **No demand for L-LSP support with frame-mode MPLS yet**

MPLS Support of DiffServ: All Done with Modular QoS CLI (MQC)

```
class-map [match-any | match-all] class-name
```

Enters Configuration Sub-mode for Class Definition

```
policy-map policy-name
```

Enters Configuration Sub-Mode for Policy Definition (Marking, Policing, Shaping, Queuing, Etc.)

```
service-policy {input | output} policy-name
```

Command in Interface Configuration Sub-Mode fo Apply QoS Policy for Input or Output Traffic

```
class-map match-all REAL-TIME
 match mpls experimental topmost 5
class-map match-all PREMIUM
 match mpls experimental topmost 1  2
!
!
policy-map OUT-POLICY
 class REAL-TIME
  priority percent 25
 class PREMIUM
  bandwidth remaining percent 50
  random-detect
 class class-default
  random-detect
!
interface POS1/0
 ip address 10.150.1.1 255.255.255.0
 service-policy output OUT-POLICY
!
```

- Template-based command syntax for QoS
- Separates classification engine from QoS functionality
- Platform-independent CLI for QoS features

MQC Snapshot

```
class-map [match-any | match-all] class-name
```

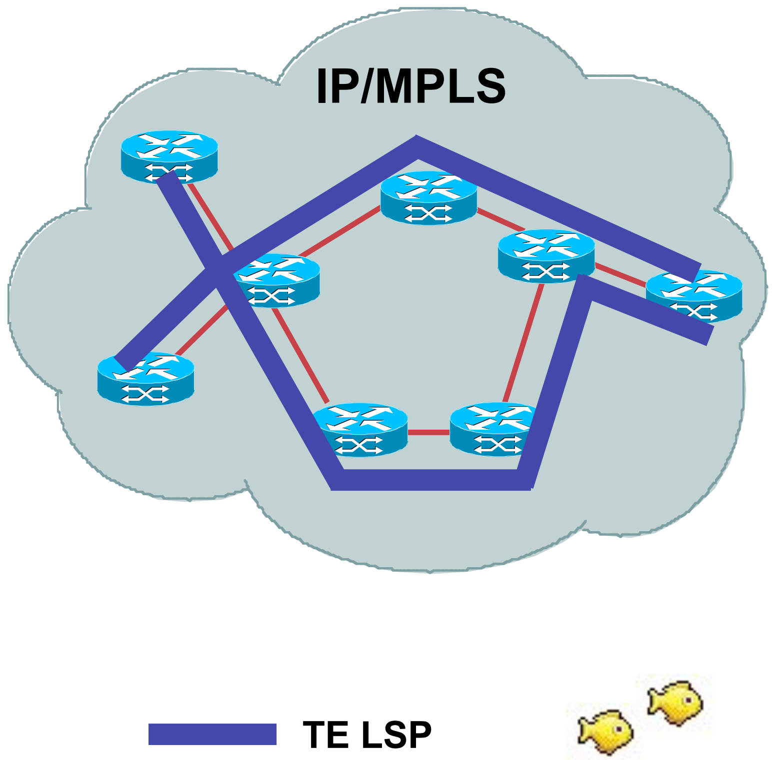
```
match { access-group { n | name n } | any | atm { clp | oam } | cos c | dscp d |
        fr-de | fr-dlci d | ip { dscp d | precedence p } | mpls exp e |
        precedence p | qos-group g | vlan v |
        protocol { arp | cdp | clns | clns_es | clns_is |
                   cmns | compressedtcp | ip | ipv6 } }
```

```
policy-map policy-name
```

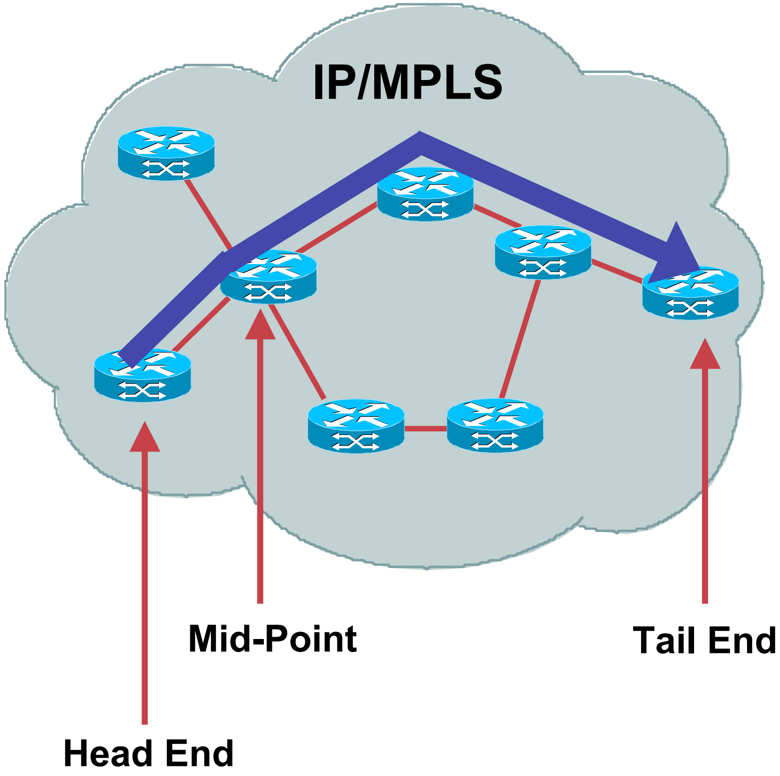
```
bandwidth {rate | percent p | remaining percent p }
police rate { r | percent p } [ burst b ] [ peak-rate { r | percent p } [ peak-burst b ]]
priority [ r [ b ]]
queue-limit l {packets cells ms us}
random-detect { discard-class-based | dscp-based | prec-based }
service-policy  p
set { dscp d | ip { dscp d | precedence p } | mpls exp { topmost e | imposition e } |
      cos c | discard-class d | fr-de f | qos-group q }
shape average { r |  percent p }
```

MPLS TE Overview

- Introduces explicit routing
- Supports constrained-based routing
- Supports admission control
- Protection capabilities
- RSVP-TE to establish LSPs
- ISIS and OSPF extensions to advertise link attributes
- Lots more in session RST-3110

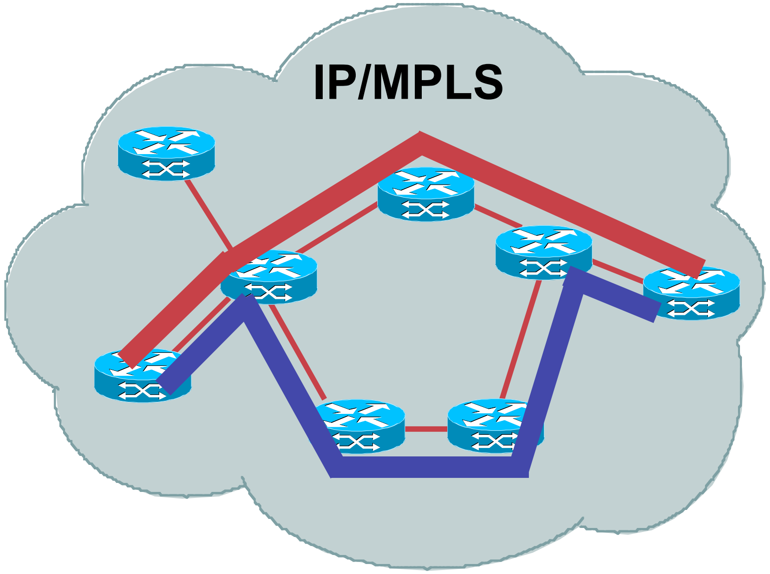


How MPLS TE Works



- **Information distribution**
  - **ISIS-TE**
  - **OSPF-TE**
- **Path calculation (CSPF)**
- **Path setup (RSVP-TE)**
- **Forwarding traffic down tunnel**
  - **Auto-route**
  - **Static**
  - **Policy-Based routing**
  - **Class-Based tunnel selection**
  - **Forwarding adjacency**
  - **Tunnel select**

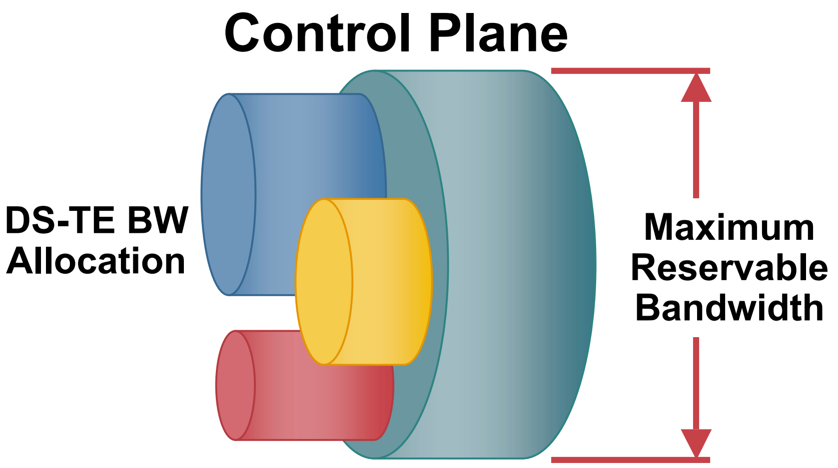
DiffServ-Aware Traffic Engineering (DS-TE)

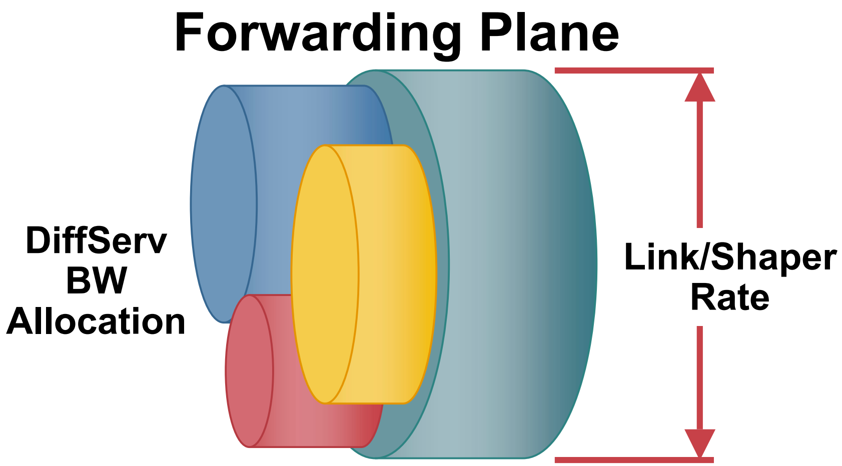


Low-Latency TE LSP with Reserved BW
Best-Effort TE LSP

- Brings per-class dimension to MPLS TE
  - Per-Class constrained-based routing
  - Per-Class admission control

DiffServ-Aware Traffic Engineering (DS-TE)

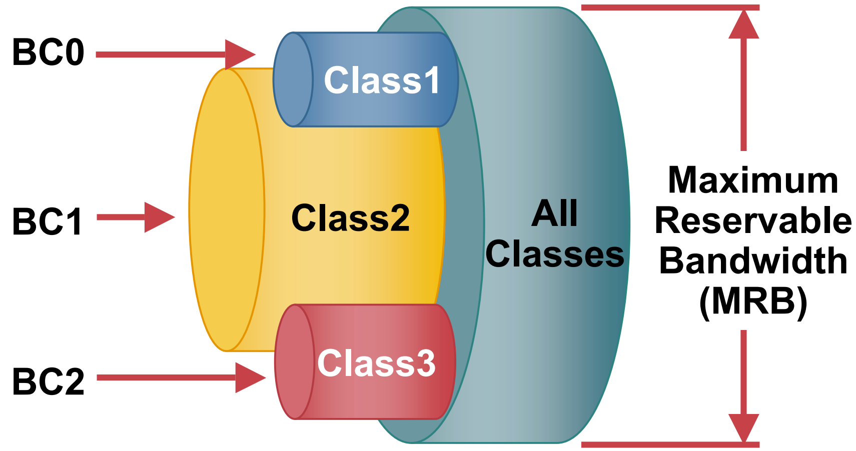




- Link BW distributed in pools or Bandwidth Constrains (BC)
- Up to eight BW pools
- Different BW pool models
- Unreserved BW per TE class computed using BW pools and existing reservations
- Unreserved BW per TE class advertised via IGP

DS-TE Bandwidth Pools: Maximum Allocation Model (MAM)

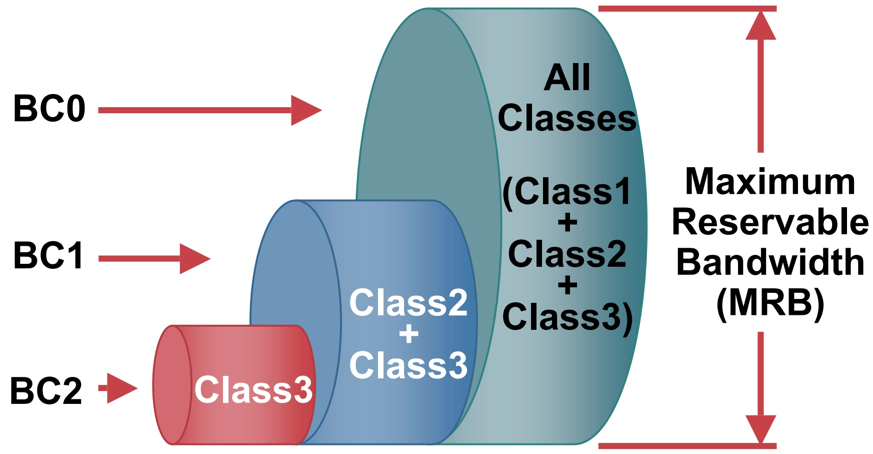
- **BW pool applies to one class**
- **Sum of BW pools may exceed MRB**
- **Sum of total reserved BW may not exceed MRB**



BC0: 20% Best Effort
BC1: 50% Premium
BC2: 30% Voice

DS-TE Bandwidth Pools: Russian Dolls Model (RDM)

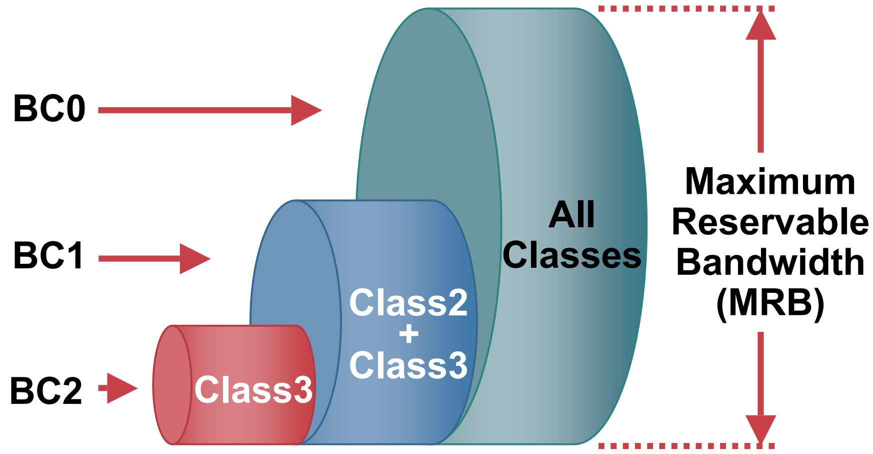
- **BW pool applies to one or more classes**
- **Global BW pool (BC0) equals MRB**
- **BC0..BCn used for computing unreserved BW for class n**



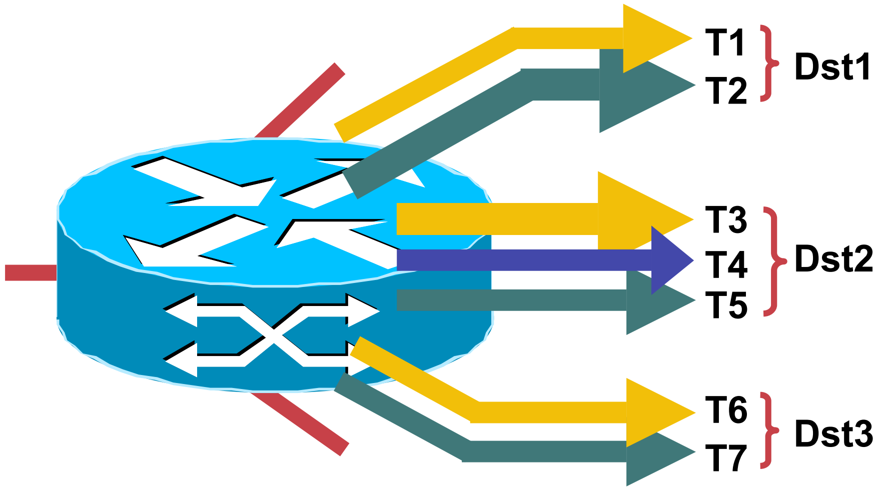
BC0: MRB	**Best Effort + Premium + Voice**
BC1: 50%	**Premium + Voice**
BC2: 30%	**Voice**

DS-TE Bandiwdth Pools: Why Russian Dolls Model?

- **Good match for common bandwidth allocation in forwarding plane**
  - VoIP gets priority treatment and is unaffected by other traffic: use BC2
  - Business data gets preferential access to link vs. BE: use BC1
  - Best effort may use MRB if other classes not fully used, but should be reduced if lots of VOIP or Business Data: use BC0
- **Good isolation between classes, efficient use of bandwidth**



Class-Based Tunnel Selection: CBTS



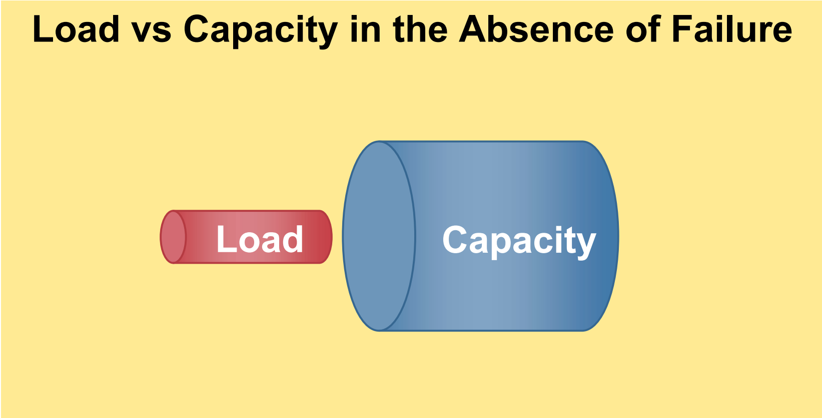
FIB

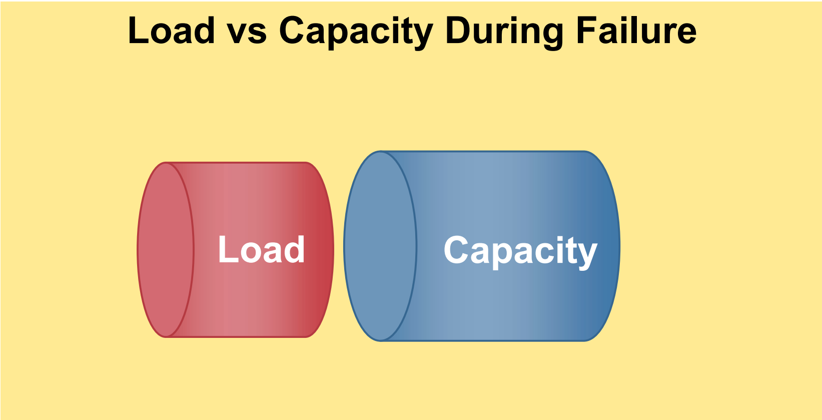
Dst1, exp 4	Tunnel1
Dst1, *	Tunnel2
Dst2, exp 4	Tunnel3
Dst2, exp 2	Tunnel4
Dst2, *	Tunnel5
Dst3, exp 4	Tunnel6
Dst3, *	Tunnel7

*Wildcard EXP Value

- EXP-based selection between multiple tunnels to same destination
- Local mechanism to head-end
- Tunnels configured with EXP values to carry
- Tunnels may be configured as default
- No IGP extensions
- Supports VRF traffic
- Simplifies use of DS-TE tunnels
- Similar operation to ATM/FR VC bundles

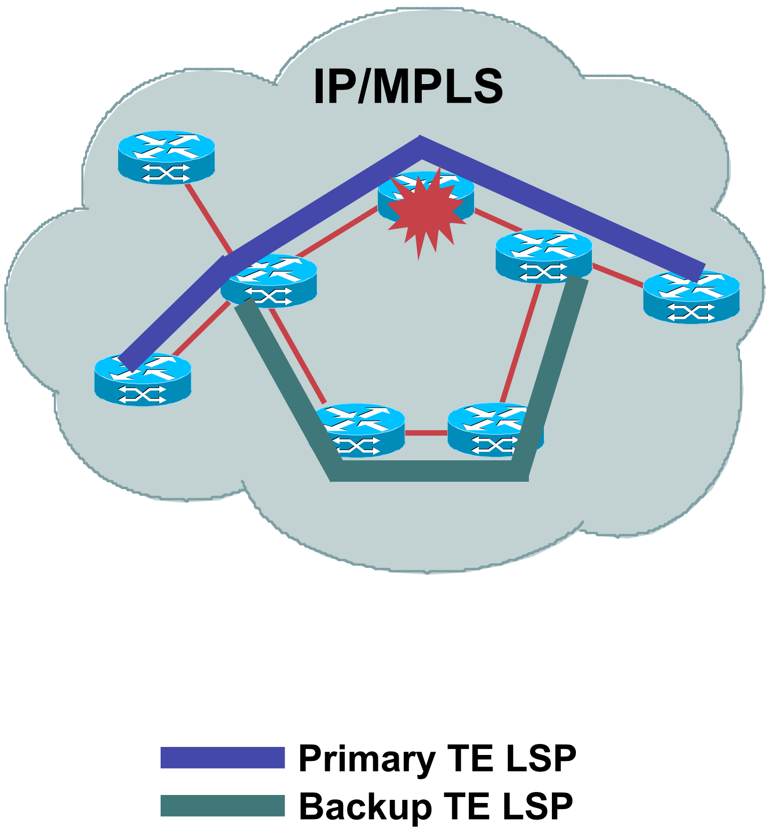
Dealing with Failure Scenarios





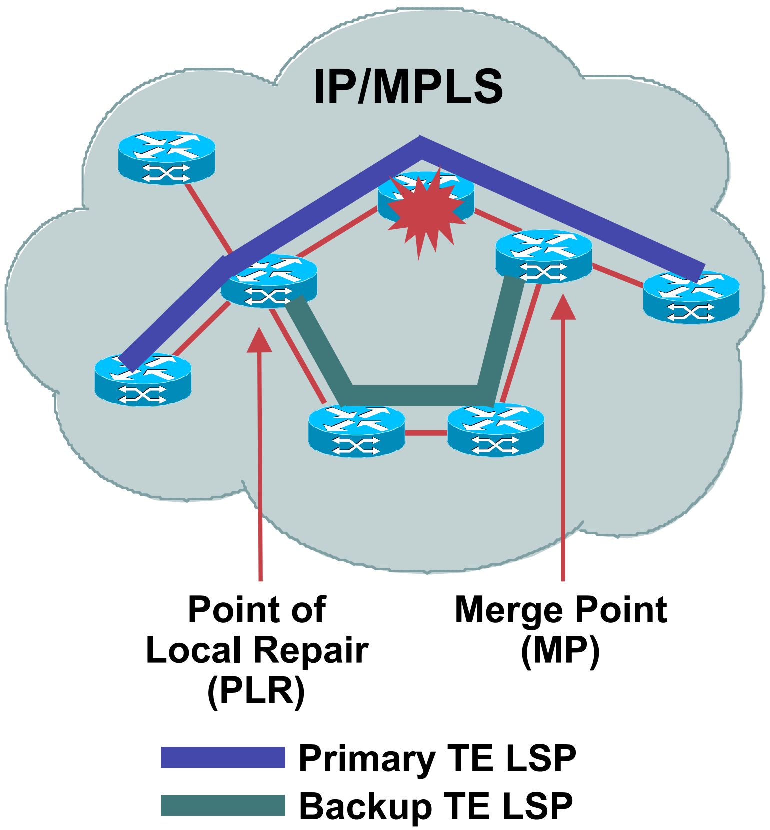
- During a failure:
  - Are you missing your SLA?
  - For how long?
- Link failure may have 2x impact on load
- Node/SRLG failure may have a 4x impact on load
- Failure impact and duration dependent on:
  - Network topology
  - Backbone QoS design

MPLS TE Fast Re-Route (FRR)



- **Subsecond recovery against node/link failures**
- **Scalable 1:N protection**
- **Bandwidth protection**
- **Greater protection granularity**
- **Cost-effective alternative to optical protection**

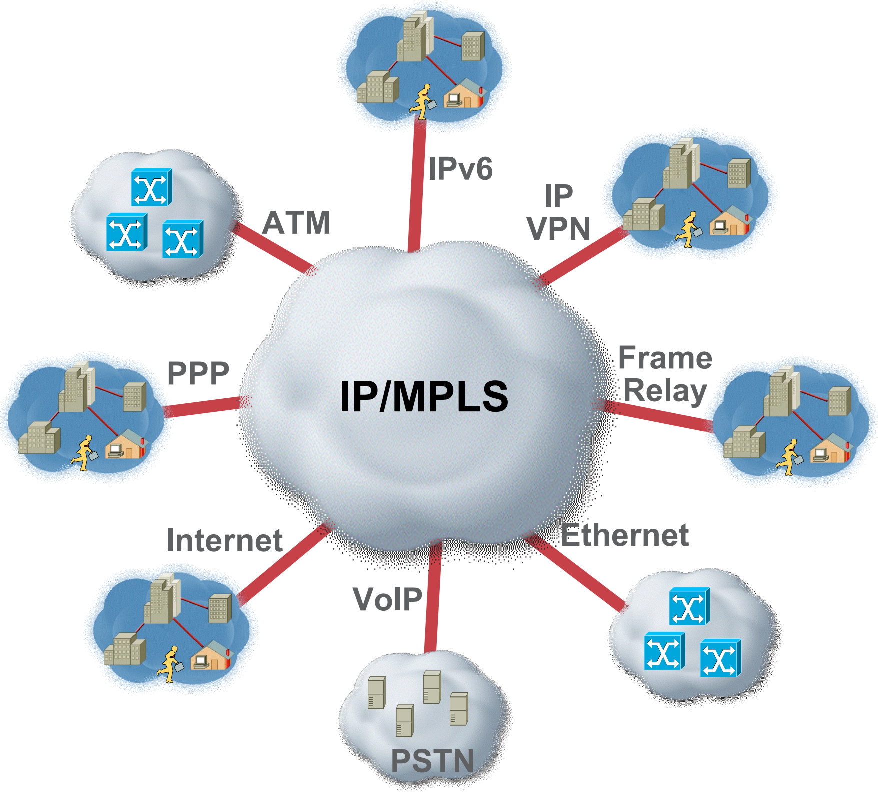
How MPLS TE FRR Works



- Next-Hop backup tunnel for link protection
- Next-Next-Hop backup tunnel for node protection
- Point of Local Repair (PLR) swaps label and pushes backup label
- Local repair in msecs
- Failure detection critical for total repair time
- PLR sends PathErr to head end triggering global re-optimization

MPLS QOS
BACKBONE INFRASTRUCTURE

Backbone Requirements

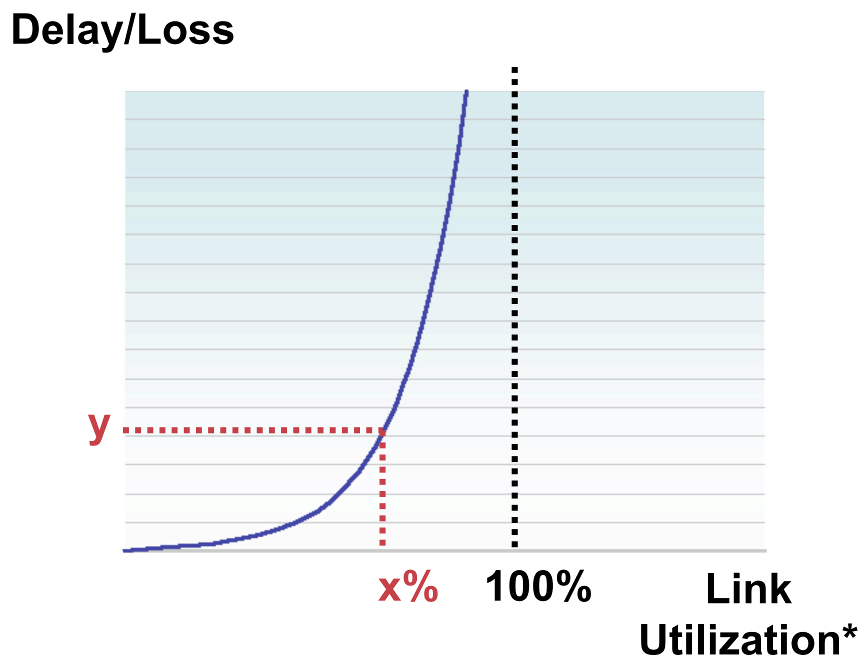


- **Growing trend: MPLS as selected choice for next generation multiservice network**
- **MPLS QoS architecture must fit multiservice strategy**
- **Architecture must be flexible and scalable**

Selecting Utilization Level (x%)

Target Utilization Level (x%) Is a Function of:

- Target QoS guarantees (delay, jitter, loss)
- Failure handling policies (link, node, SLRG)
- Schools of thought for "queuing theory"
- Heuristics
- Risk tolerance
- Testing
- Politics
- Technology religion, etc.



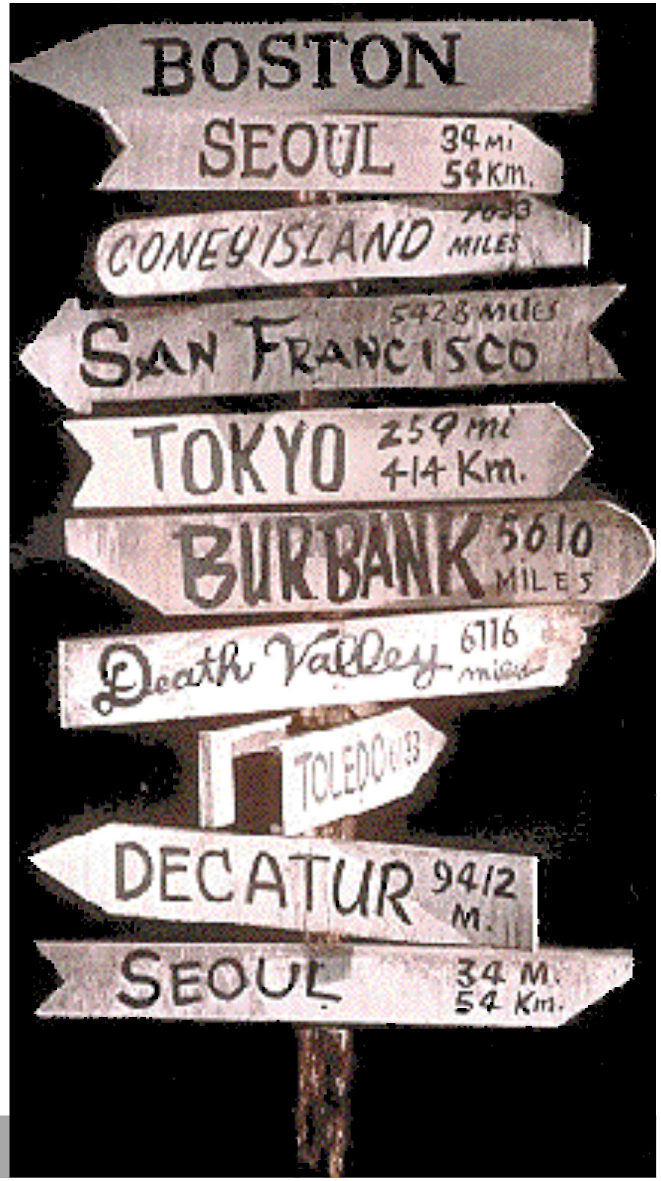
*Measured on a Large Timescale

Enforcing Utilization Level (x%)

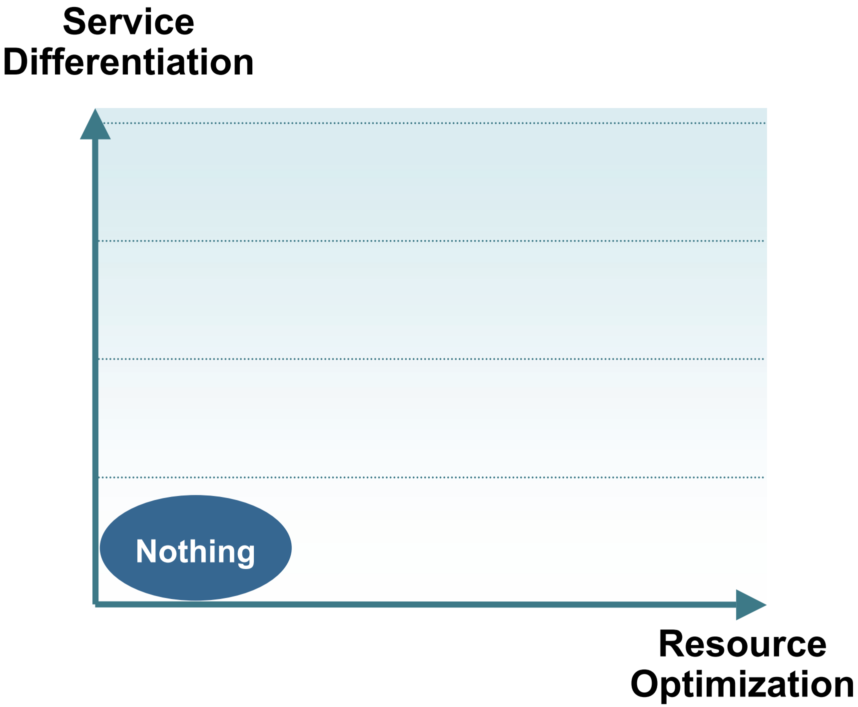
- **Aggregate capacity planning**
  - **Adjust link capacity to expected link load**
- **MPLS DiffServ**
  - **Adjust class capacity to expected class load**
- **MPLS traffic engineering**
  - **Adjust link load to actual link capacity**
- **MPLS DiffServ-Aware TE (DS-TE)**
  - **Adjust class load to actual class capacity**

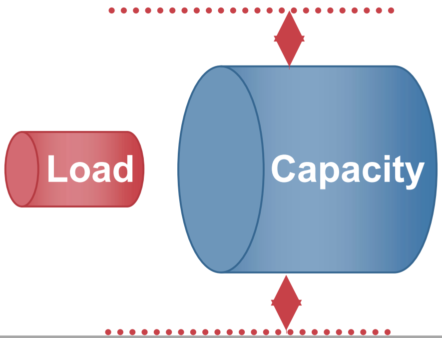
What Should I Use in My Backbone?

- Nothing
- MPLS TE
- MPLS DiffServ
- MPLS DiffServ + MPLS TE
- MPLS DiffServ + MPLS DS-TE
- Any of the above + MPLS TE FRR



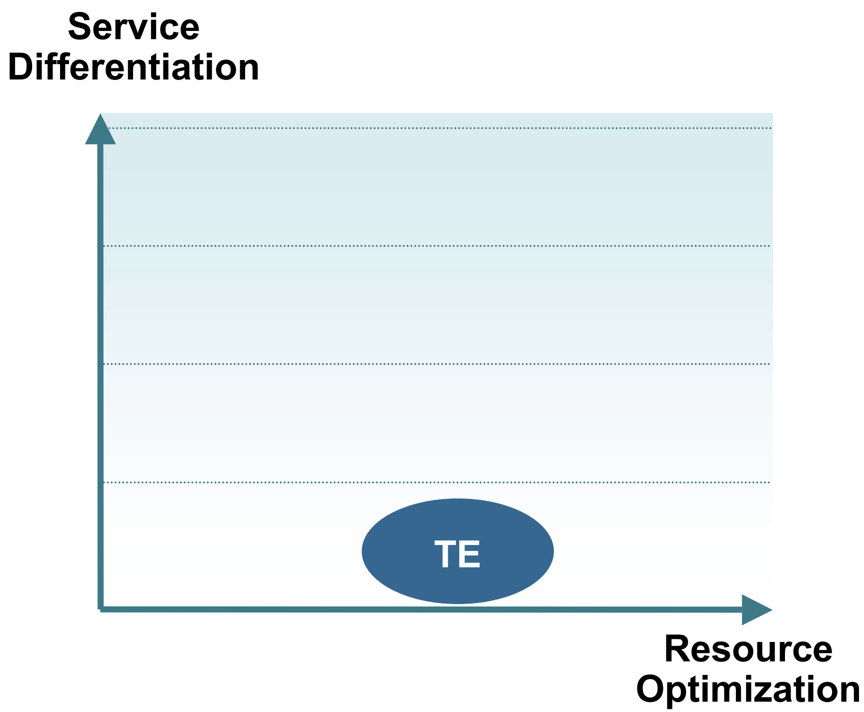
Backbone with Nothing: No MPLS DiffServ and No MPLS TE



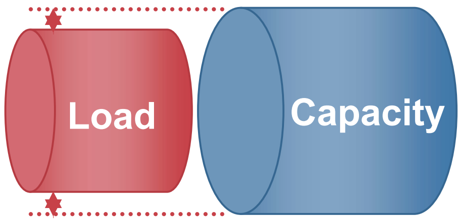


- A solution when:
  - No differentiation required
  - No optimization required
- Capacity planning as QoS tool
- Link over-provisioning to meet all SLAs
- Adjust link capacity to expected link load

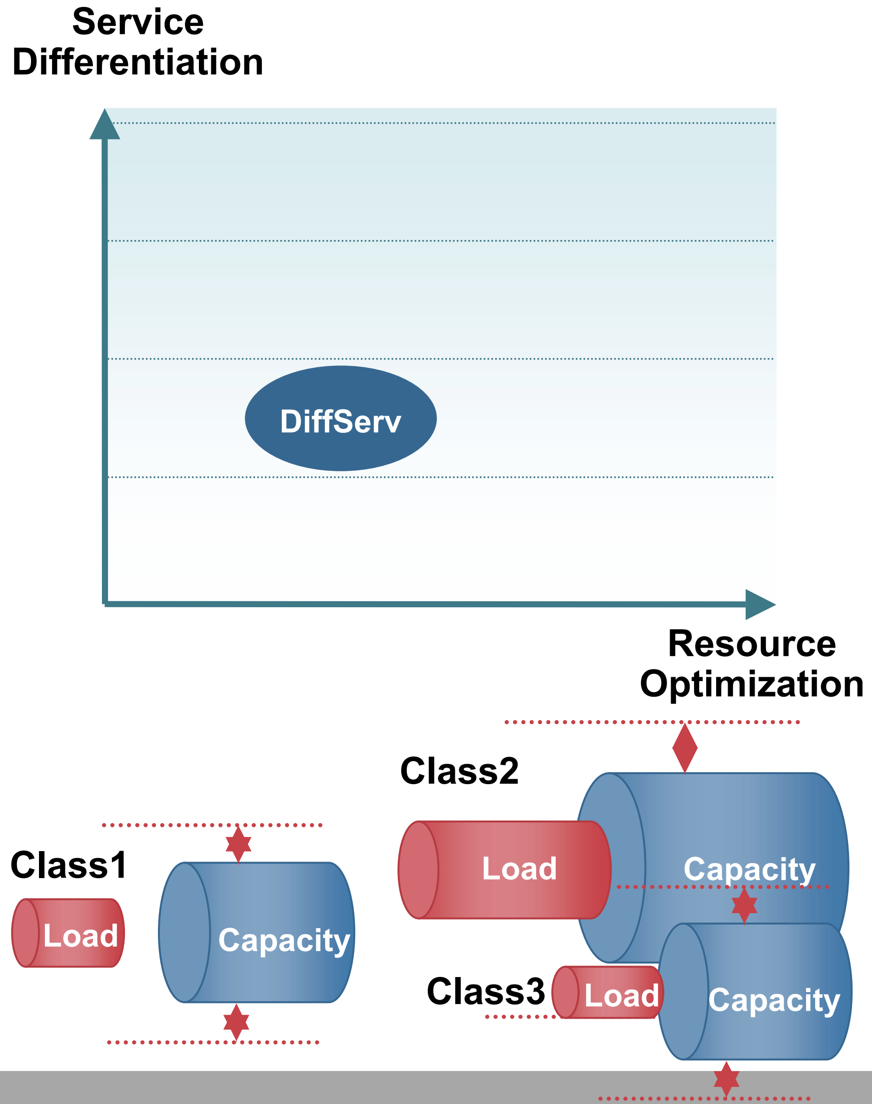
Backbone with MPLS TE



- A solution when:
  - No differentiation required
  - Optimization required
- Full mesh or selective deployment to avoid over-subscription
- Increased network utilization
- Adjust link load to actual link capacity

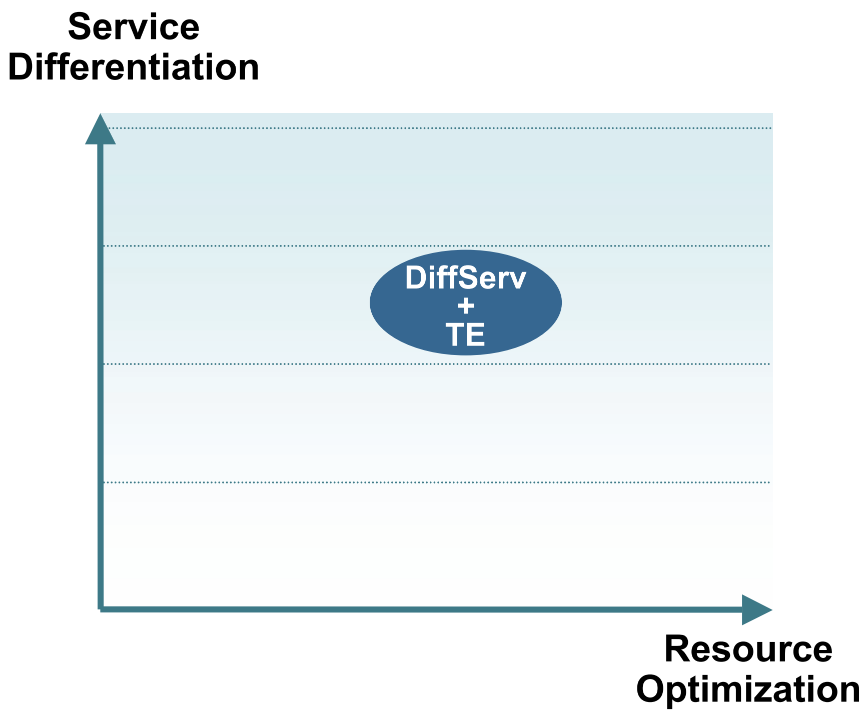


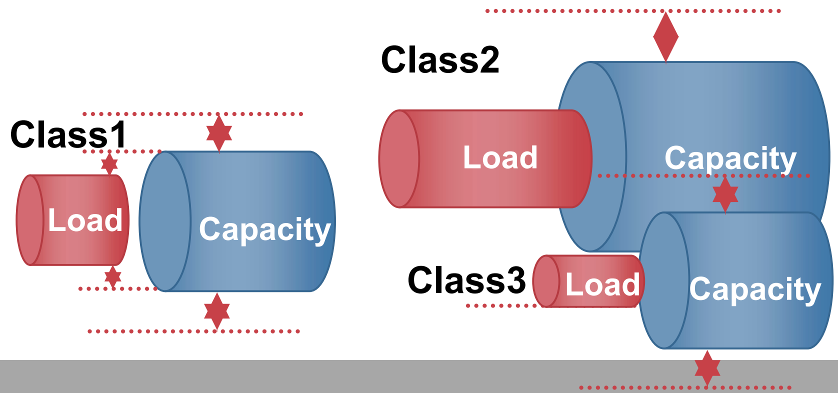
Backbone with MPLS DiffServ



- A solution when:
  - Differentiation required
  - Optimization required
- Per-class capacity planning
- Same or lower number of classes than edge
- Adjust class capacity to expected class load

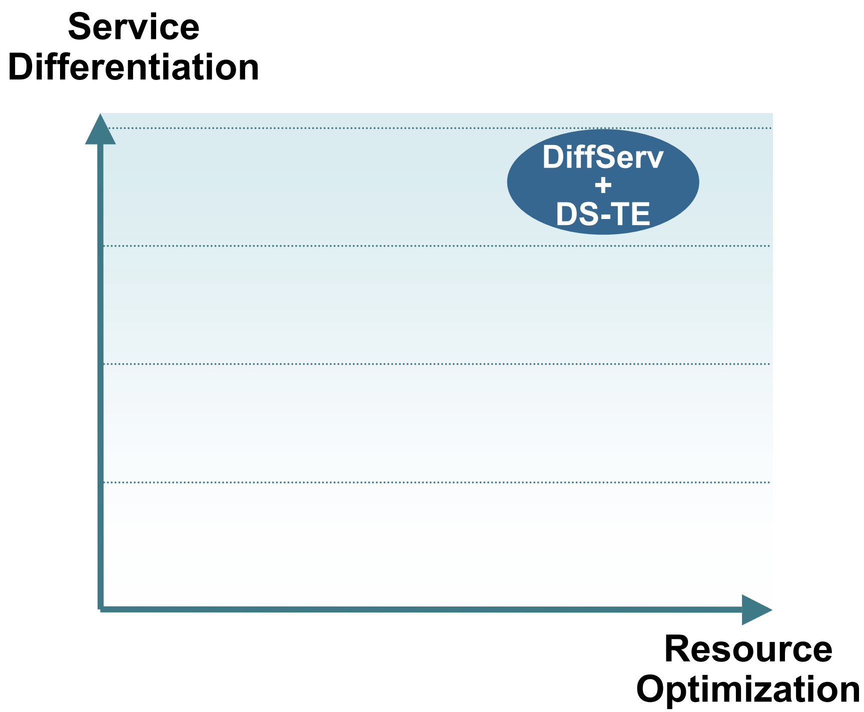
Backbone with MPLS DiffServ and MPLS TE

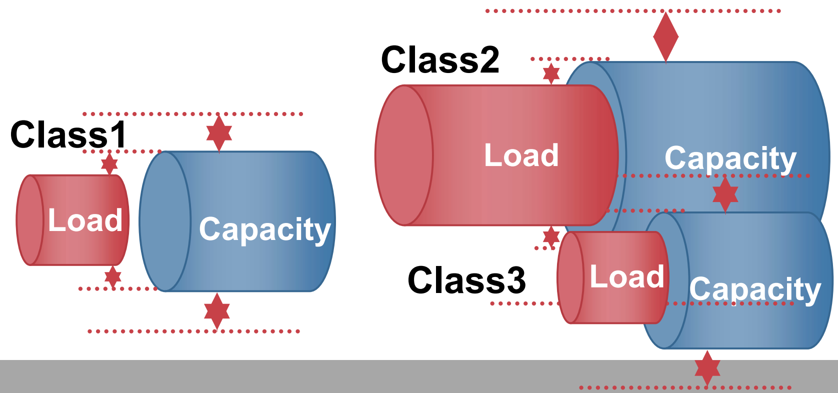




- A solution when:
  - Differentiation required
  - Optimization required
- Adjust class capacity to expected class load
- Adjust class load to actual class capacity for one class
- Alternatively, adjust link load to actual link capacity

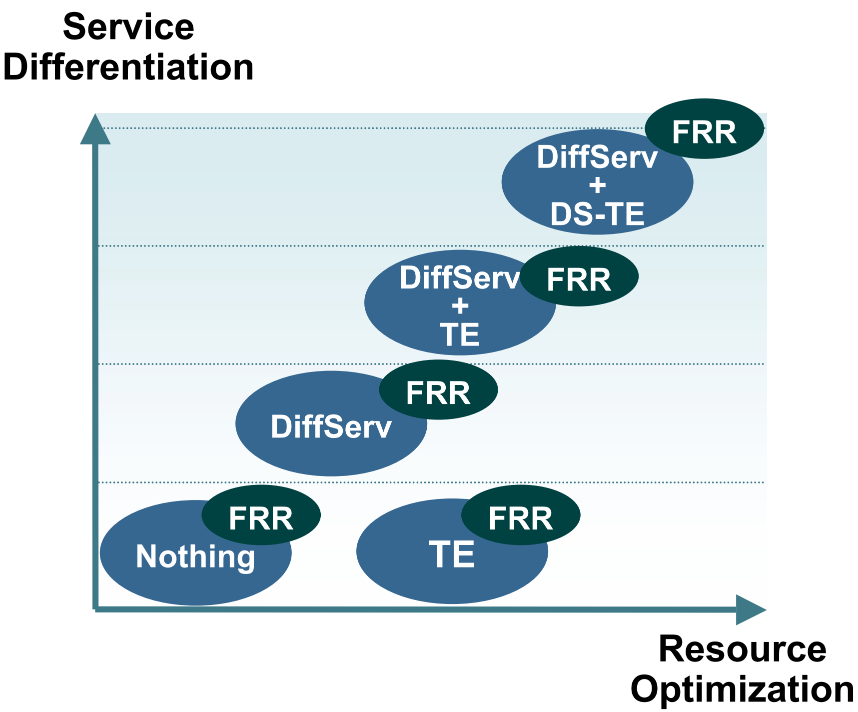
Backbone with MPLS DiffServ and MPLS DS-TE





- A solution when:
  - Strong differentiation required
  - Fine optimization required
- Adjust class capacity to expected class load
- Adjust class load to actual class capacity

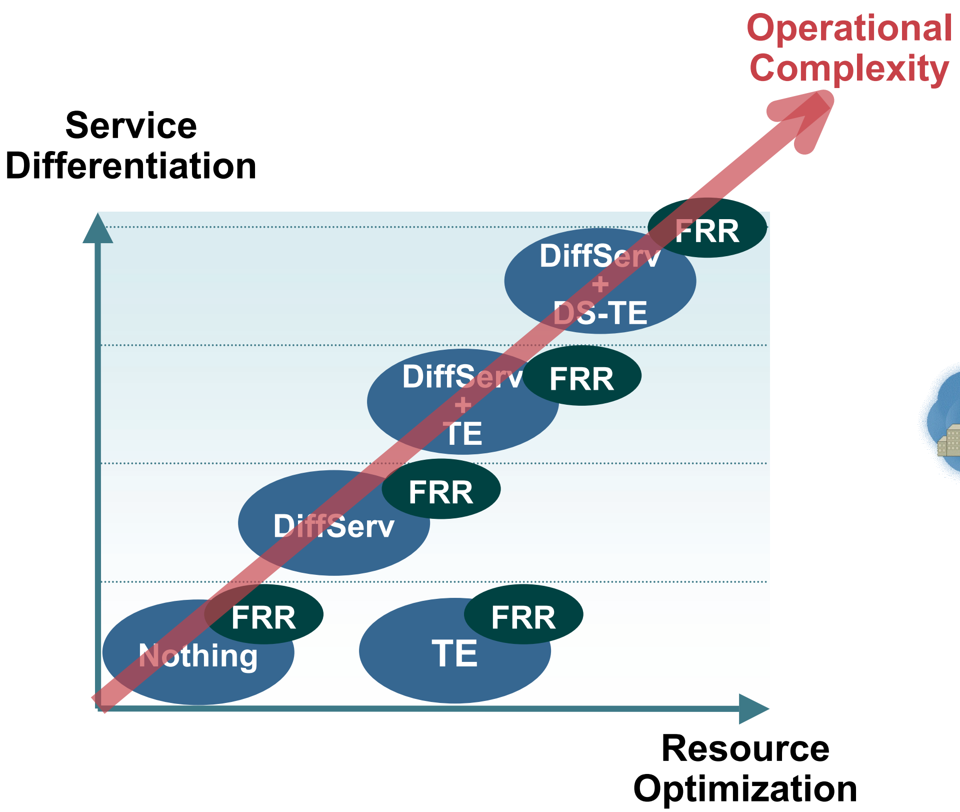
Bringing MPLS TE FRR into the Mix

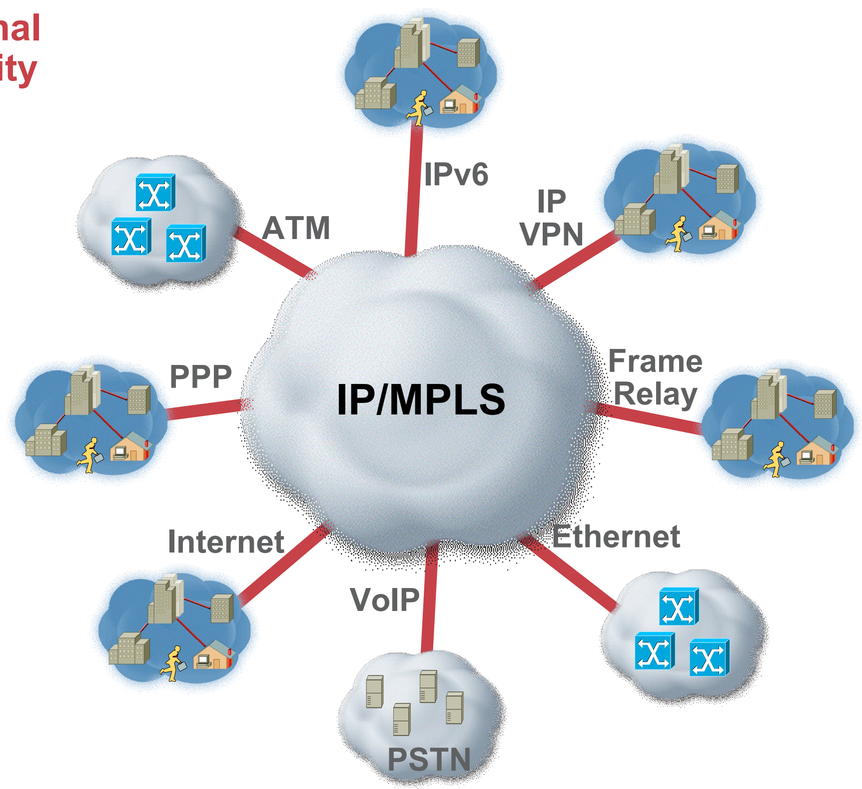


- **Increases resiliency regardless of backbone QoS design**
- **Stronger SLAs during single failure conditions (link, node, shared-risk link group)**
- **Optimization of backup resources**

What Model to Use?

Take Your Pick!
As Sophisticated as Necessary, but Not More

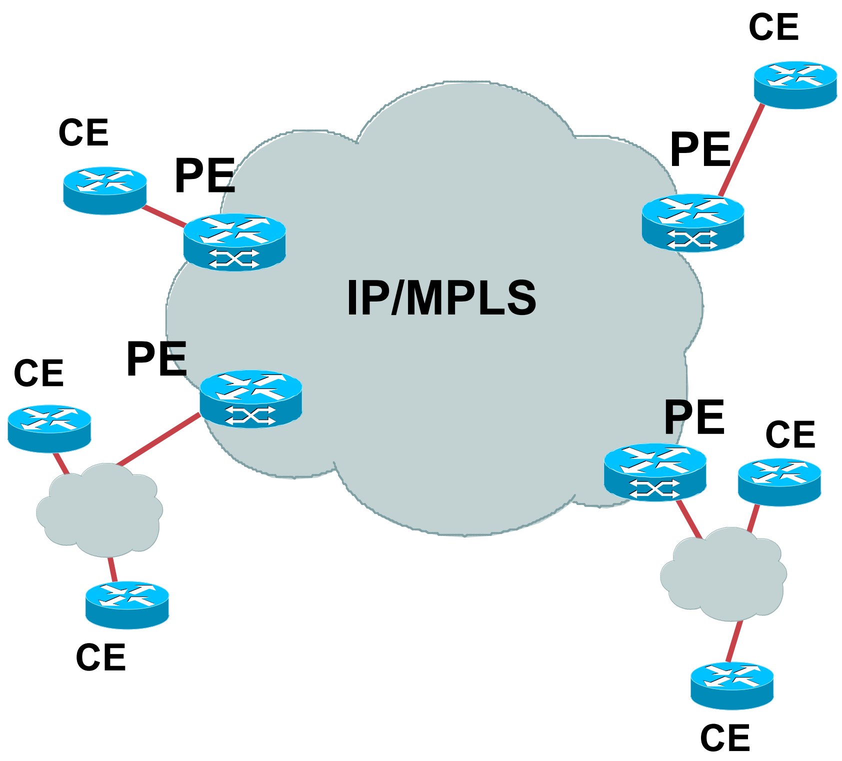




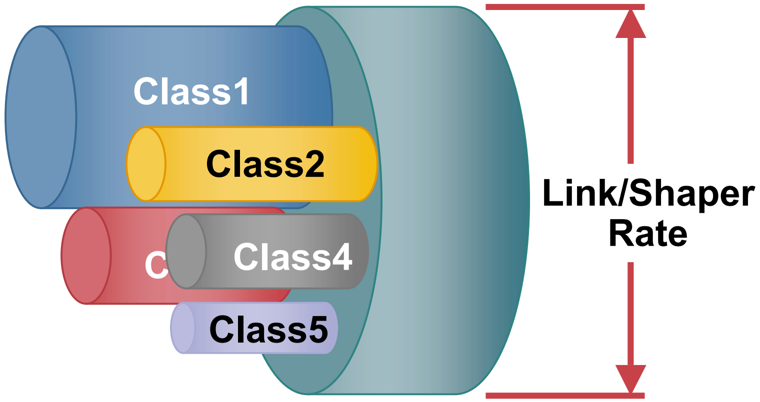
MPLS QOS
IP SERVICES

QoS for IP Services

- **Elaborate DiffServ Edge implementation**
  - Access link capacity controlled by customer (prone to congestion)
  - Trust boundary (SLA enforcement)
- **Applies to both IPv4 and IPv6**
- **Backbone must be able to support customer SLA**
- **Per-customer QoS policies only at the edge**



Site IP SLA

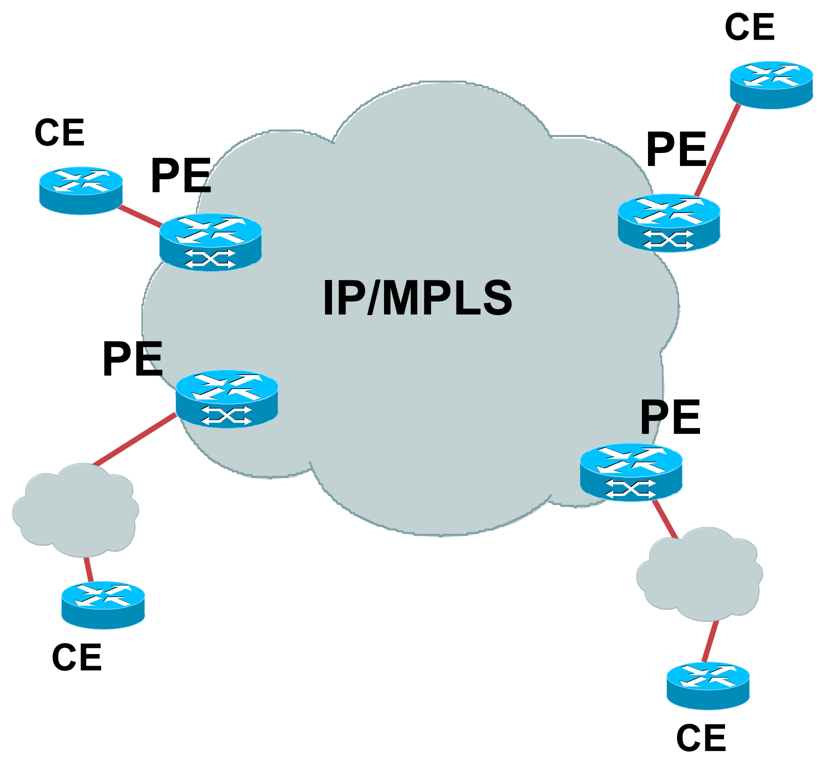


Class	Committed BW	Delay	Jitter	Loss
Real time	X	Low	Low	Low
Interactive	Y	Low	NA	Low
Business	Z	NA	NA	Low
Best Effort	NA	NA	NA	NA

- **Typically between 3 and 5 classes (real time, video, interactive, business, BE)**
- **Delay, jitter and loss guarantees for conforming real-time traffic**
- **Combination of delay and loss guarantees for data traffic**
- **Sum of committed bandwidth (per-class CIR) not to exceed link/shaper rate**
- **Additional classes not visible to customer may exist (e.g. management, control traffic)**

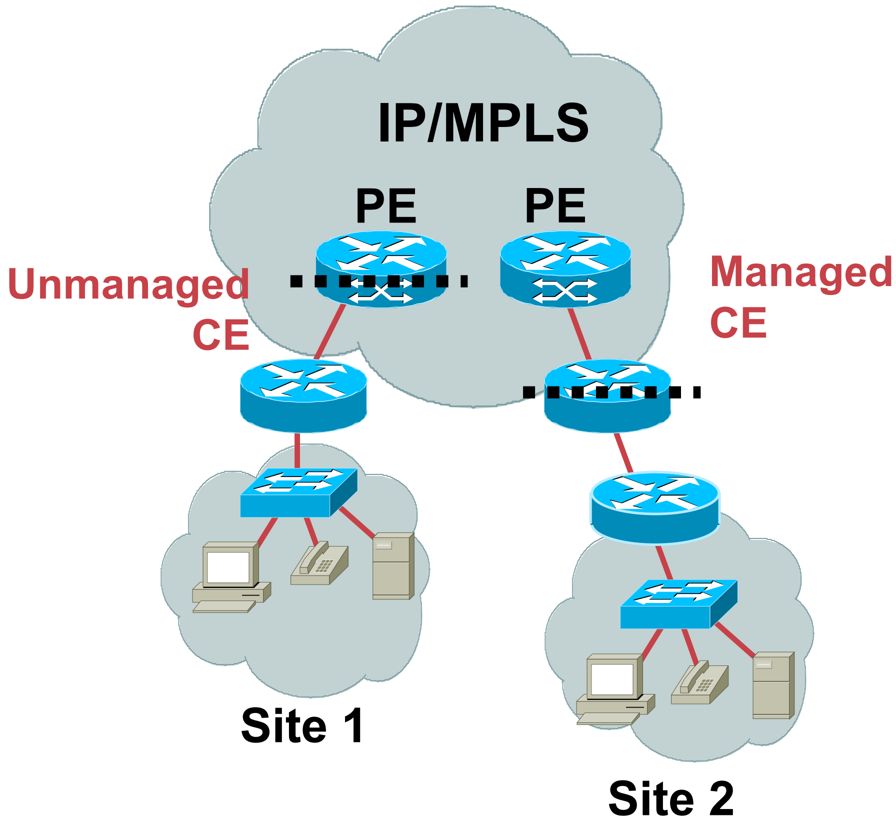
IP SLA Between Sites

- Site-to-network (point-to-cloud) guarantees for conforming traffic
- Each site may send x% of class n to network per SLA
- Each site may receive x% of class n from network per SLA
- No site-to-site (point-to-point) guarantees



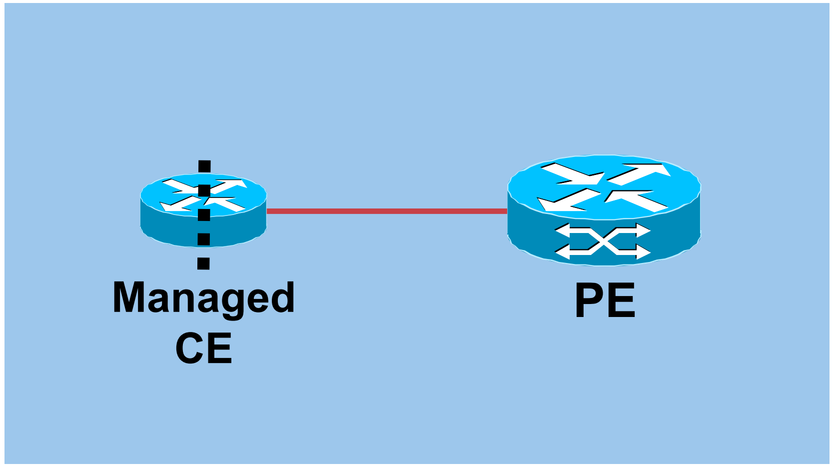
IP SLA Enforcement

- Managed vs. unmanaged IP service
- Trust boundary on PE for unmanaged service
- Trust boundary on CE for managed service
- Trust boundary defines SLA enforcement point
- Different QoS design options



Let's See How SLA enforcement Is Done IP QoS: Managed Service

- CE output and PE output policies enforce SLA
- Traffic classification and marking on CE
- No input QoS policies generally needed
- Explicit-null encapsulation may be used on CE to avoid remarking customer traffic
- Session RST-2502 provides enterprise (CE) details



CE Output Policy

Classification/ Marking

LLQ

WRED

[Shaping]

[LFI/cRTP]

PE Output Policy

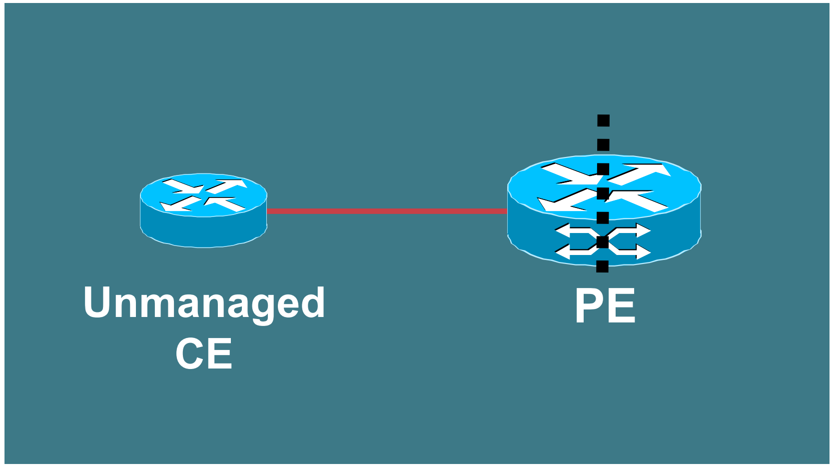
LLQ

WRED

[Shaping]

[LFI/cRTP]

IP QoS: Unmanaged Service



CE
Output Policy

<NOT SP controlled >

PE
Input Policy

Classification/ Marking

Policing

PE
Output Policy

LLQ

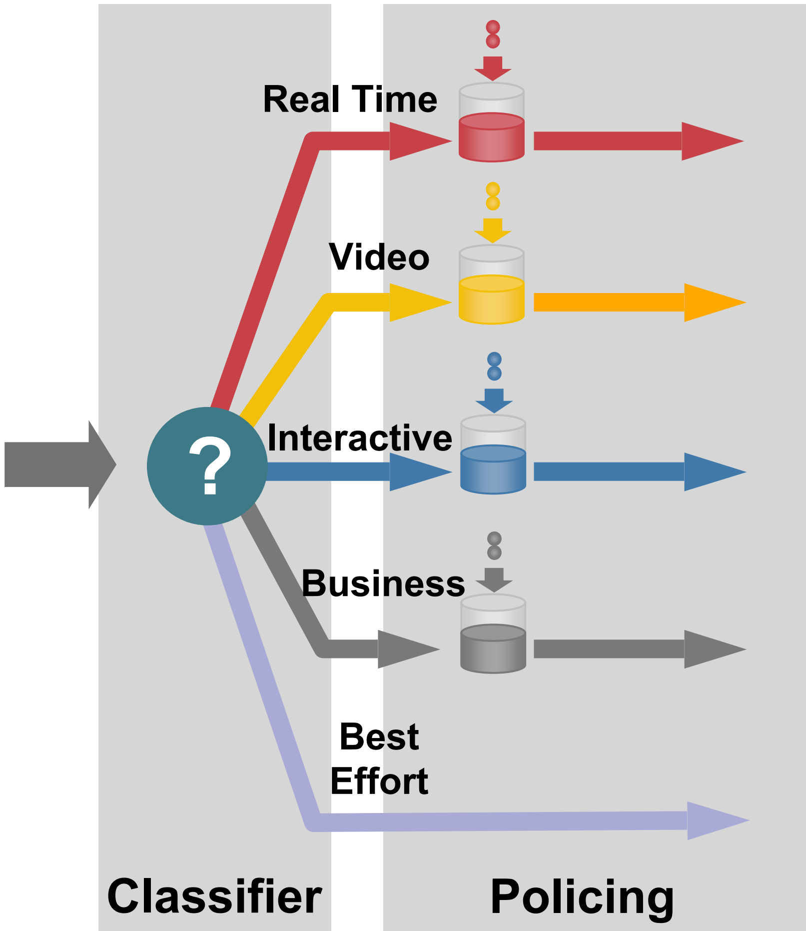
WRED

[Shaping]

[LFI/cRTP]

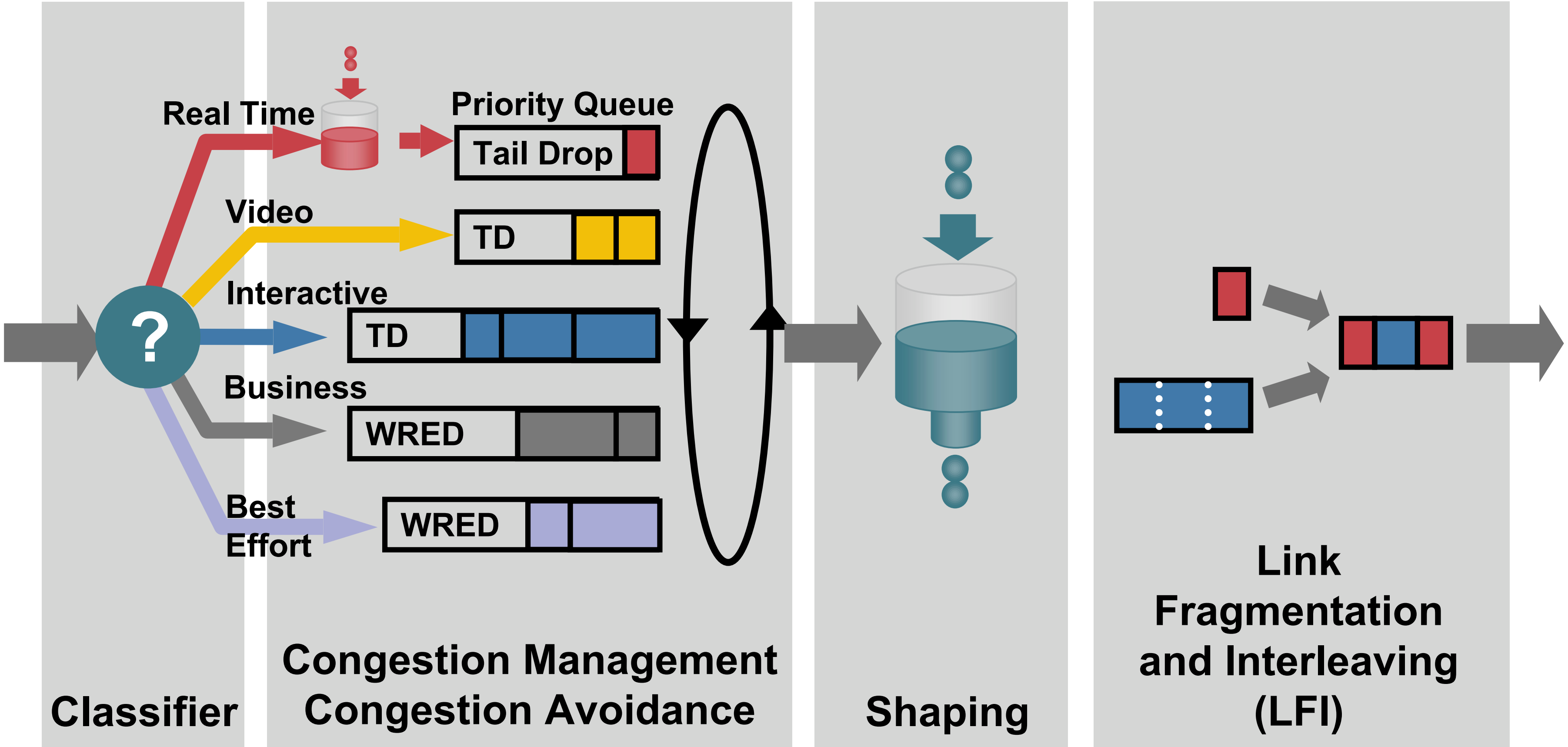
- **PE input** and **PE output** policies enforce SLA
- Traffic classification and markings on PE
- CE policies require coordination with PE policies (e.g. LFI, cRTP, end-to-end latency)
- Session RST-2502 provides enterprise (CE) details

Sample PE Input Policy: Unmanaged Service



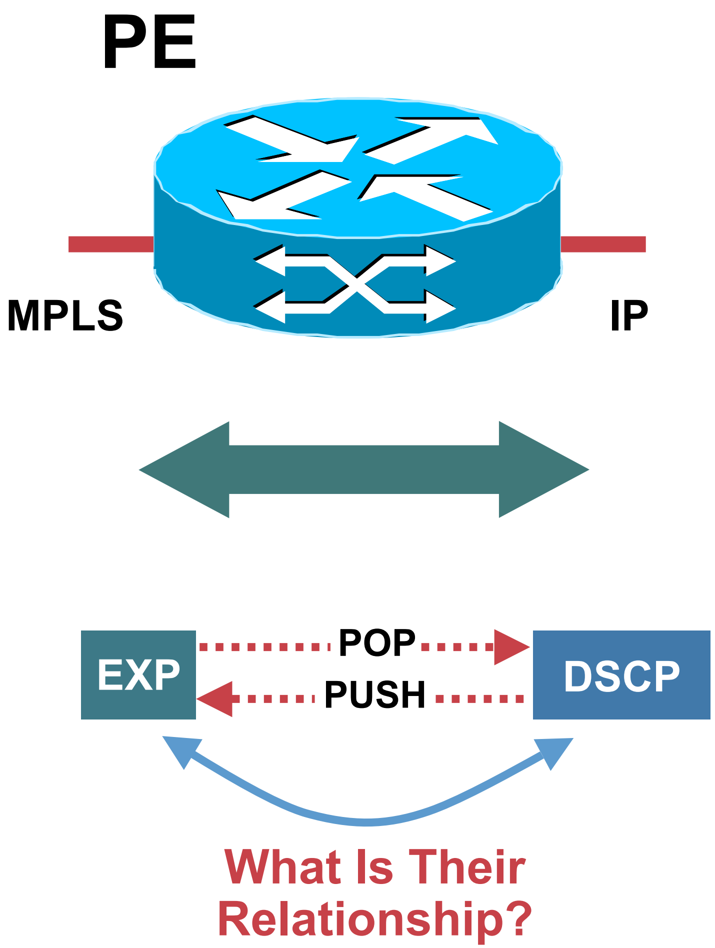
- Excess real time (voice) usually dropped
- Excess data marked down
- Dropping excess data at policer would affect many TCP sessions
- Best effort typically not policed
- Limited bandwidth sharing between classes with aggregate sub-rate
- Voice and video will benefit from admission control

Sample CE Output Policy: Managed Service



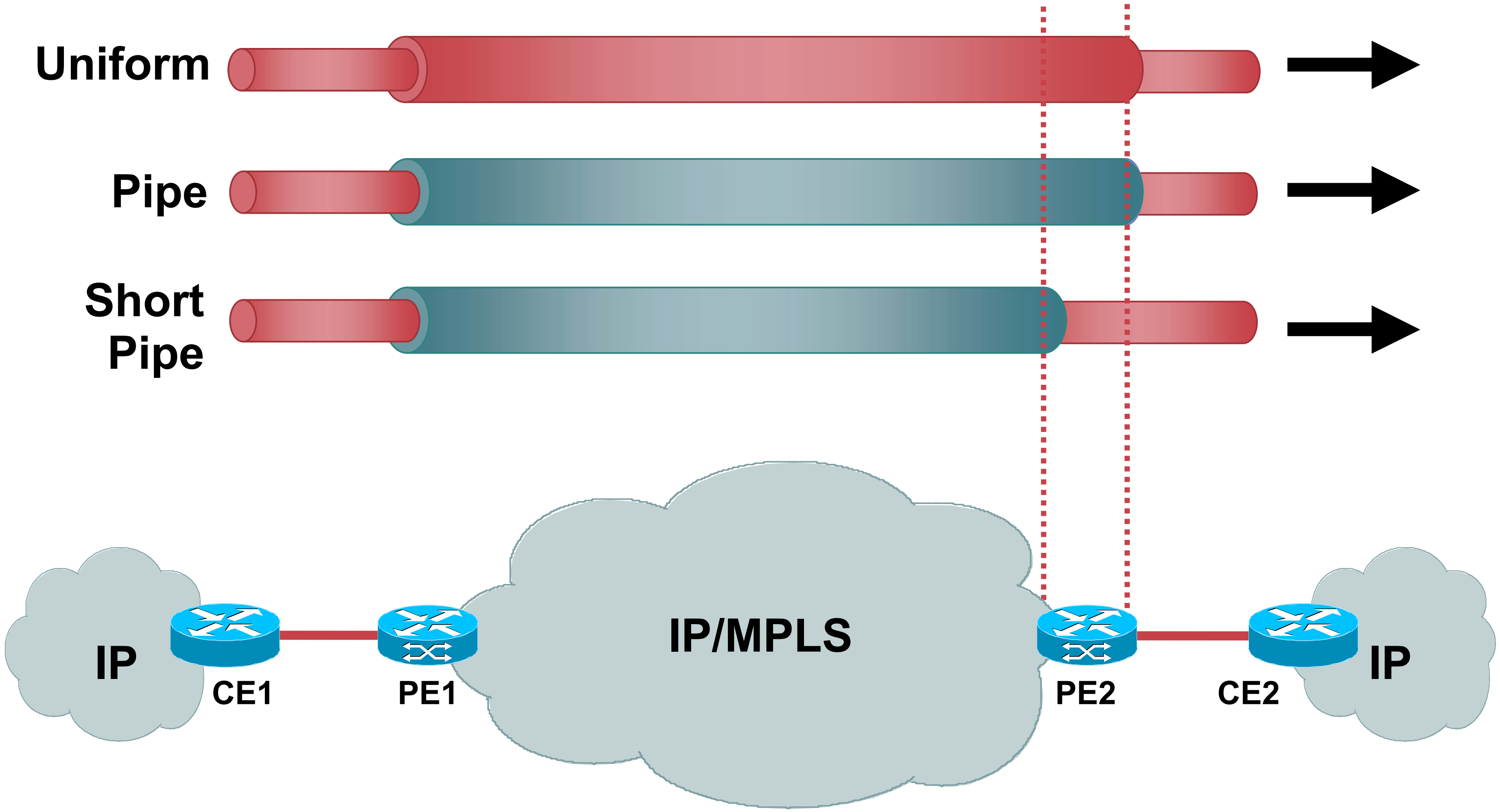
- LFI used in slow links to reduce delay and jitter for real-time traffic
- WRED used for TCP-friendly packet dropping

How DiffServ Markings Interact: DiffServ Tunneling Modes

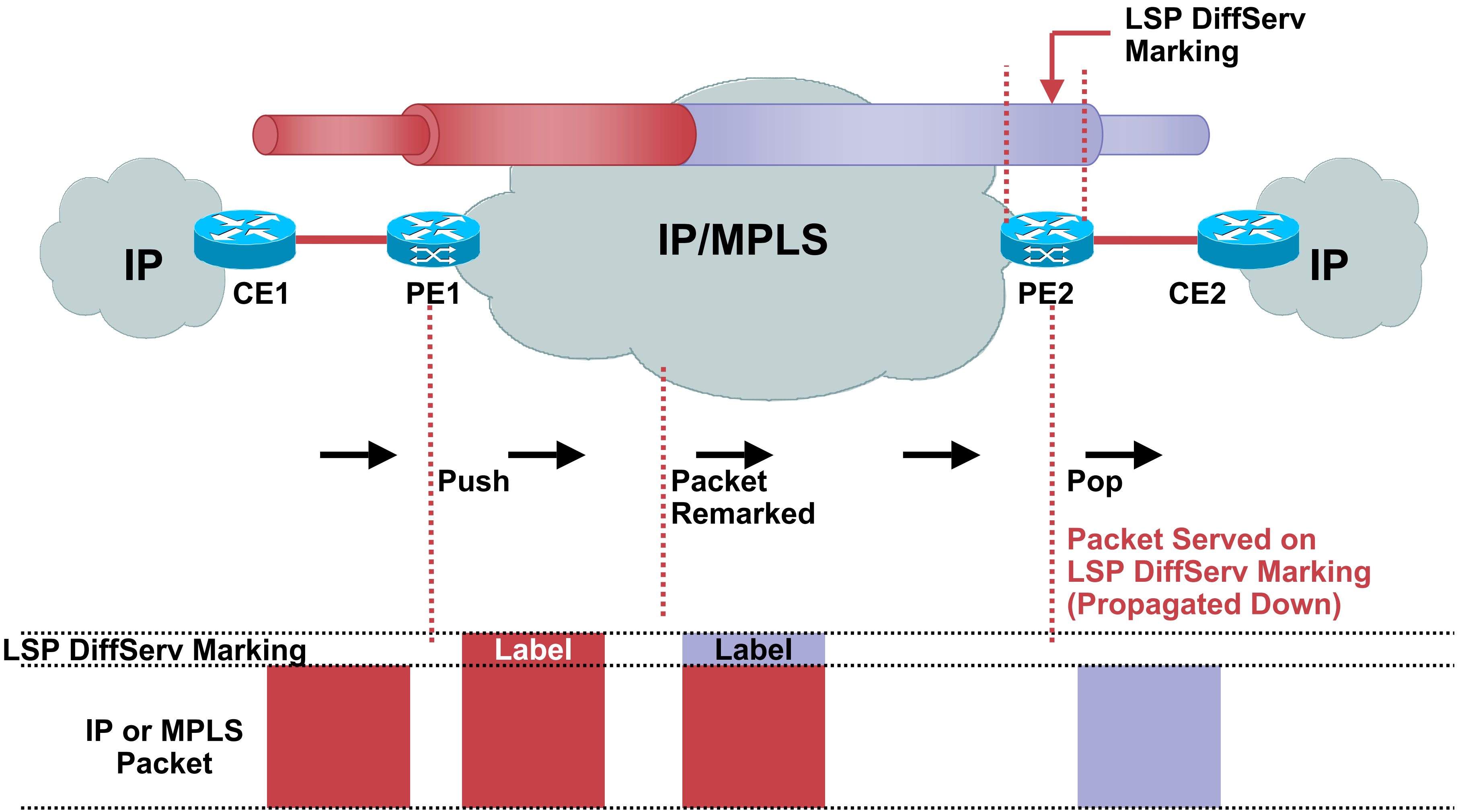


- **Several models (modes) of interaction between these markings**
- **RFC2983 defines models (uniform/pipe) for DiffServ with IP tunnels**
- **RFC3270 defines models (uniform/pipe/short-pipe) for MPLS**
- **Only relevant where pop or push operations take place (both on IP or MPLS packets)**
- **Explicit NULL label may be used for managed services**

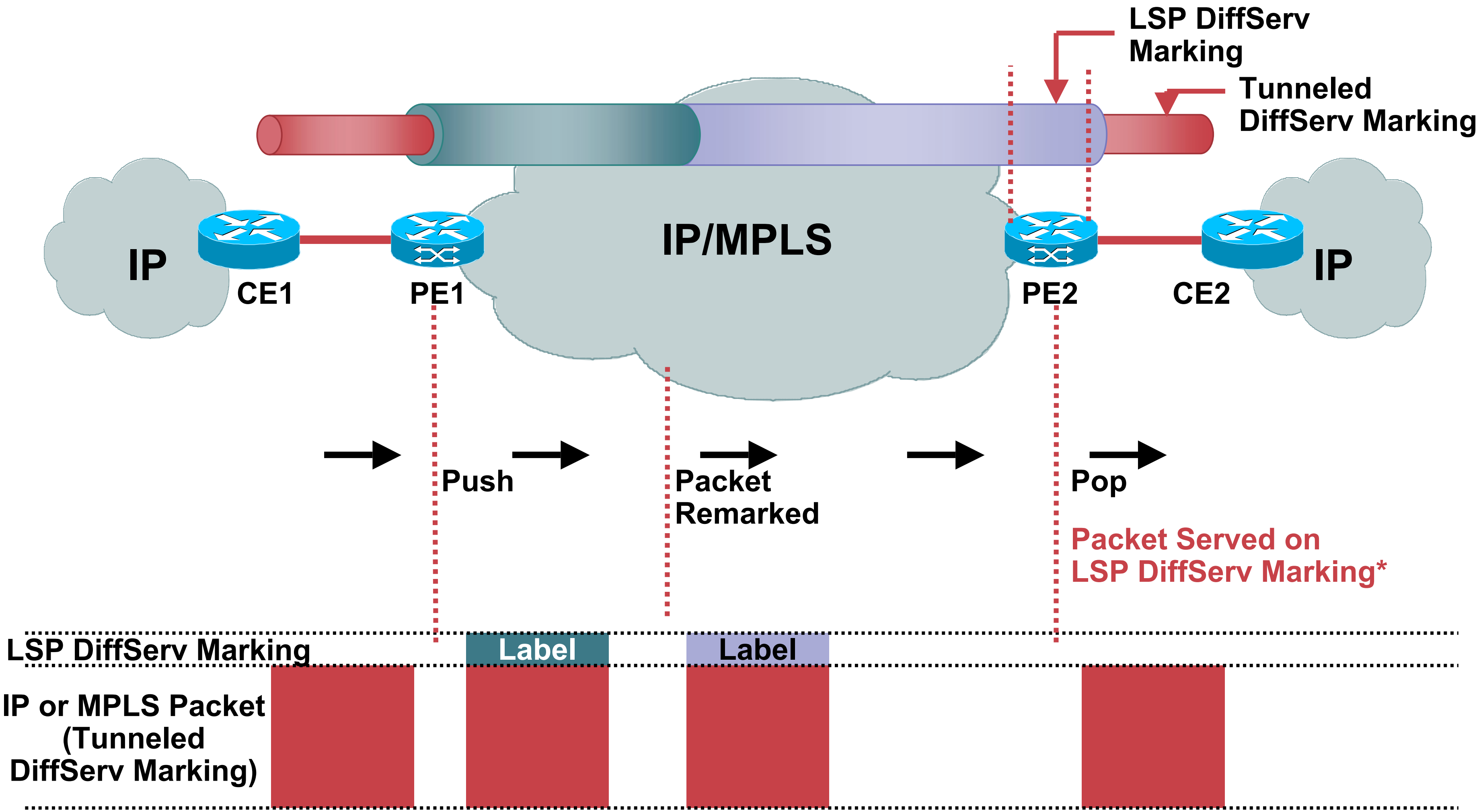
MPLS DiffServ Tunneling Modes



Uniform Mode

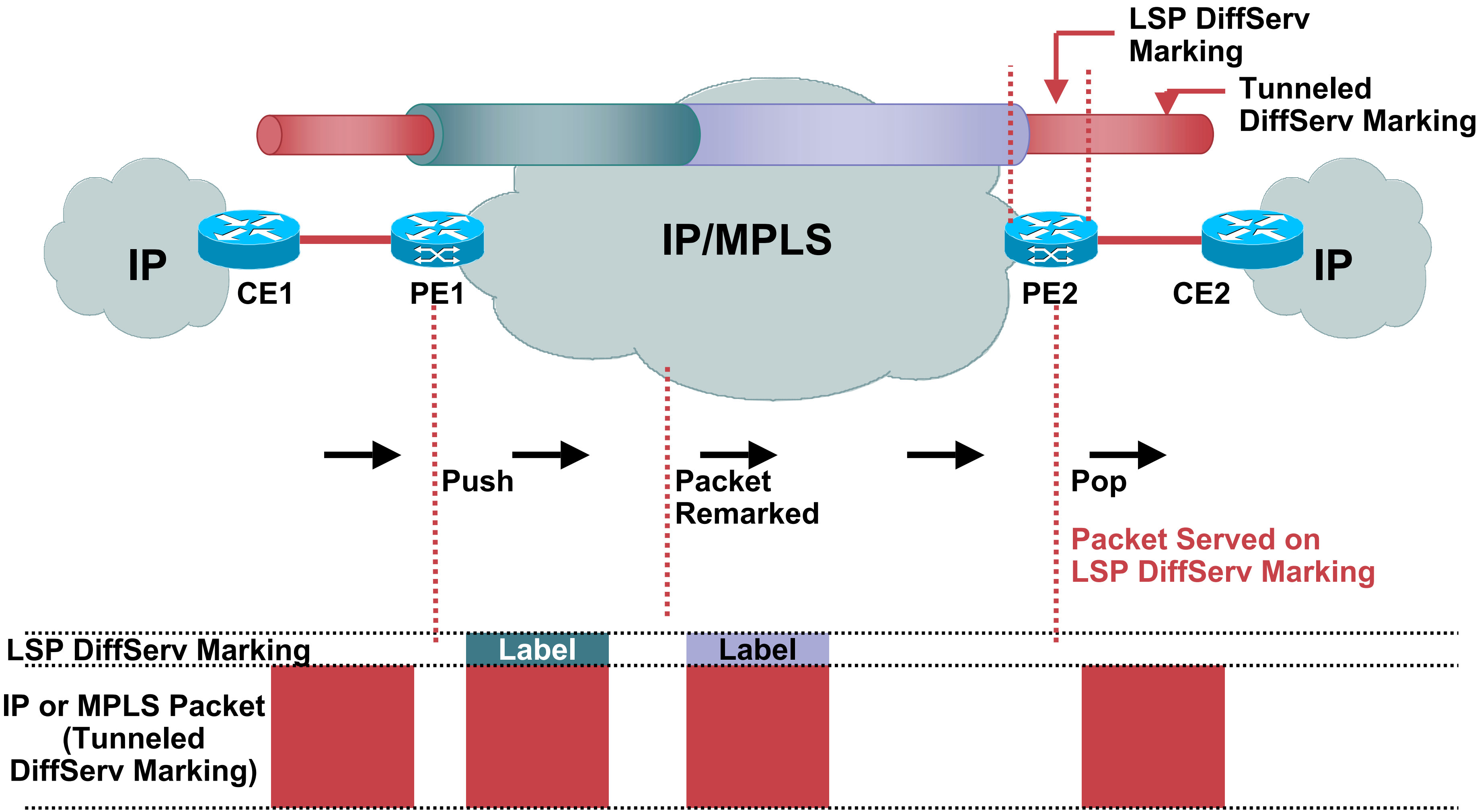


Pipe Mode

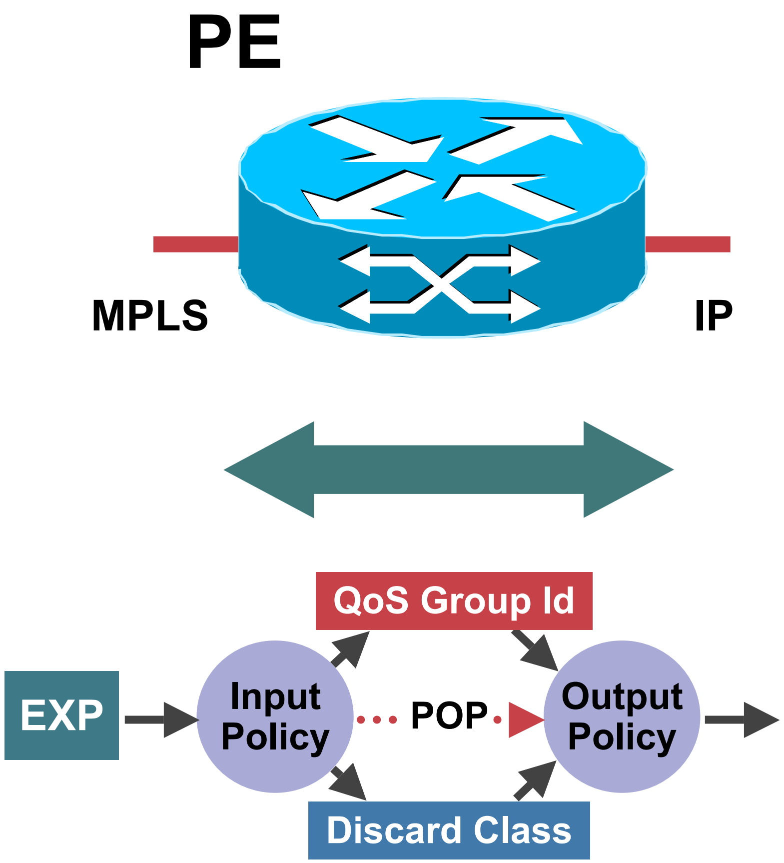


*Pipe Mode Precludes Penultimate Hop Popping (PHP)

Short Pipe Mode

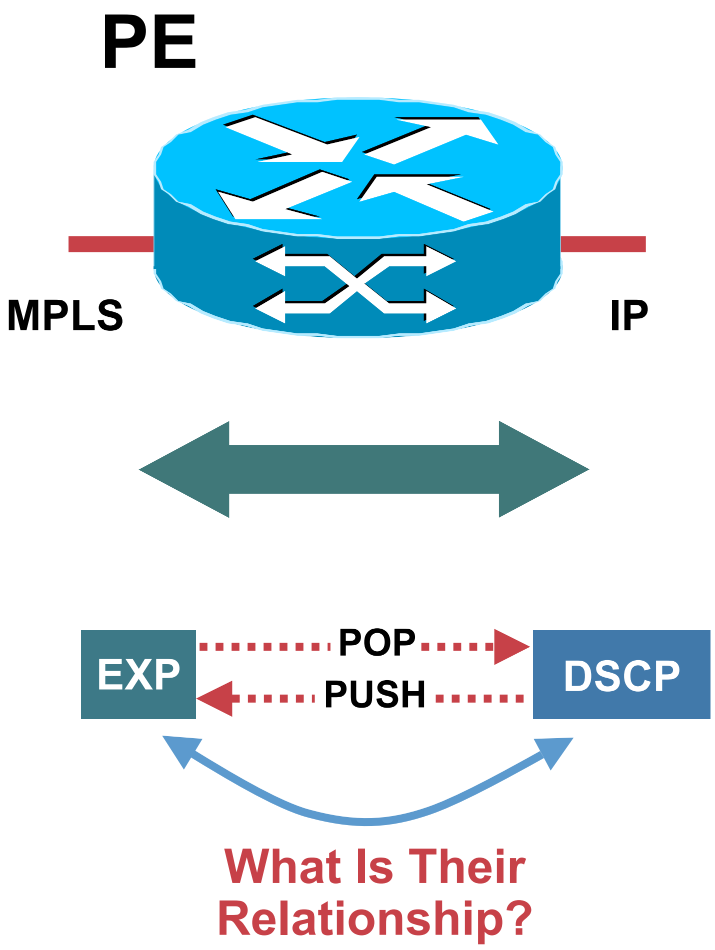


Local Packet Marking



- QoS Group Id and discard class for local packet marking
- Always an input feature (before label POP)
- Used to implement uniform and pipe mode
- Recommended semantics
  - QoS group identifies class
  - Discard class identifies drop precedence
- Discard class can drive WRED
- Not all classes will have a drop precedence (e.g. EF, best effort)

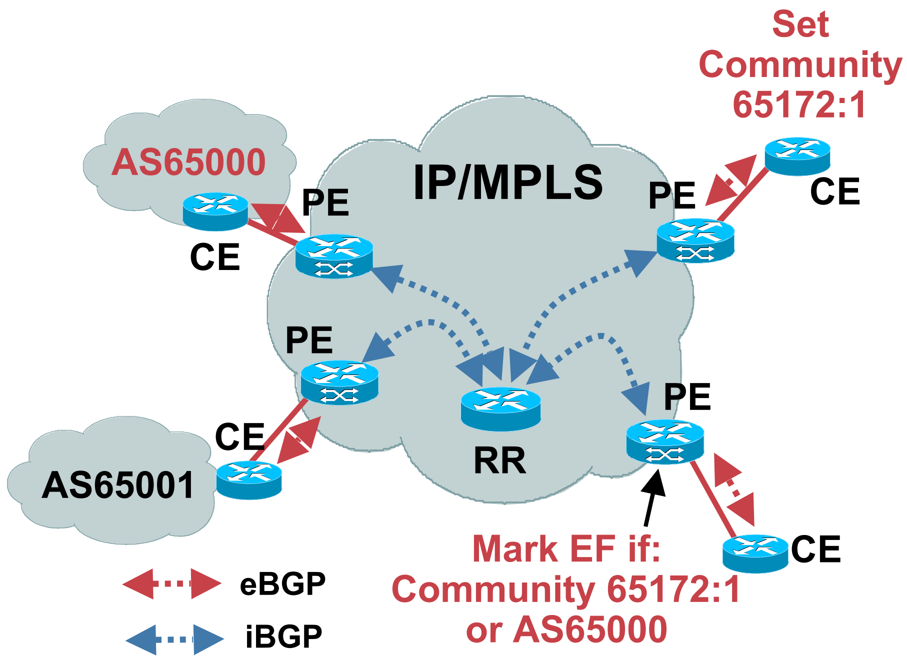
DiffServ Tunneling Modes: Keep in Mind...

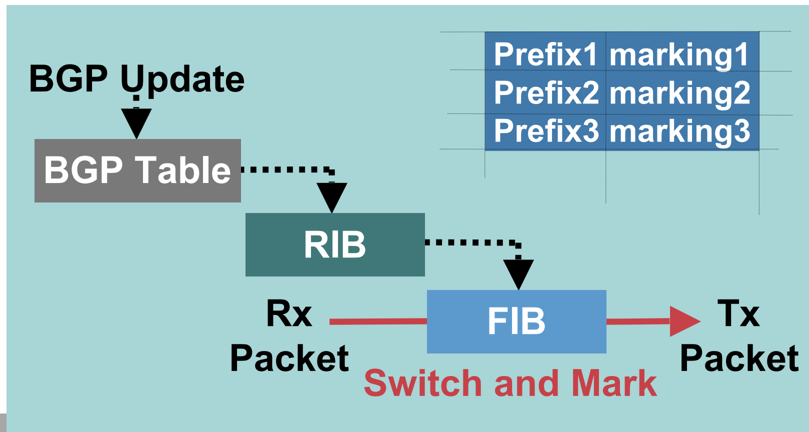


- When input policy defines EXP to be imposed, value applies to all imposed labels
- If no imposition EXP defined, IP precedence copied to all imposed labels
- EXP maintained during label swaps
- EXP not propagated down by default during disposition
- Pipe mode precludes PHP

Some Advanced Configurations: QoS Policy Propagation via BGP (QPPB)

- Despite the name, no policies are really propagated
- Input packet marking (IP precedence, QoS Group Id) based on
  - Community
  - AS path
  - IP prefix
- Packet marking happens before input QoS policy
- Supports IPv4 and VPNv4 addresses
- Could add intelligence to IP SLA between sites

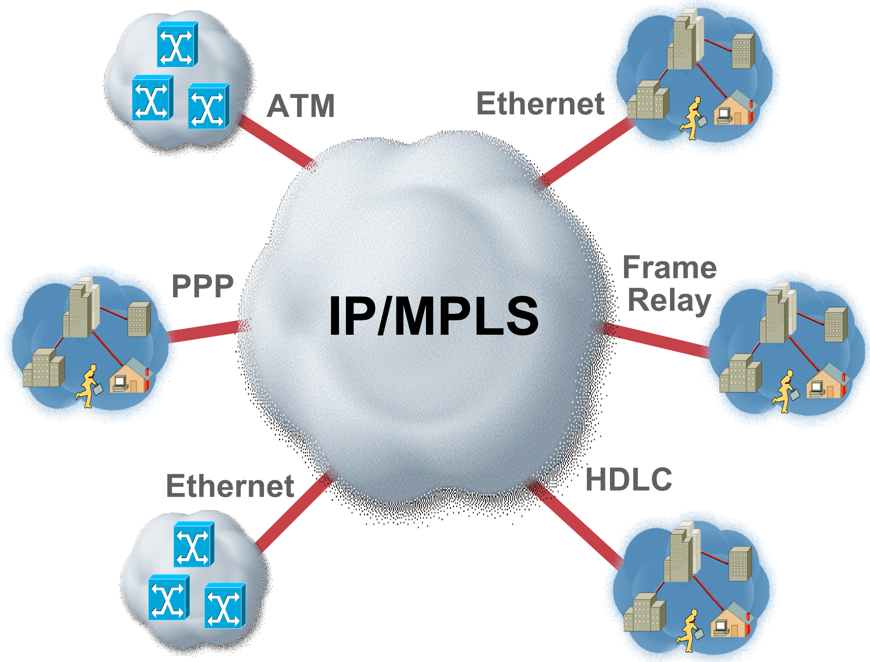




MPLS QOS
LAYER-2 SERVICES

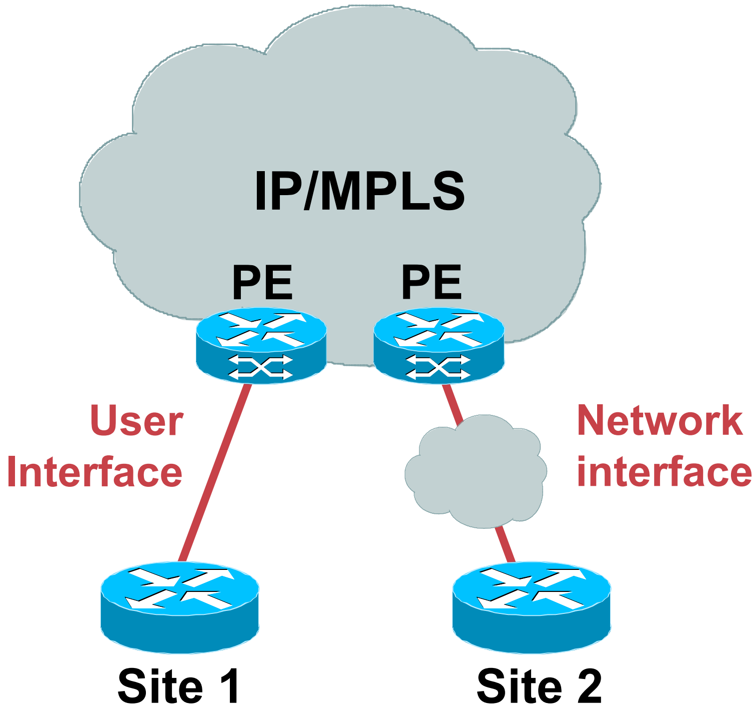
QoS for Layer-2 Services

- Well-defined SLAs for Frame Relay/ATM
- Differentiation for Ethernet services
- Point-to-Point SLA with exception of VPLS
- Backbone must be able to support customer SLA
- TE-enabled backbone attractive



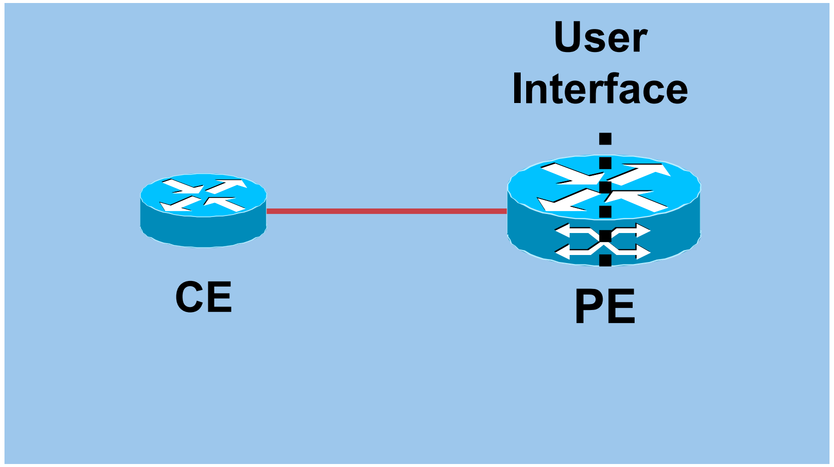
Layer-2 SLA Enforcement

- User interface vs network interface
- Trust boundary on PE for user interface
- Trust boundary on access network for network interface
- Trust boundary defines SLA enforcement point
- Different QoS design options



Let's See How SLA Enforcement Is Done
Layer-2 QoS: User Interface

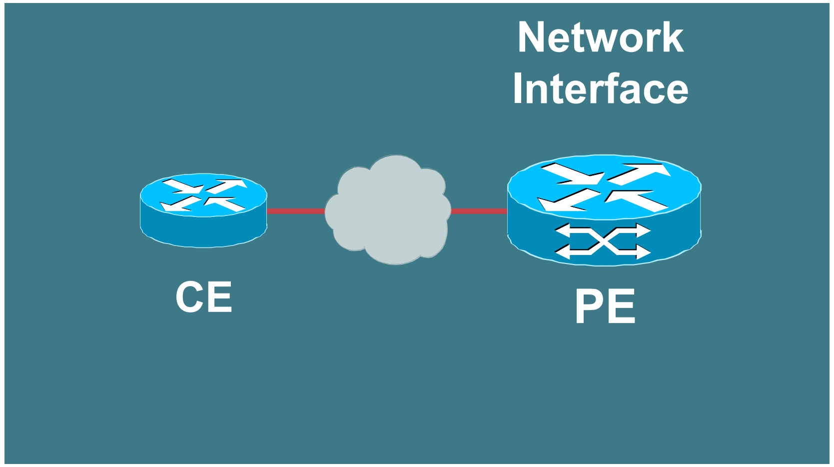
- **PE input** and **PE output** policies enforce SLA
- Drop precedence may be marked for FR/ATM/Ethernet
- Output drop precedence (e.g. ATM CLP, FR DE) marking when input marking not possible
- Ethernet may support multiple classes (802.1p bits)



PE
Input Policy
Policing
[Marking]

PE
Output Policy
Queuing (LLQ)
WRED
[Marking]
[shaping]

Layer-2 QoS: Network Interface

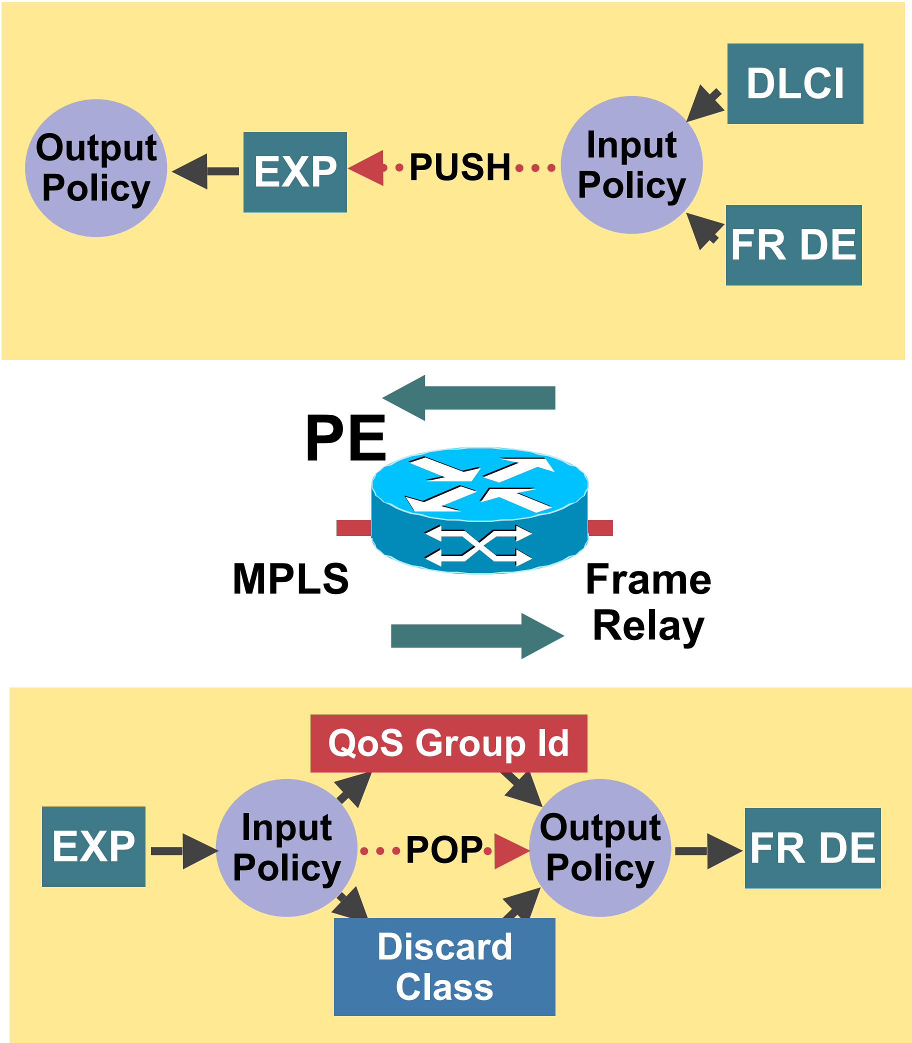


- SP enforces SLA on access network
- PE may only need simple aggregate policies

Access Network Input Policy	PE Input Policy
Policing [Marking]	[Marking]

Access Network Output Policy	PE Output Policy
Queuing (LLQ) Dropping (WRED) [Shaping]	<optional>

Encapsulation Details
Layer-2 QoS: Frame Relay



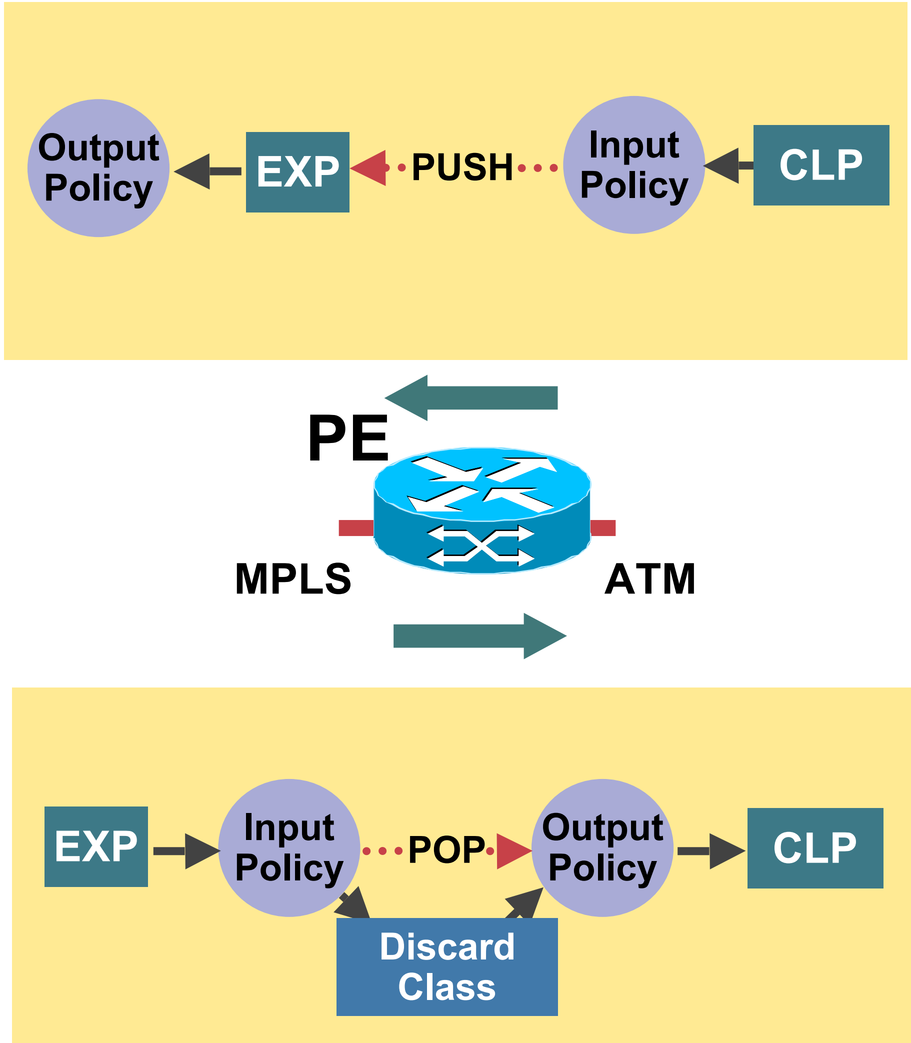
- Incoming traffic classified by DE or DLCI for DLCI-to-DLCI mode
- Input policer may exclude DE-marked frames from CIR metering
- Several classes of service may be implemented
  - CIR (EIR=0)
  - CIR+EIR
  - CIR=EIR=0
- Output DE marking when input marking not possible
- FECN/BECN marking supported on egress PE only
- Control word carries original DE/FECN/BECN values

Layer-2 QoS: ATM

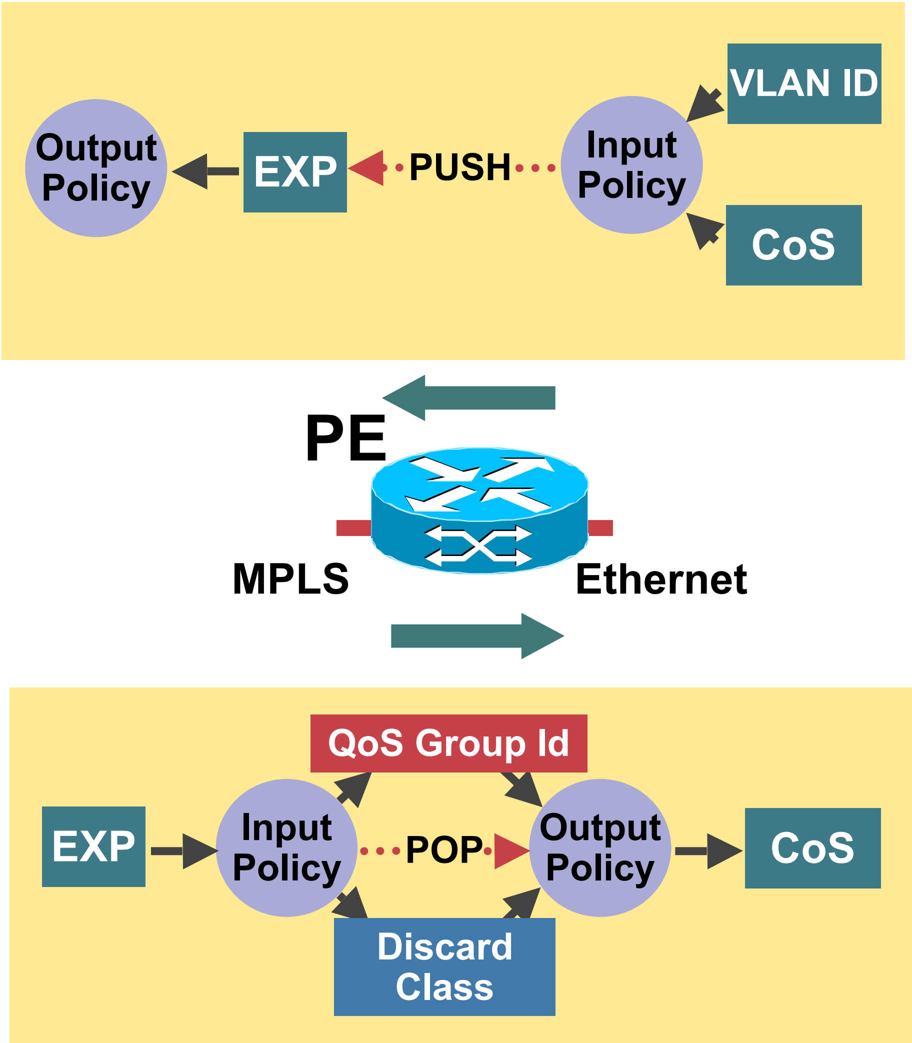
- Incoming traffic classified by CLP
- Support for all service categories (CBR, rt-VBR, nrt-VBR, ABR, UBR)
- Different traffic conformance supported (CBR.1, VBR.1, VBR.2, VBR.3, UBR.1, UBR.2)
- ATM TM 4.0 metering parameters converted to MQC (token-bucket) policer parameters

```
CIR = SCR*53*8
PIR = PCR*53*8
bc/be = CDVT*(CIR+53)*8
bc = MBS*PCR/SCR
```

- Output queuing handled by ATM hardware
- Cell-relay transport for delay sensitive traffic
- Control word carries original CLP and EFCI values

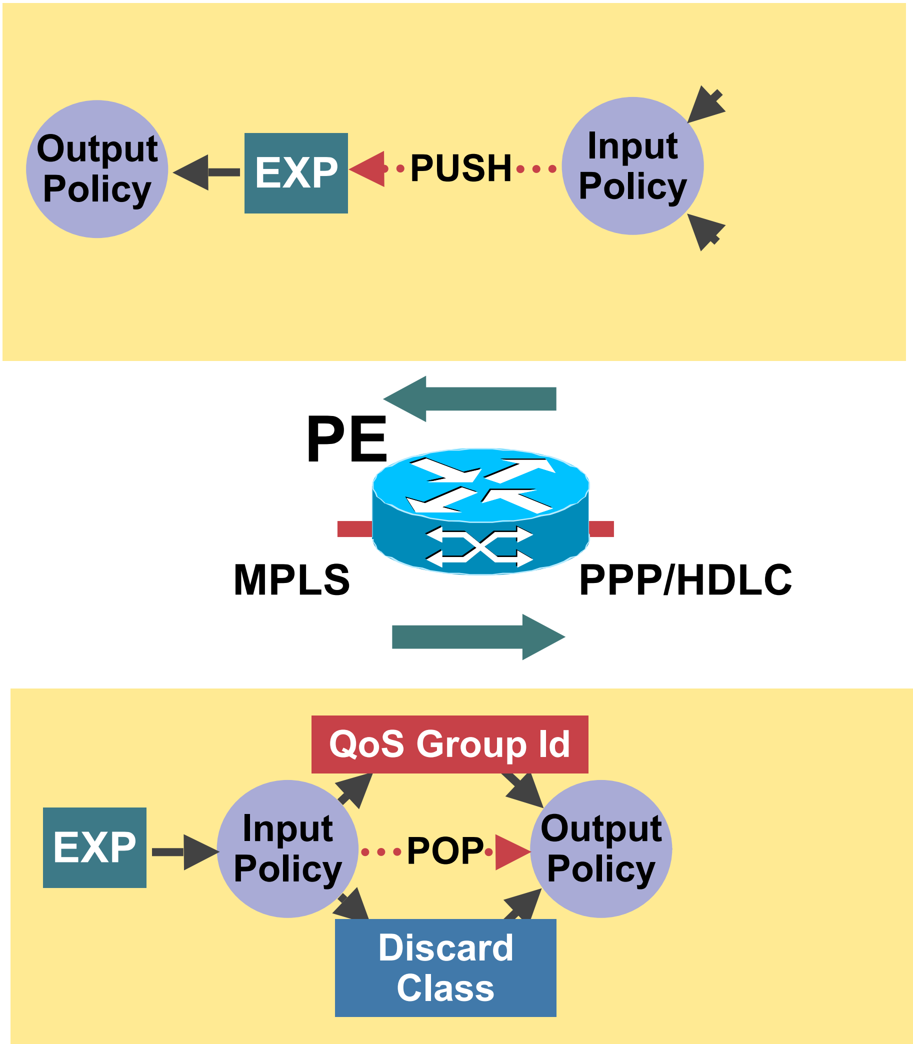


Layer-2 QoS: Ethernet



- **Incoming traffic classified by CoS (802.1p)**
- **Service characteristics being proposed at the Metro Ethernet Forum (BW Profile: CIR, CBS, EIR, EBS, CF, CM)**
- **Site-to-Network (point-to-cloud) SLA for VPLS**
- **Control word does not carry any CoS (802.1p) info**

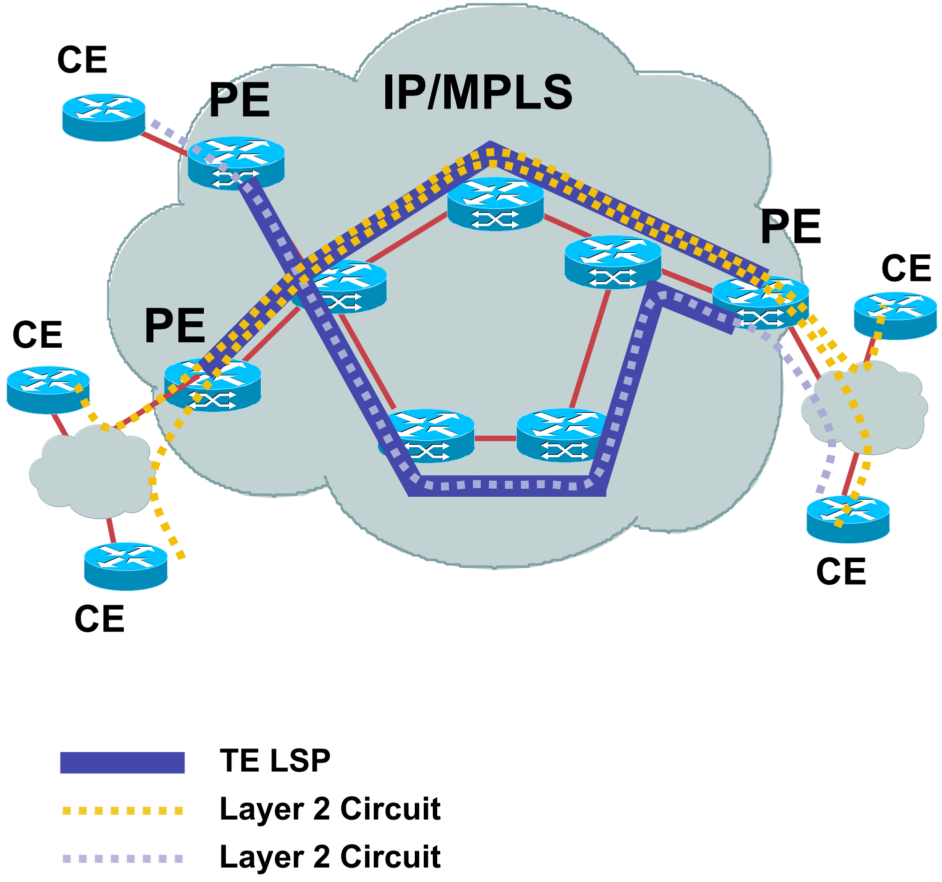
Layer-2 QoS: PPP/HDLC



- No layer-2 marking to set or classify on
- No standard service definition but classes of service are possible

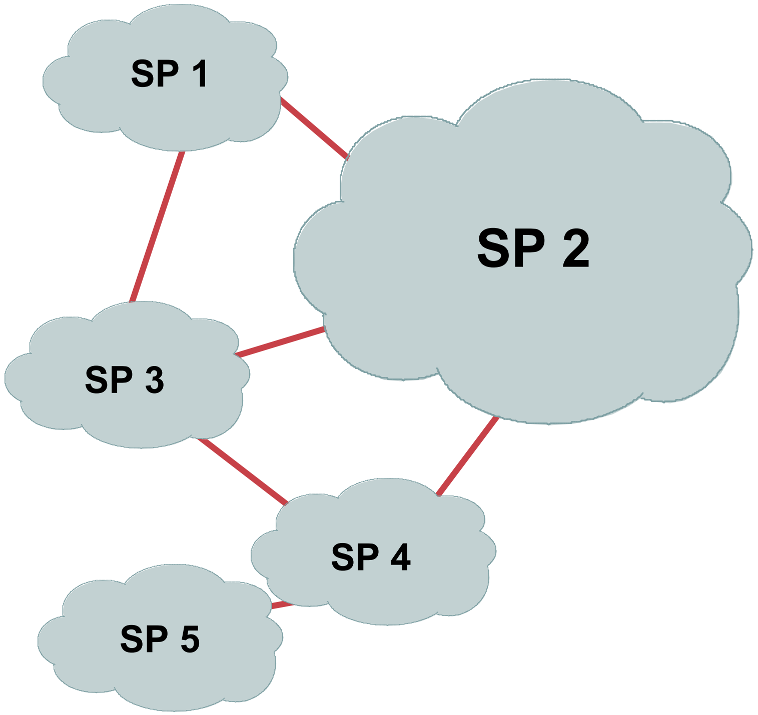
Coupling Layer-2 Services with MPLS TE Tunnel Selection

- Static mapping between pseudo-wire and TE Tunnel on PE
- Implies PE-to-PE TE deployment
- TE tunnel defined as preferred path for pseudo-wire
- Traffic will fall back to peer LSP if tunnel goes down



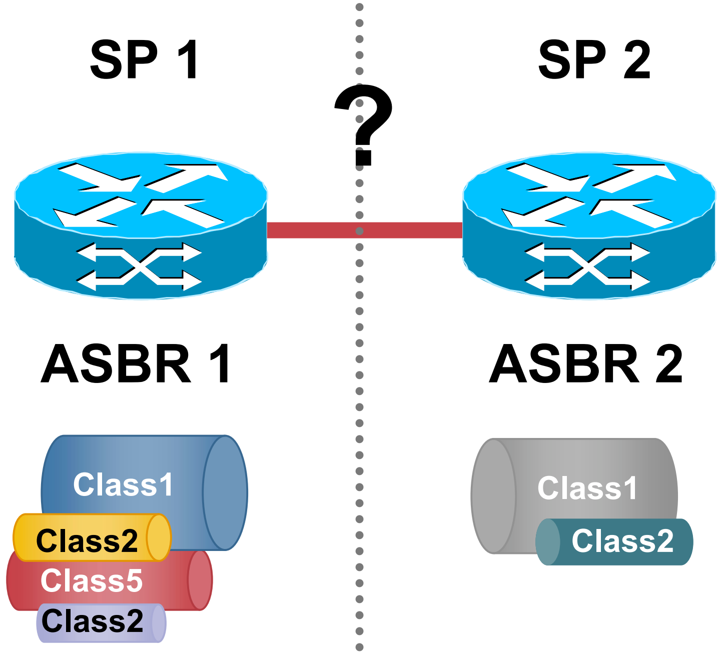
INTERPROVIDER QOS

Interprovider QoS



- **Current efforts to standardize and define framework**
- **Areas of focus**
  - **Service class definition**
  - **Signaling/protocol QoS extensions**
  - **SLA budgets and monitoring**
- **Standard bodies/forums**
  - **Interprovider QOS Working Group at MIT Communications Futures Program**
  - **IETF (PW3E, PCE)**
  - **ITU (NGN)**

Interprovider Service Class Definition



Class	Delay	Jitter	Loss
Class 1	Low	Low	Low
Class 2	Low	NA	Low
Class 3	NA	NA	Low
Class 4	NA	NA	NA

Class	Delay	Jitter	Loss
Class 1	Low	Low	Low
Class 2	NA	NA	Low

- Standard service class definition to facilitate interconnection
- Standardization and differentiation are opposite goals
- MIT QoSWQ focusing on small number of classes
- draft-baker-diffserv-basic-classes-04.txt proposes three control/mgmt classes and ten application/ subscriber classes

Signaling/Protocol QoS Extensions

- **Current signaling capabilities**

  QPPB: no QoS intelligence in BGP, routing info used to influence QoS

  Inter-AS TE: resource reservation and protection across multiple autonomous systems

- **Early discussions for new protocol extensions**

  QoS extensions to BGP (multi-topology routing), QoS info used to influence routing

  QoS extensions to PW signaling (traffic profile and QoS requirements), specially for multi-segment PW

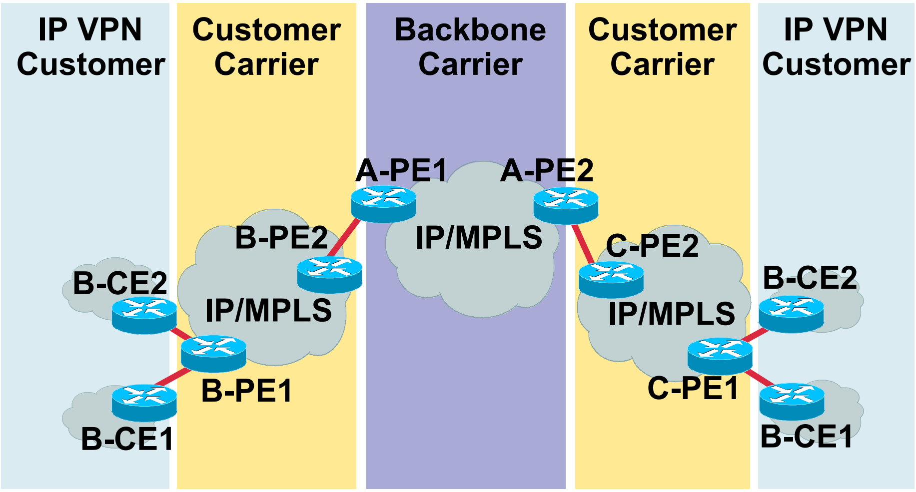
SLA Budgets and Monitoring

Issues:

- End-to-end SLA budgeting
- Common metric definitions
- Standardization of performance monitoring technology
- Monitoring accuracy vs. scalability (end-to-end, additive?)

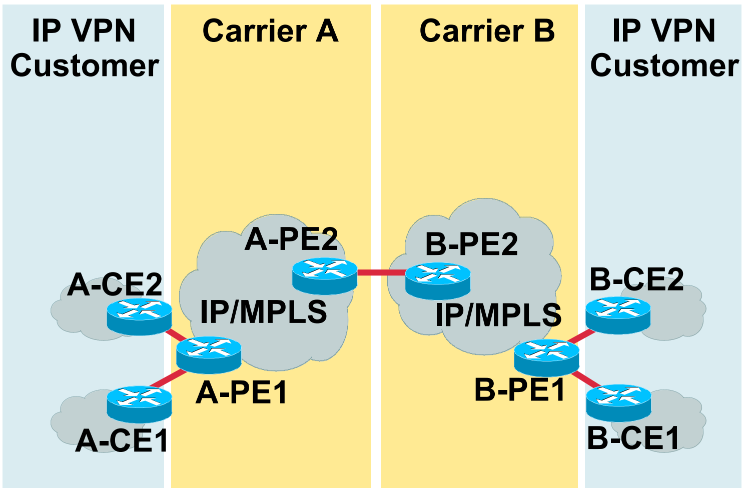
Interprovider QoS Capabilities Today

Carrier Supporting Carriers (CsC)



- Supports MPLS DiffServ tunnel modes
- No need to remark customer carrier traffic

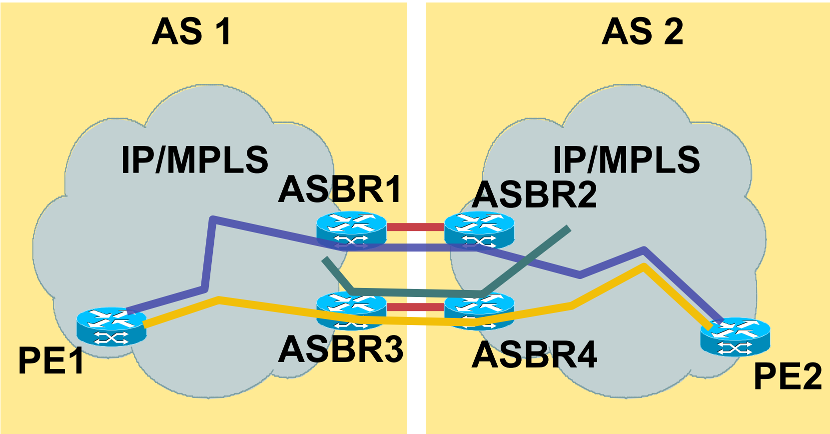
Inter-AS



- Option A exposes customer markings, but provides granular control
- Option B/C provides aggregate QoS and may require EXP remarking

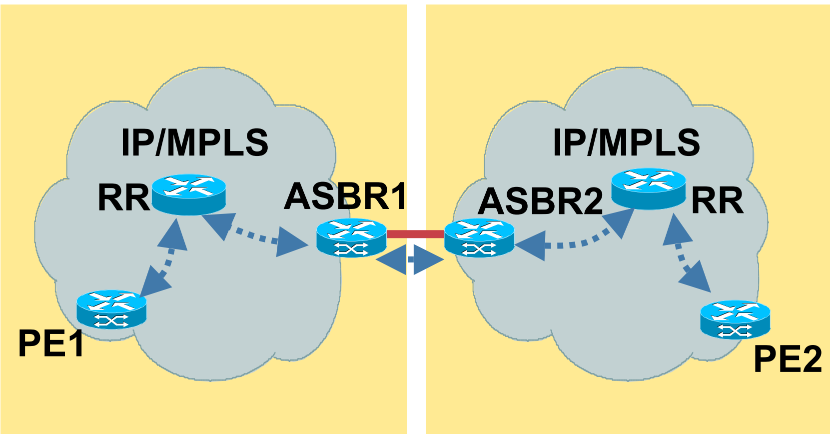
Interprovider QoS Capabilities Today (Cont.)

Inter-AS TE



- Bandwidth reservation across autonomous systems
- Signaled protection requirements
- Support for DS-TE

QPPB



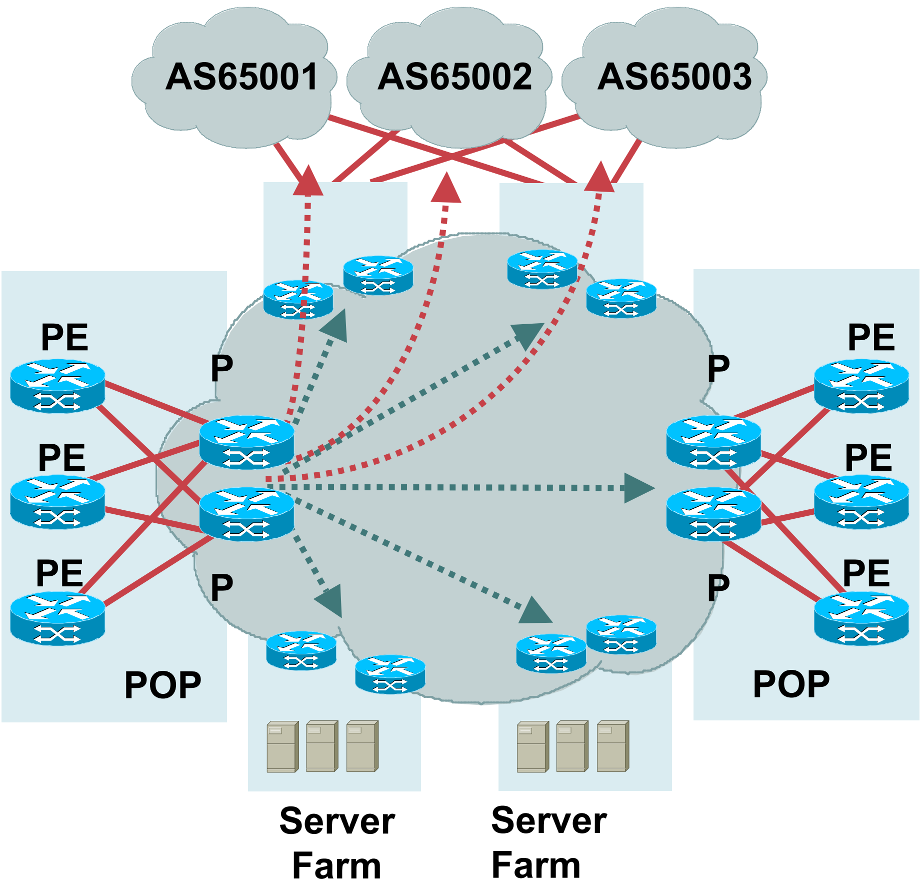
- Applicable to Inter-AS and CSC
- Routing attributes influence QoS policies

MPLS QOS MANAGEMENT

Some Monitoring Tools: Monitoring Utilization Level (x%)

Measuring Internal and External Traffic Matrix

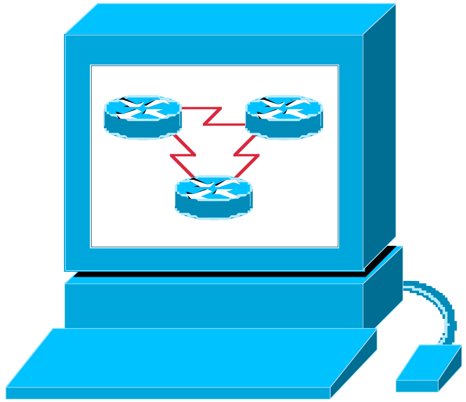
- Interface MIB
- MPLS LSR MIB
- Cisco class based QoS MIB
- NetFlow
    - NetFlow BGP Next Hop
    - MPLS-Aware NetFlow
    - Egress/Output NetFlow
- BGP policy accounting
    - Communities
    - AS path
    - IP prefix



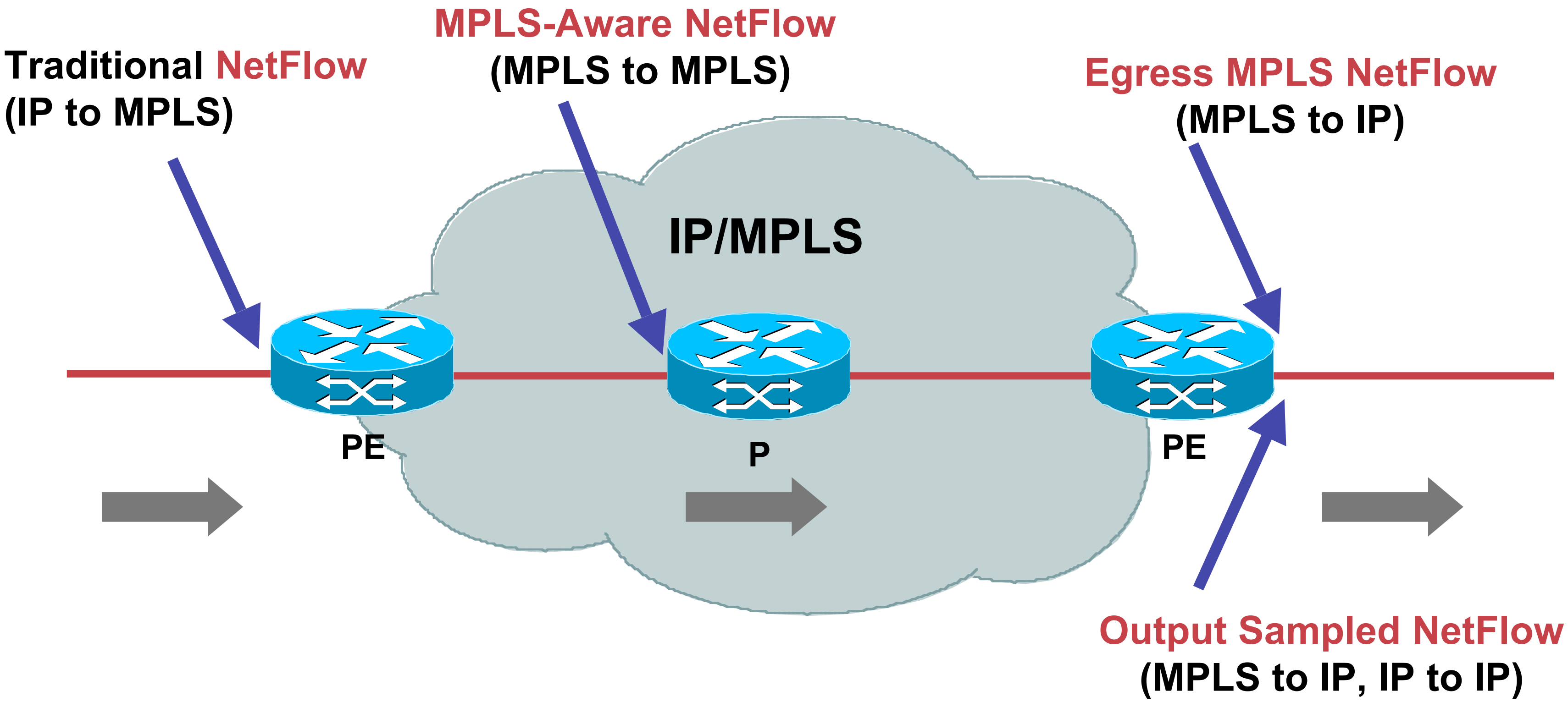
Cisco Class-Based QoS MIB

- **Primary per-link accounting mechanism for QoS:**
  - **Classification** (cbQosMatchStmtStats/ cbQosClassMapStats)
  - **Marking** (cbQosClassMapStats)
  - **Policing** (cbQosPoliceStats)
  - **Shaping** (cbQosTSStats)
  - **Congestion management** (cbQosQueueingStats)
  - **Congestion avoidance** (cbQosREDClassStats)
- **QoS policy must be applied to interface/PVC for accounting to happen**
- **Read access to configuration and statistical information for MQC**

Management Station



NetFlow MPLS Features Overview



Lots of Detailed Info in Session NMS-3132

NetFlow Partners

Traffic Analysis

CRANNOG SOFTWARE

NetQoS

QUALLABY CORPORATION

Flow-Tools

MICROMUSE NETCOOL® SOLUTIONS

hp invent

CISCO SYSTEMS
EMPOWERING THE INTERNET GENERATION

IBM®

caida

EVIDENT™ SOFTWARE

ARBOR™ NETWORKS

Adlex

InfoVista™
Business Oriented Service Level Management

WiredCity

CONCORD

Denial of Service

PROTEGO NETWORKS

ARBOR™ NETWORKS

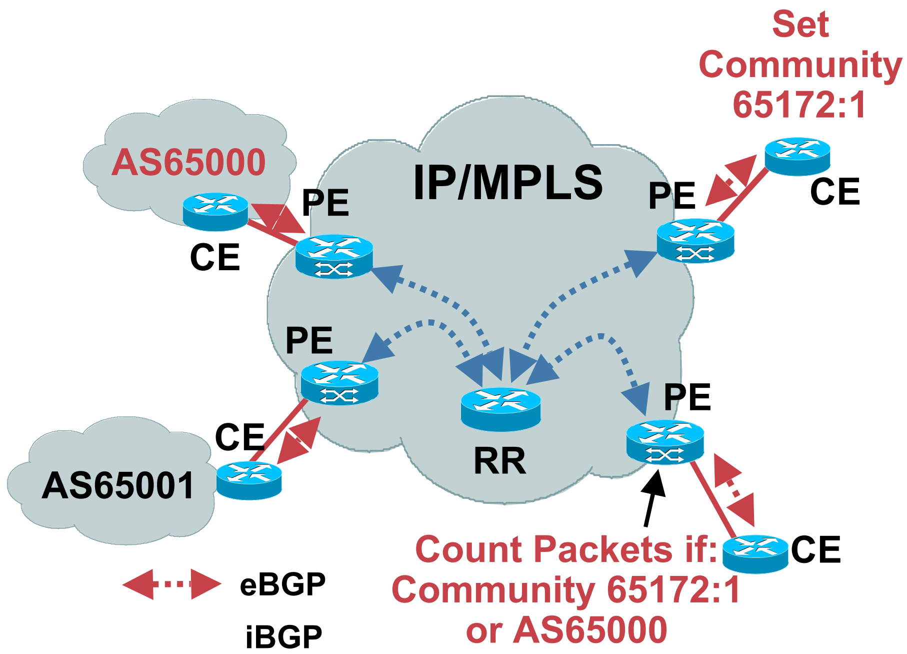
Adlex

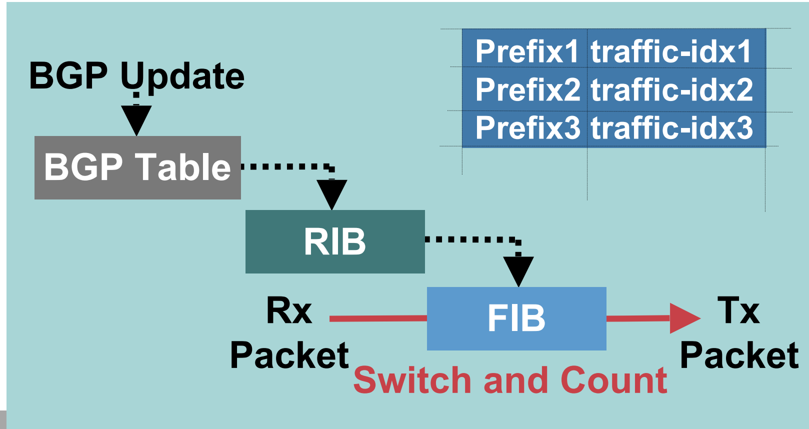
Billing

PORTAL
Real Time No Limits

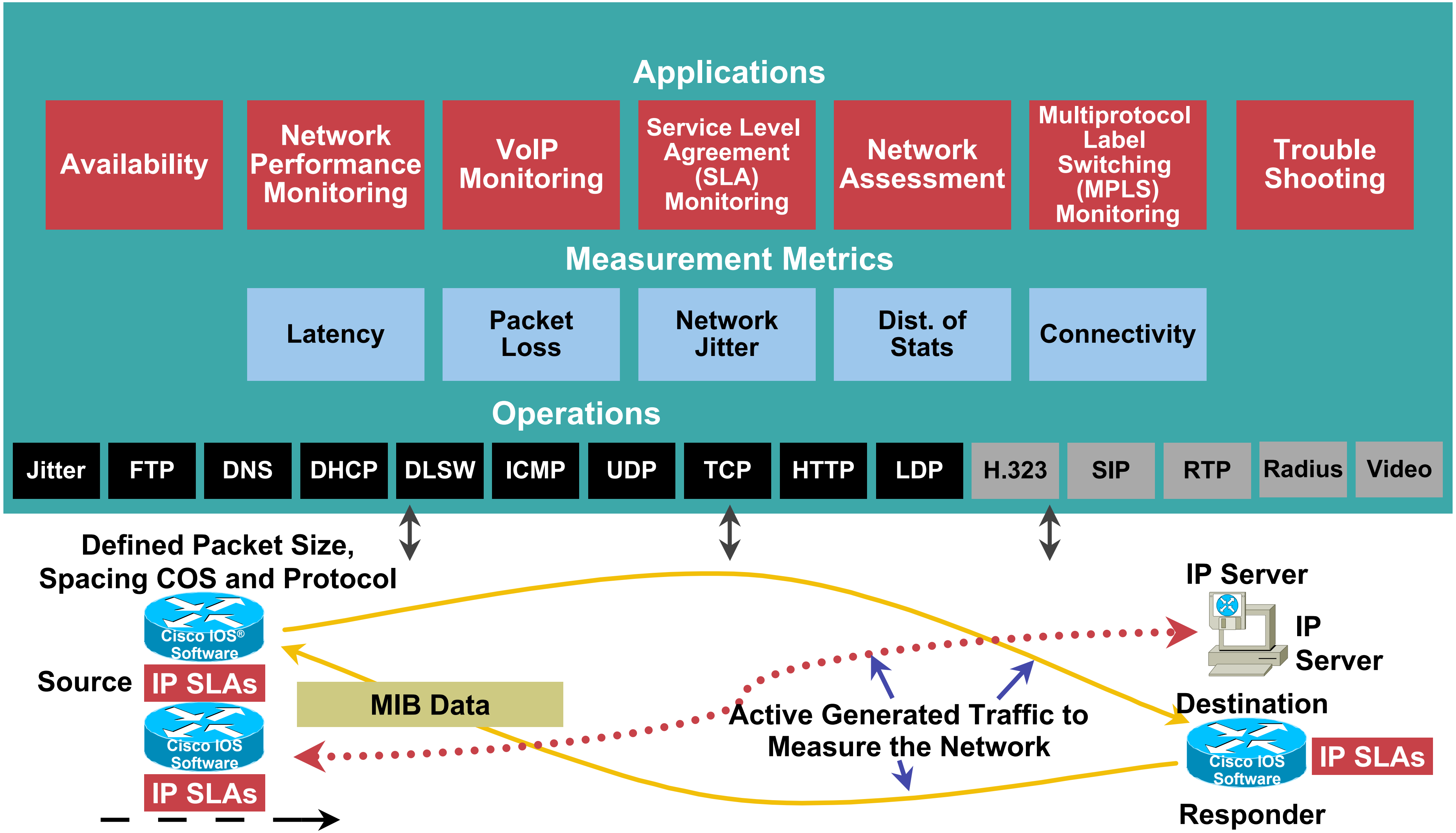
BGP Policy Accounting

- **Assign counters (traffic-index) to IP traffic based on:**
  - **Community**
  - **AS path**
  - **IP prefix**
- **Up to 64 counters (traffic-index)**
- **Supports IPv4 and VPNv4 addresses**
- **Similar in concept/operation to QPPB, but accounting instead of marking**





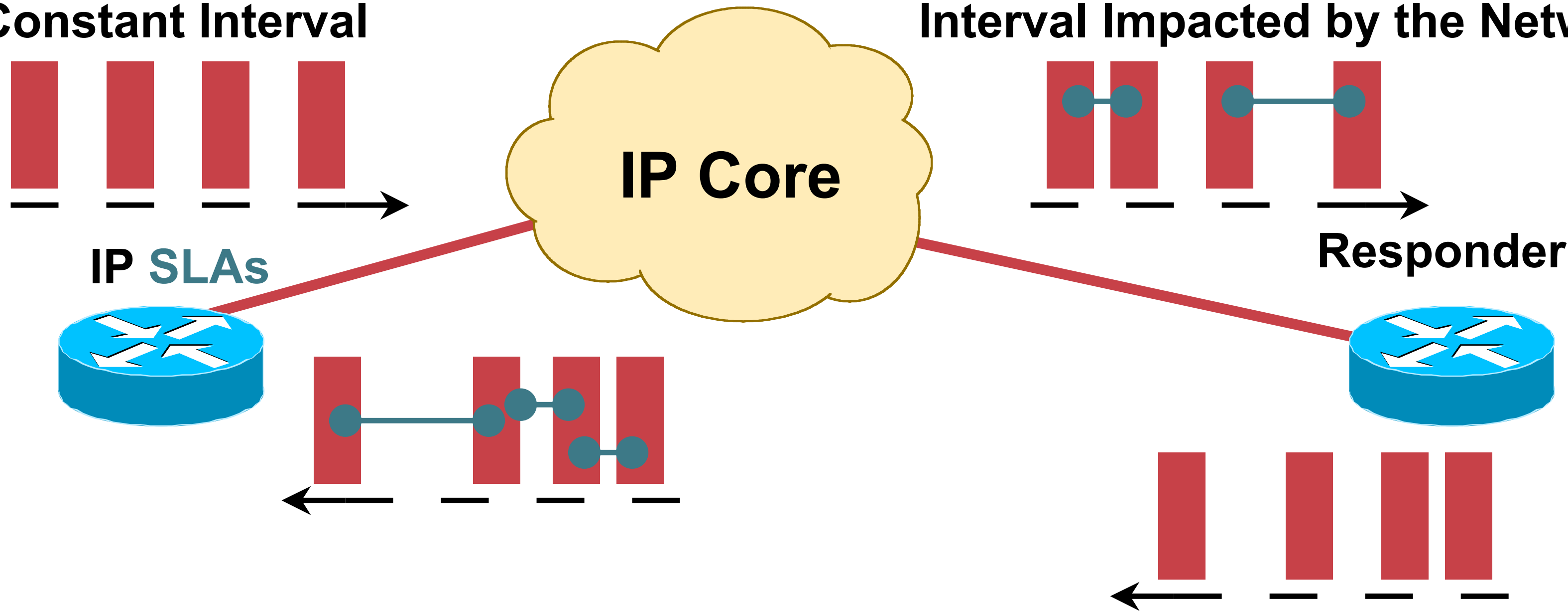
Example: Multi-Protocol Measurement and Management with Cisco IOS IP SLAs



UDP Jitter Operation
Packet Stream

Sends Train of Packets with Constant Interval

Receives Train of Packets at Interval Impacted by the Network



Add a Receive Time Stamp and Calculate DELTA (the Processing Time)

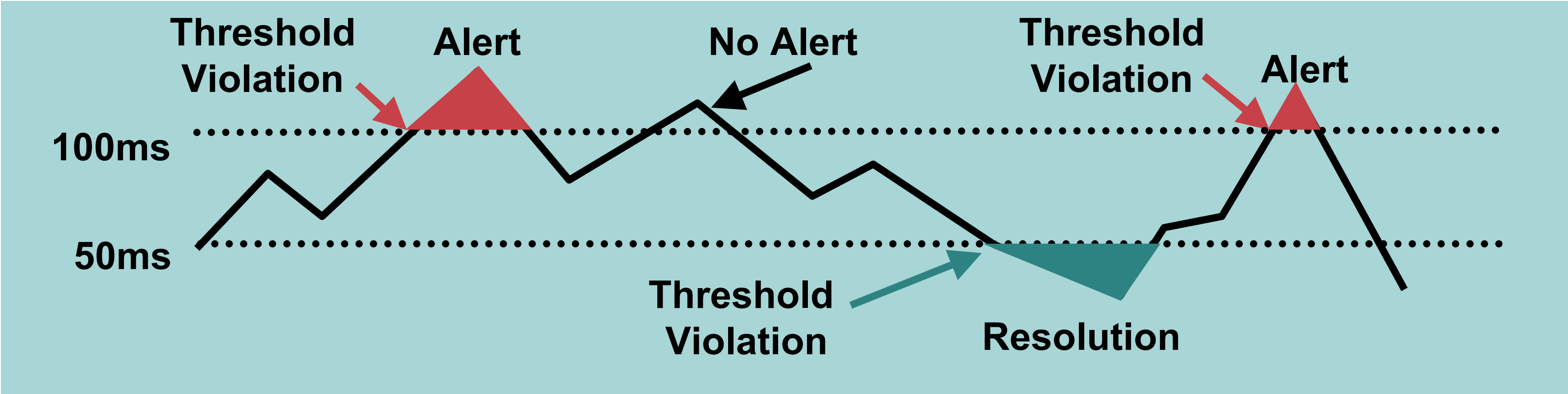
Responder Replies to Packets (Does Not Generate Its Own)

Per-Direction Inter-Packet Delay (Jitter)

Per-Direction Packet Loss

Average Round Trip Delay

Cisco IP SLA Reaction Conditions



Event Triggers

- Connection loss/timeout
- Latency (one way, round trip)
- Jitter (one way, round trip)
- Loss (one way, round trip)
- MOS

Trigger Threshold Definitions

- Immediate
- Average
- Consecutive
- X out of Y times

Triggers Can Generate SNMP Trap or Another Probe

Cisco IOS IP SLAs Partners

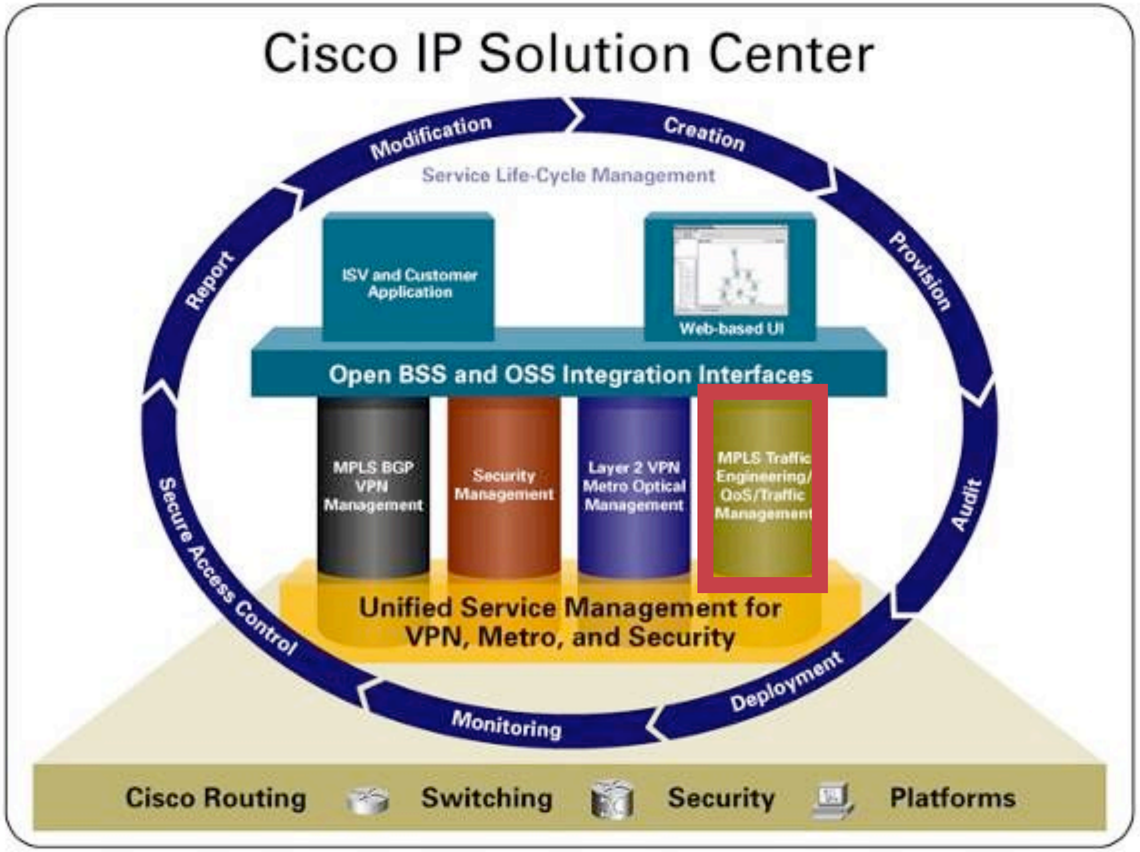
Cisco Network Management Solution	
Cisco IP Solution Center	MPLS VPN and SLA Monitoring
CiscoWorks IP Telephony Monitor	Telephony Monitoring
Internetworking Performance Monitor	Enterprise Performance Measurements

THIRD PARTY PRODUCTS

HEWLETT PACKARD Expanding Possibilities
concord
InfoVista Seeing IT
IBM
Agilent Technologies
MICROMUSE NETCOOL SOLUTIONS
QUALLABY CORPORATION
WiredCity
CRANNOGSOFTWARE

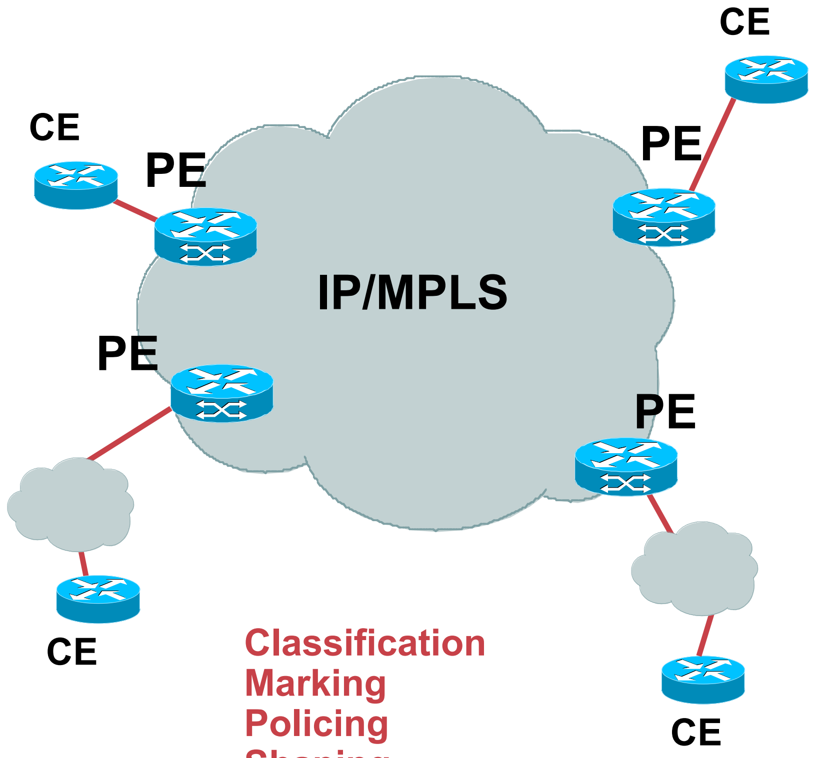
Provisioning: Cisco IP Solution Center

Unified Management for MPLS VPN, L2VPN, Security, and MPLS TE



ISC QoS Management Features

- QoS provisioning on access link (both CE and PE)
- Internal constrain matrix check software and hardware dependencies
- Support for pre-MQC QoS functionality
- QoS provisioning on backbone links using Smart Template utility



Classification
Marking
Policing
Shaping
Congestion Management
Congestion Avoidance
LFI
cRTP

ISC TE Management Features

- **Discovery and Audit**
  - **TE enabled devices and tunnels**
  - **Visualization and tunnel audit**
- **Bandwidth Protection during element failure**
  - **FRR tunnel audit and calculation**
- **Primary tunnel placement & repair**
- **Global optimization of network utilization**
- **Deployment and tunnel activation**