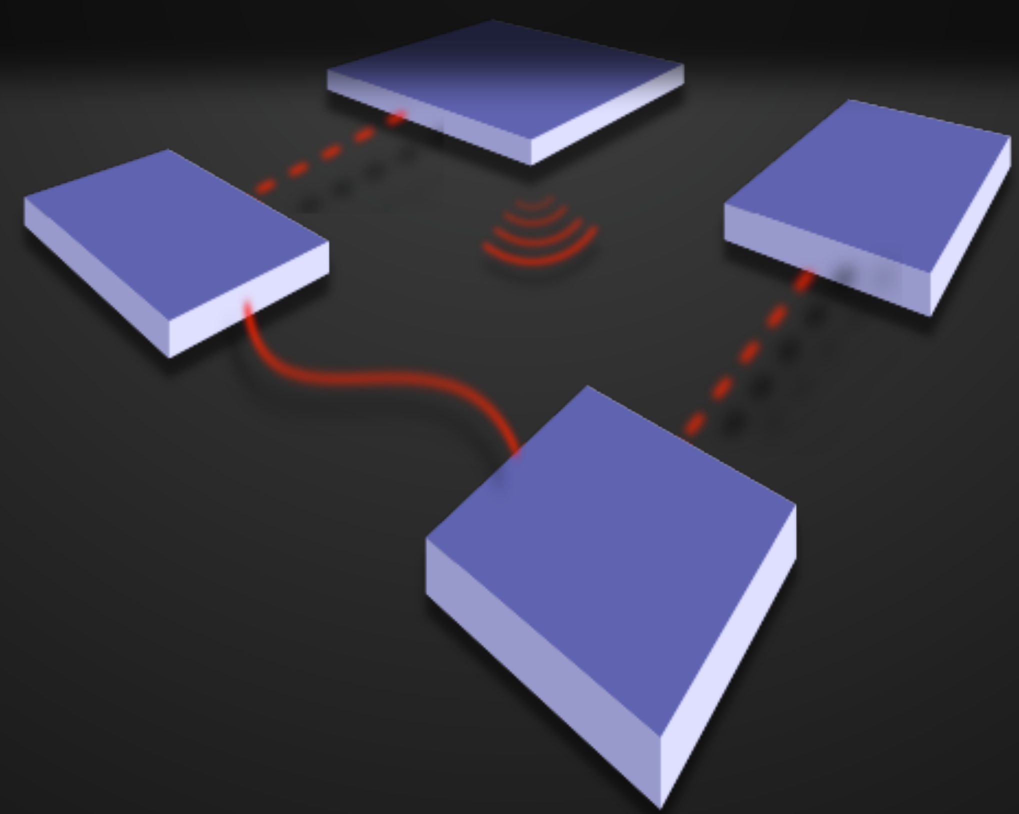
CS-435
spring semester 2025

Network Technology & Programming Laboratory

University of Crete
Computer Science Department

Stefanos Papadakis



CS-435

Lecture #08 preview

- QoS
  - IntServ
  - DiffServ
  - Tags
- RSVP
- MPLS

IETF Integrated Services

- Current Internet Protocol (IPv4) provides best-effort service.
- Congestion degrades TCP/IP performance.
- The Internet Engineering Task Force (IETF) first developed the "**Integrated Services**" model to provide QoS in the Internet (IntServ).
  - the router reserves resources **for each individual flow**.
- **RSVP** was/is the control protocol to implement the Integrated Services QoS model.

IETF Differentiated Services

- However, a core network IP-router may support millions of flows. Reserving resources in the router for each flow is infeasible.
- The IETF then introduced the “**Differentiated Services**” Model (DiffServ),
  - a simpler and
  - more scalable QoS protocol.
- The key idea is to aggregate multiple traffic flows into **a single aggregated traffic class**, and offer QoS for the entire aggregated traffic class
- DiffServ supports multiple **traffic classes**, and resources are reserved on an end-to-end path **for each class**

IETF IntServ

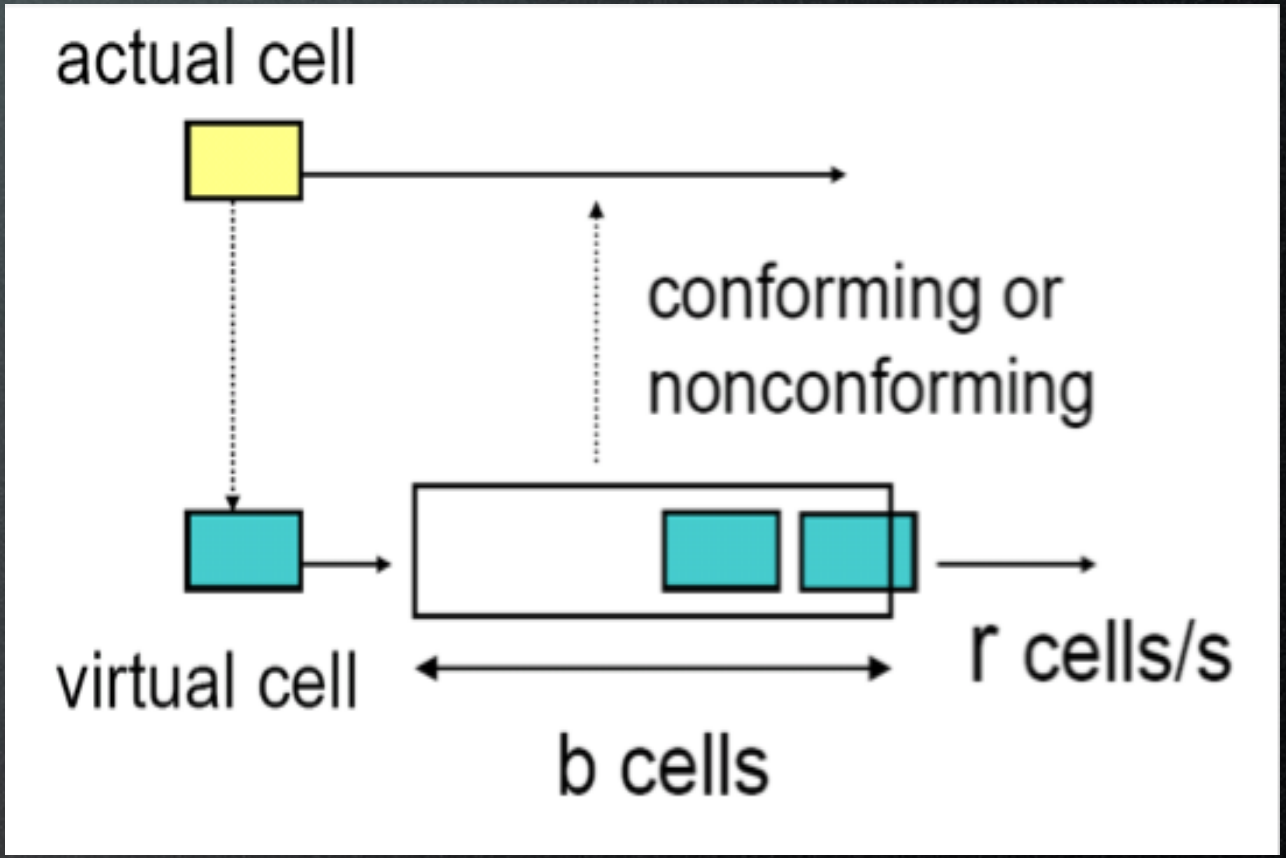
- Connection-oriented solution (end-to-end)
- QoS guarantees on a per flow basis
- Intermediate routers keep per flow state
- Building blocks:
    - resource reservation protocol (RSVP): end-to-end signaling
    - admission control
    - policing: check if traffic conforms to profile
    - shaping: modify traffic timings so that it conforms to profile
    - classification: identify packets that are to receive certain level of service
    - scheduling: isolate flows and support minimum bandwidth

IETF IntServ

- Guaranteed Service:
  - deterministic delay guarantee (provable)
  - zero packet loss
  - token bucket used to specify traffic
  - specification of requested service
- Controlled-Load Service:
  - network provides service close to that provided by a best-effort network under lightly loaded conditions
  - token bucket used to specify traffic
- Best-Effort Service:
  - no guarantees

Remember: ATM Leaky bucket

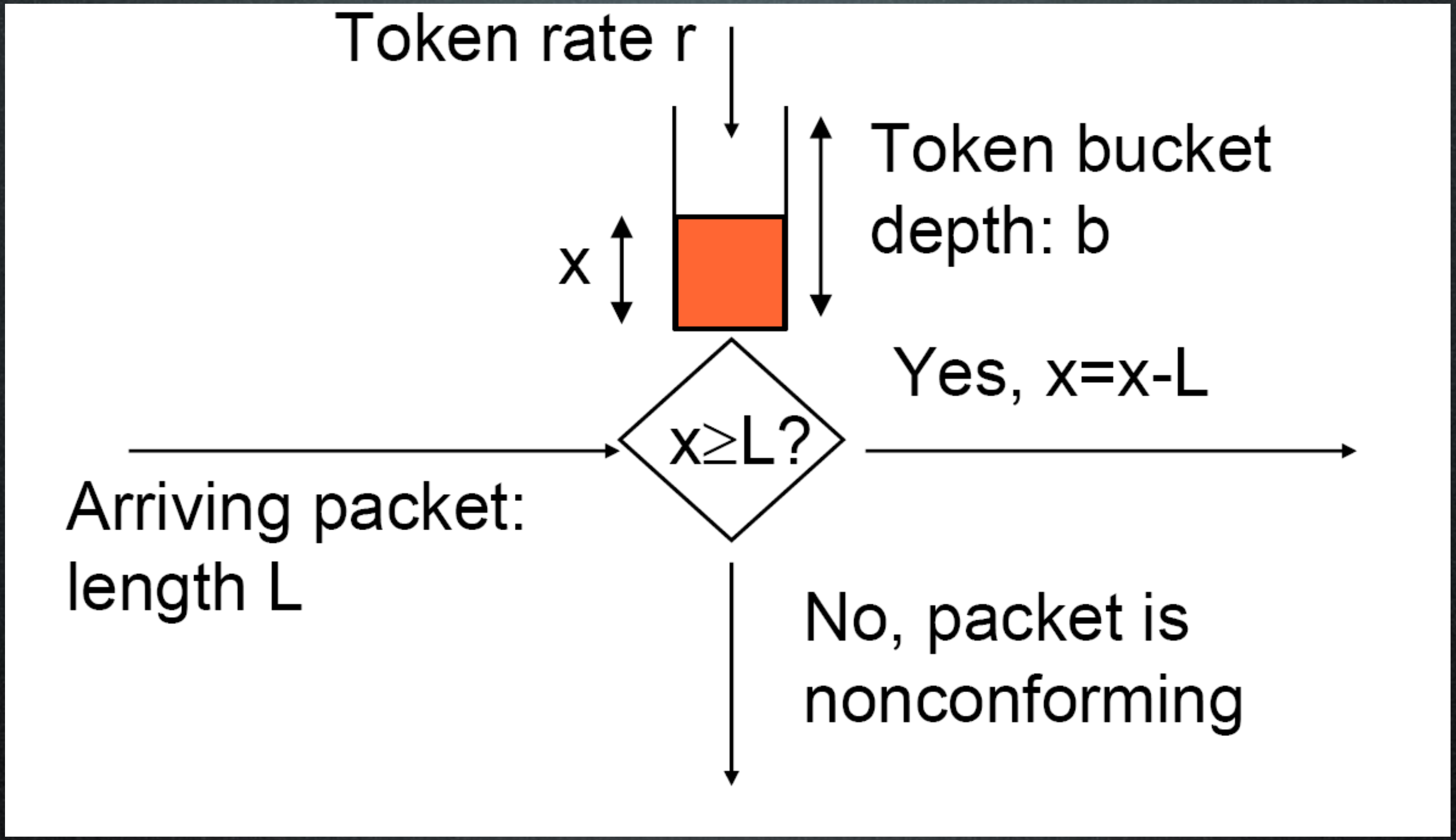
- Bucket size b
- Leak rate r



Bucket contents ‘B’ increase by 1 for each conforming cell

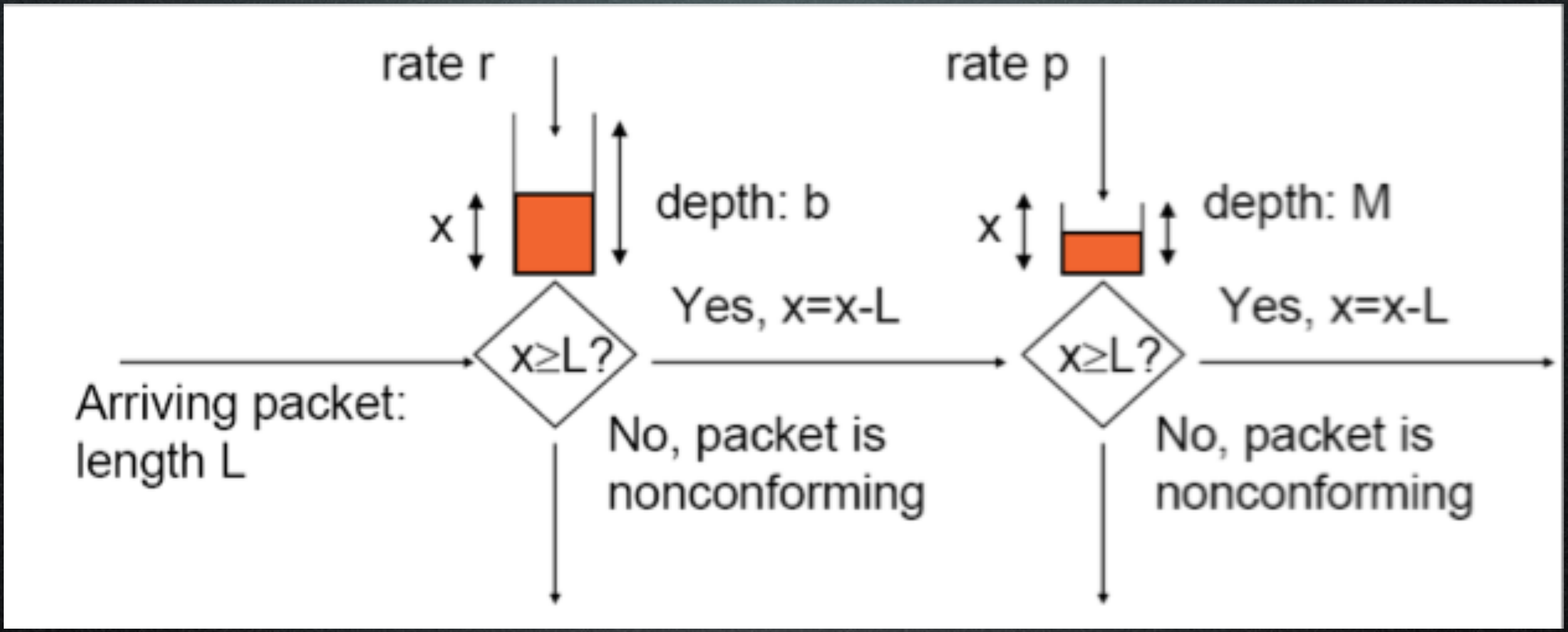
```
if B+1 > b
   cell non conforming
else
   cell conforming
   B = B+1
```

Token bucket policing



- Equivalent to leaky bucket
- Amount of data over period of time T :
- $D(T) \leq rT+b$

Complete token bucket specs



- Three additional parameters:
  - minimum policed unit **m**: policing required to remove at least m tokens for each **conforming** packet
  - maximum packet size **M**: largest permissible packet size
  - peak rate **p**

Guaranteed QoS service class

- Traffic Specification: Tspec=(r, b), p, M, m
- Service request specification: Rspec=(R,S) minimum reserved capacity
- S is a slack term representing the difference between the required delay and the maximum delay using reservation R
  - controls maximum delay, not minimum, average, or jitter

Bottom line

- Parameter selection:

  - Given Tspec, Dtot, Dmax the application sees:

    R (= Rspec)

  - Given Tspec & Rspec network chooses the buffers required for zero packet loss

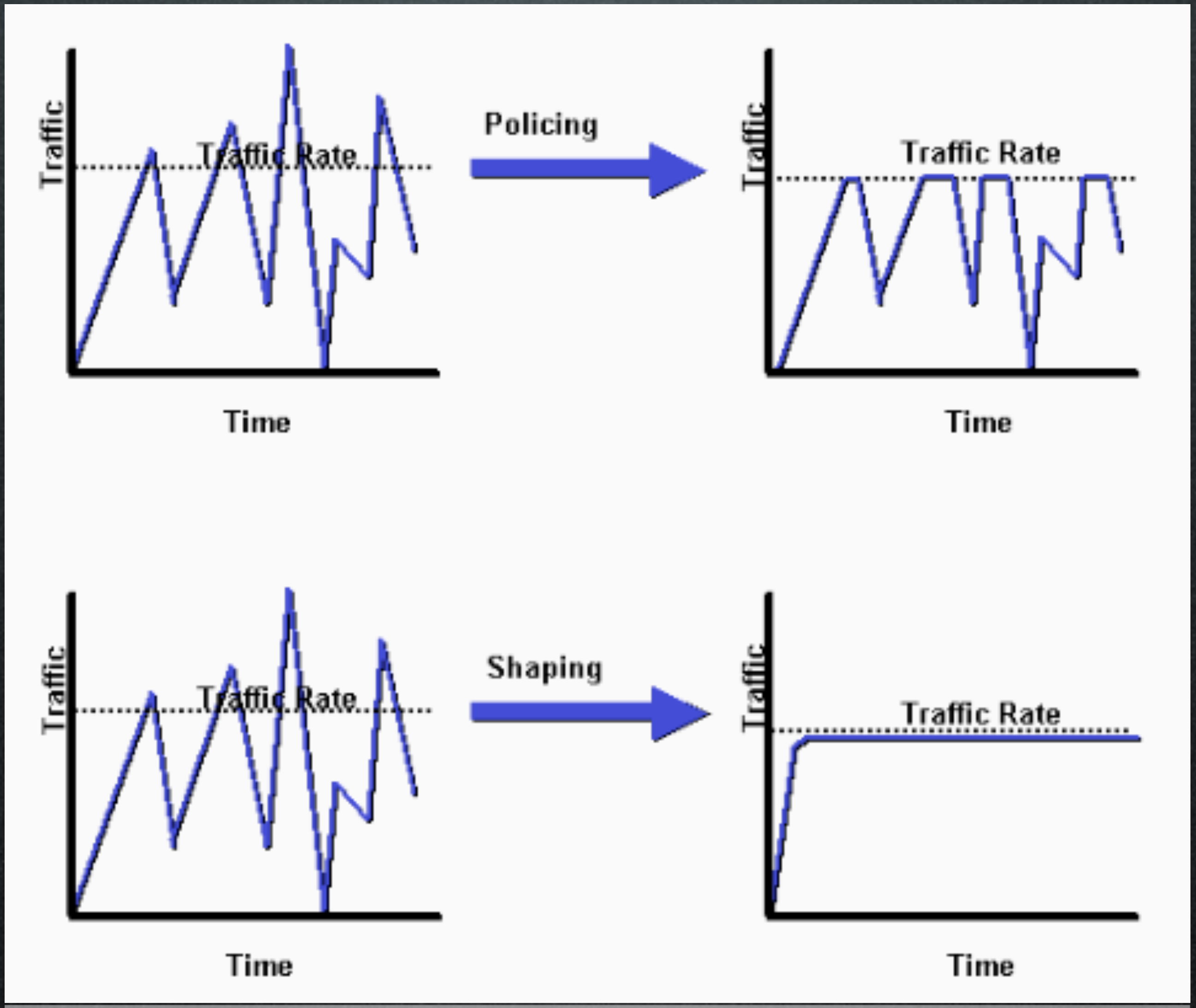
Controlled-load service class

- Intended to support applications highly sensitive to overloaded conditions
- Service provided tightly approximates service of best-effort networks under **unloaded** conditions
  - A very high percentage of transmitted packets will be successfully delivered
  - transit delay experienced by a very high percentage of delivered packets will not greatly exceed minimum transmit delay
- Uses only Tspec = (r, b), p, M, m and not Rspec

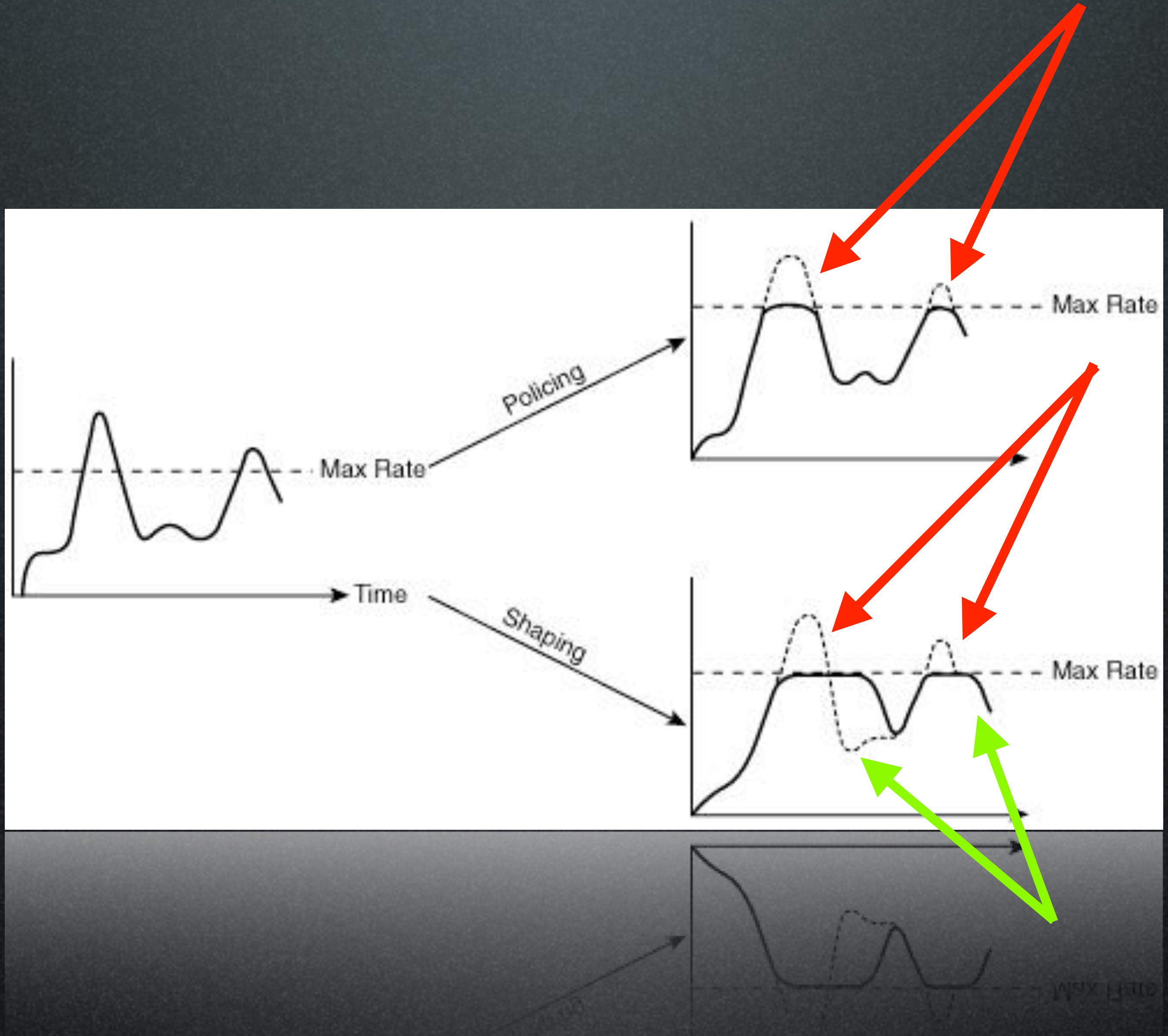
Policing / Shaping

- **Policing** performed at **ingress** of network
  - non-conforming packets treated as best-effort
  - possibility of out of order delivery (bad, e.g. for real-time)
- Re-shaping done at intermediate point of the network
  - may be necessary due to distortions as traffic flows through network
  - normalizes bursty traffic

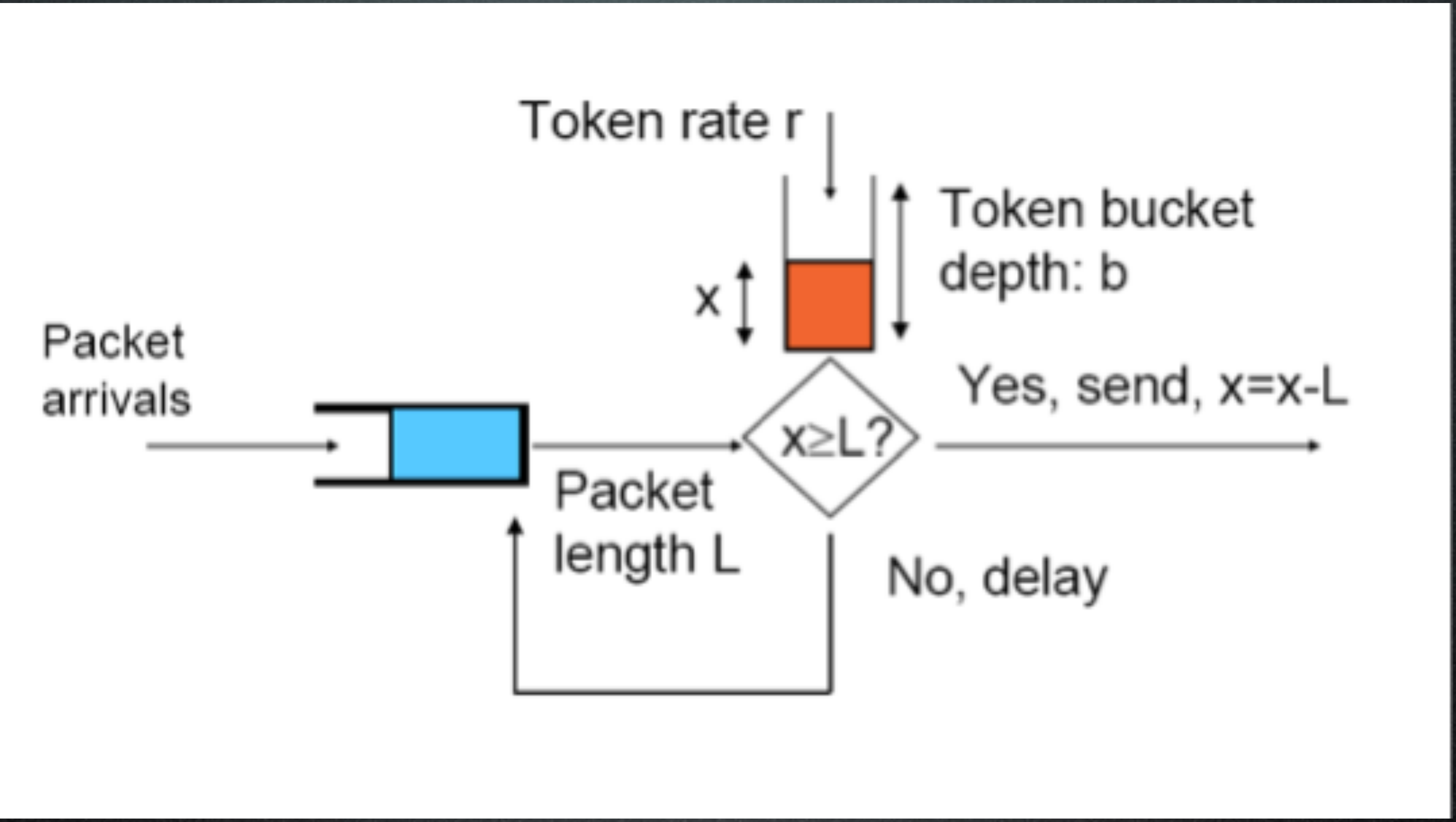
Policing / Shaping



Policing / Shaping



Token bucket shaping

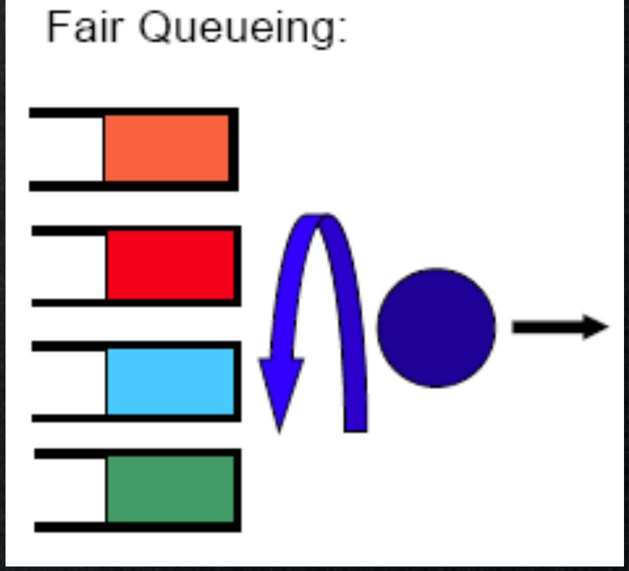


- Under stable conditions, tokens flow and data flow would match

Scheduling

- give different flows a different bandwidth share
- support minimum bandwidth guarantees
- isolation: one flow cannot monopolize whole resource
- implementation, admission control decisions, etc

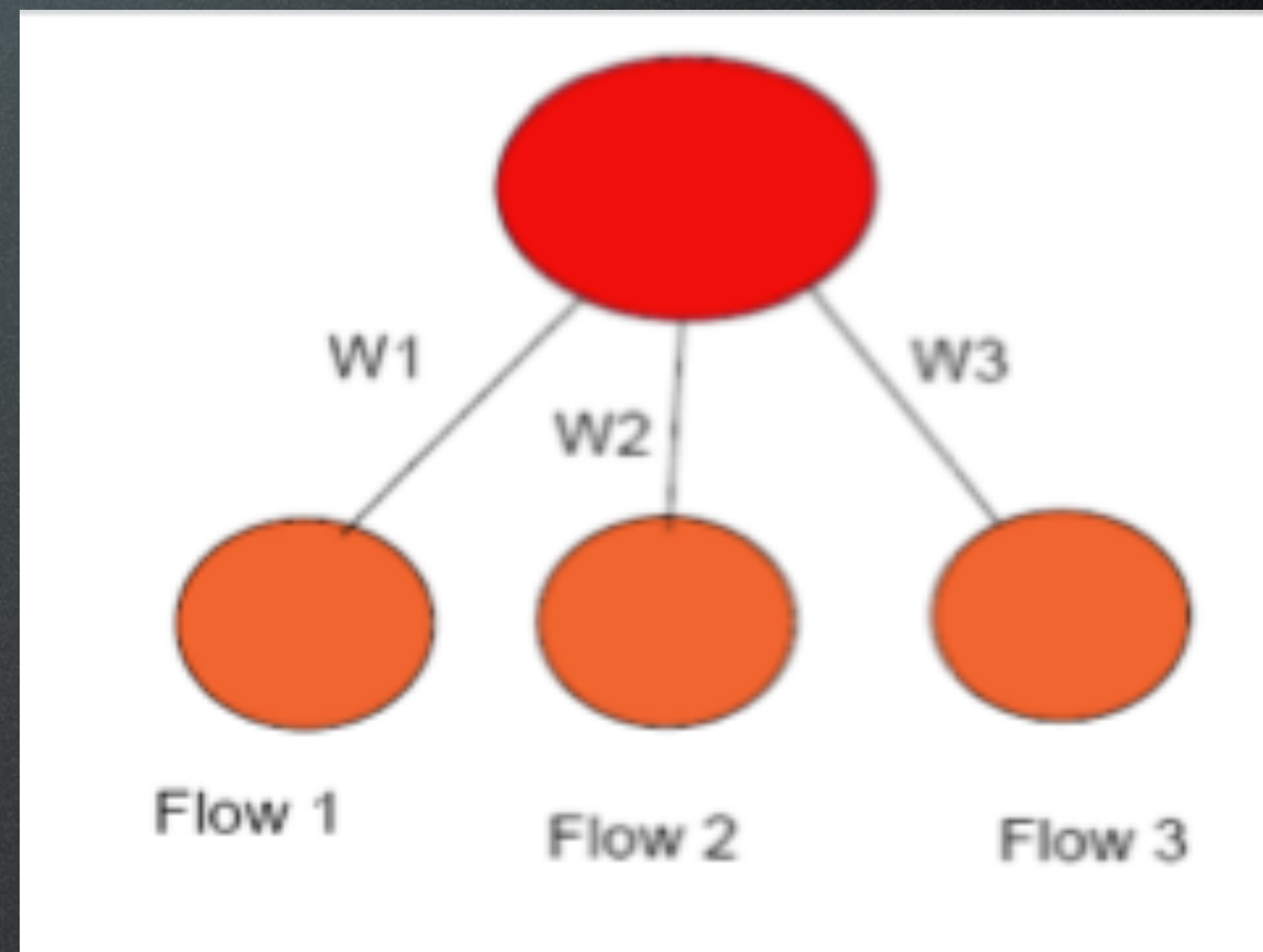
- Schemes:
  - FIFO
  - Priority Queuing
    - high priority can starve lower priority
  - Fair Queuing/Weighted Fair Queuing
    - each flow gets share of bandwidth
    - isolation of flows
  - Class Based Queuing
    - proportional bandwidth sharing among classes



Weighted Fair Queuing (WFQ)

- In congestion:

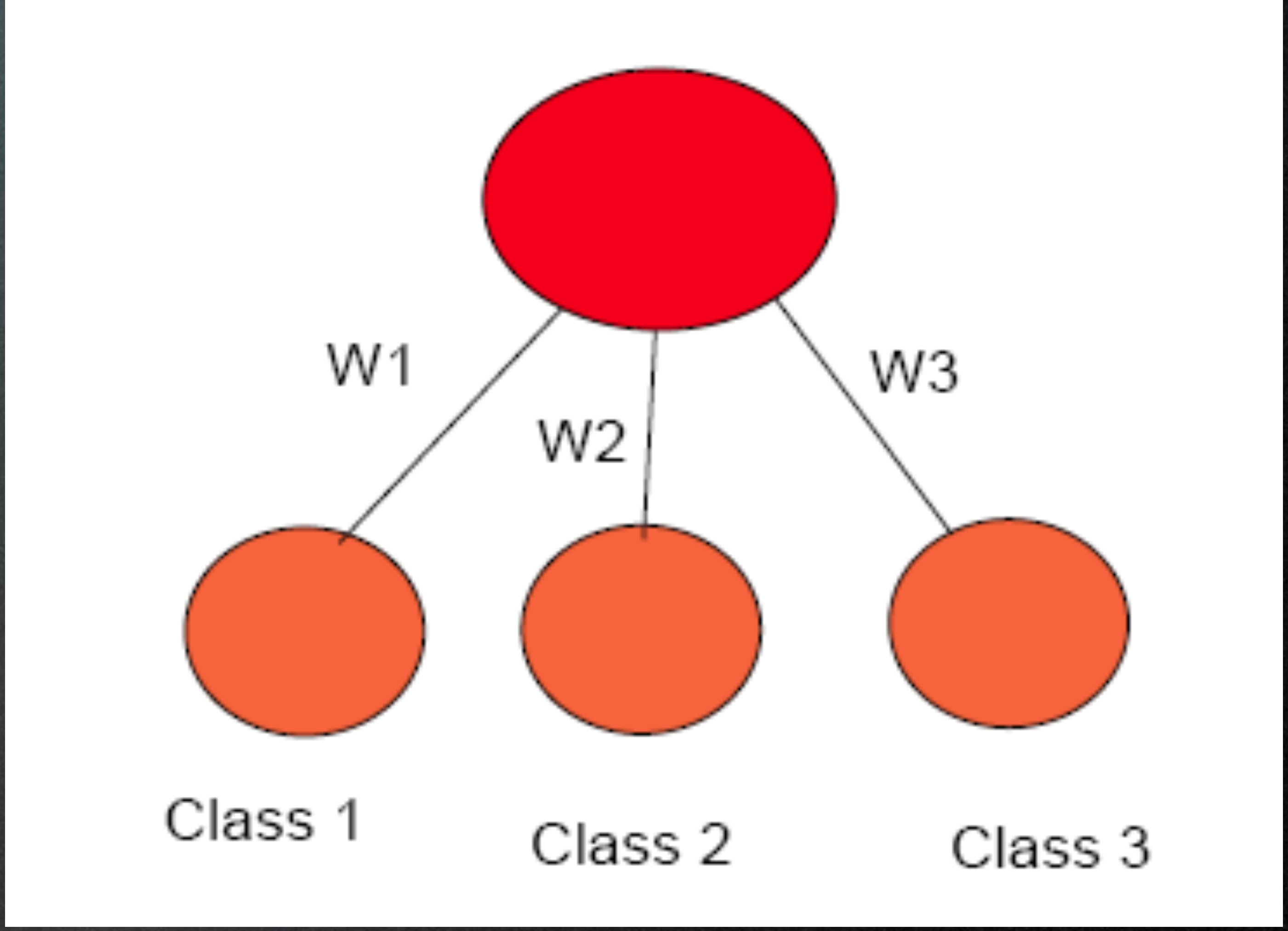
$$Ri = \frac{Wi}{\sum_{j} Wj} \times C$$



Class-based Fair Queuing (WFQ)

- In congestion:

$$Ci = \frac{Wi}{\sum_j Wj} \times C$$



IntServ and ATM

- Similarities
  - Both require signaling
  - Both operate on per flow basis
  - Both use admission control

IntServ and ATM

- Differences
  - ATM: hard state
  - IntServ: soft state
    - need to periodically refresh reservation
    - refresh request can be denied
    - user allowed to change reservation
  - ATM more “predictable” network
  - ATM QoS negotiable
  - IntServ: Guaranteed service determined from Tspec,Rspec; not negotiable for controlled load

IntServ and ATM

- Issues
  - Complexity in routers: packet classification & scheduling
  - Scalability in core since both operate on per-flow basis
  - Ease of deployment
  - Need concept of “virtual paths” or aggregated flow-groups in core

DiffServ

- Goal: offer differing levels of performance (Quality of service - QoS) to different users
- improve revenues (premium pricing)
- competitive differentiation
- Key concepts:
    - scalability
    - simple model:
        - traffic entering network is classified into a small number of classes
        - a class (“behavior aggregate”) is characterized by a **tag**
        - a router services packets according to the tags
        - QoS per class (aggregate traffic), **not** per individual flow
    - keep forwarding path simple to allow easy and early deployment;
    - push complexity to network edge

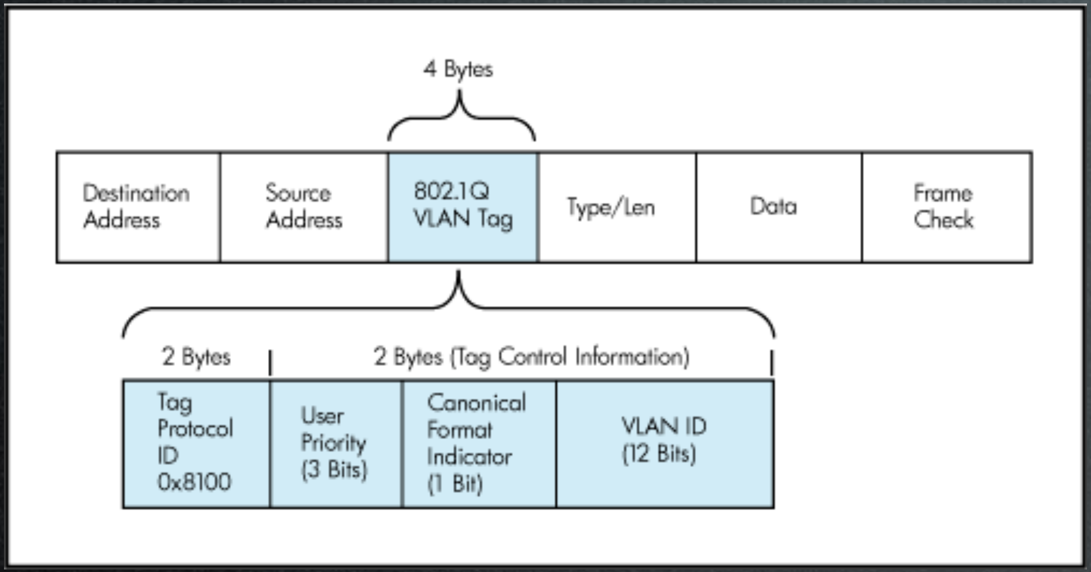
DiffServ

- Key concepts:
  - avoid “strong” assumptions on traffic types
  - marking based on static/long term “Service Level Agreements” (SLAs); **avoids signaling**
  - don’t develop/specify services, but rather standardize “Per Hop Behaviors” (PHBs); but leave some DS Code Point patterns for experimental and local use
  - use PHBs to construct services
  - ability to provide services depends on ability to manage and configure routers in a coordinated manner

QoS Tags

- CoS - Class of Service
    - IEEE 802.1p / 802.1Q
- ToS - Type of Service
    - DSCP - Differentiated Services Code Point

Class of Service



- Layer 2
- inside extra header
  - IEEE 802.1Q
- 3bit [PRI - user priority]

IEEE 802.1p	
Priority	**Traffic Type**
0	Best Effort
1	Background
2	Spare
3	Excellent Effort
4	Controlled Load
5	Video
6	Voice
7	Network Control

Type of Service

- Layer 3
- 8bit
- 6bit DSCP
  - Precedence: 0 - 7 (higher is better)
  - D: requests low delay
  - T: requests high throughput
  - R: requests high reliability
- 2bit ECN

0	1	2	3	4	5	6	7
Precedence			D	T	R	ECN	

Type of Service

QoS Values Calculator v3

CoS = Class of Service
DSCP = Differentiated Services Code Point
ToS = Type of Service
AF = Assured Forwarding
IPP = IP Precedence
CS = Class Selector
DP = Drop Probability
ECN = Explicit Congestion Notification

ToS							
DSCP						ECN	
AF (CS;DP)							
IPP=CS			DP				
			Delay	Thruput	Reliability		

	8th bit	7th bit	6th bit	5th bit	4th bit	3rd bit	2nd bit	1st bit
ToS	128	64	32	16	8	4	2	1
DSCP	32	16	8	4	2	1		
CoS=IPP	4	2	1					

Application	CoS=IPP	AF	DSCP	ToS	ToS HEX	DP	8th bit	7th bit	6th bit	5th bit	4th bit	3rd bit	2nd bit	1st bit
Best Effort	0	0	0	0	0		0	0	0	0	0	0	0	0
Scavanger	1	CS1	8	32	20		0	0	1	0	0	0	0	0
Bulk Data	1	AF11	10	40	28	Low	0	0	1	0	1	0	0	0
	1	AF12	12	48	30	Medium	0	0	1	1	0	0	0	0
	1	AF13	14	56	38	High	0	0	1	1	1	0	0	0
Network Mgmt.	2	CS2	16	64	40		0	1	0	0	0	0	0	0
Transaction Data	2	AF21	18	72	48	Low	0	1	0	0	1	0	0	0
	2	AF22	20	80	50	Medium	0	1	0	1	0	0	0	0
	2	AF23	22	88	58	High	0	1	0	1	1	0	0	0
Call Signaling	3	CS3	24	96	60		0	1	1	0	0	0	0	0
Mission-Critical	3	AF31	26	104	68	Low	0	1	1	0	1	0	0	0
Streaming Video	3	AF32	28	112	70	Medium	0	1	1	1	0	0	0	0
	3	AF33	30	120	78	High	0	1	1	1	1	0	0	0
	4	CS4	32	128	80		1	0	0	0	0	0	0	0
Interactive Video	4	AF41	34	136	88	Low	1	0	0	0	1	0	0	0
	4	AF42	36	144	90	Medium	1	0	0	1	0	0	0	0
	4	AF43	38	152	98	High	1	0	0	1	1	0	0	0
	5	CS5	40	160	A0		1	0	1	0	0	0	0	0
Voice	5	EF	46	184	B8		1	0	1	1	1	0	0	0
Routing	6	CS6	48	192	C0		1	1	0	0	0	0	0	0
	7	CS7	56	224	E0		1	1	1	0	0	0	0	0

RSVP

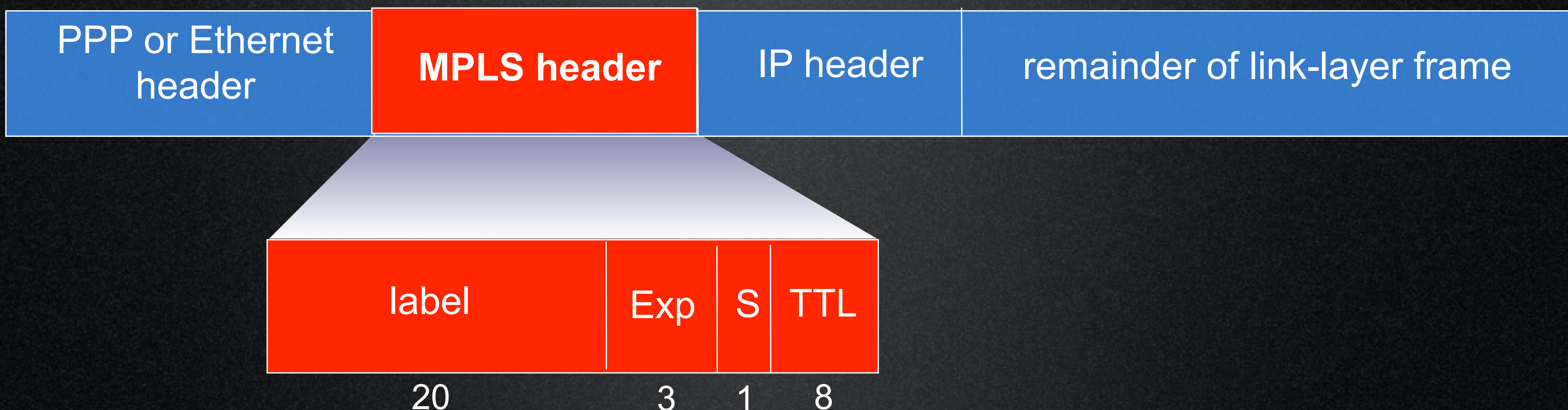
- RSVP was an IntServ implementation
  - Tight relationship though other implementations possible
- it is being used in MPLS nowadays

RSVP

- Used to request a specific QoS from the network
  - simplex (unidirectional) connections
  - routing performed by an underlying protocol (IP), no other assumptions
  - **receiver** initiated reservation
  - soft state
  - designed with multicast group communication in mind

Multi-Protocol Label Switching (MPLS)

- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



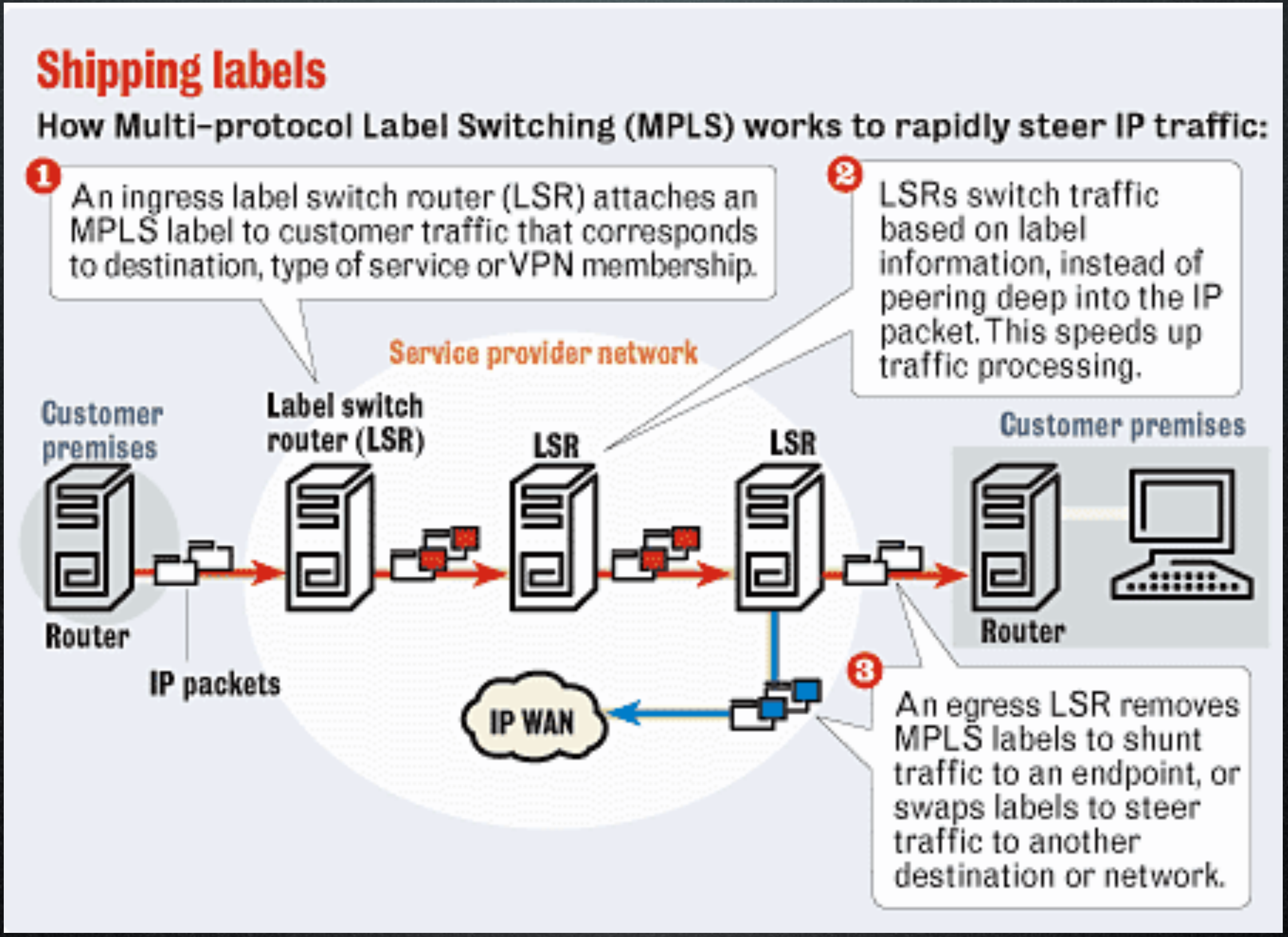
MPLS capable routers

- a.k.a. label-switch router
- forwards packets to outgoing interface based only on label value (does not inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
  - RSVP-TE (Traffic Engineering)
  - forwarding possible along paths that IP alone would not allow (e.g., source-specific routing) !!
  - use MPLS for traffic engineering
- must co-exist with IP-only routers

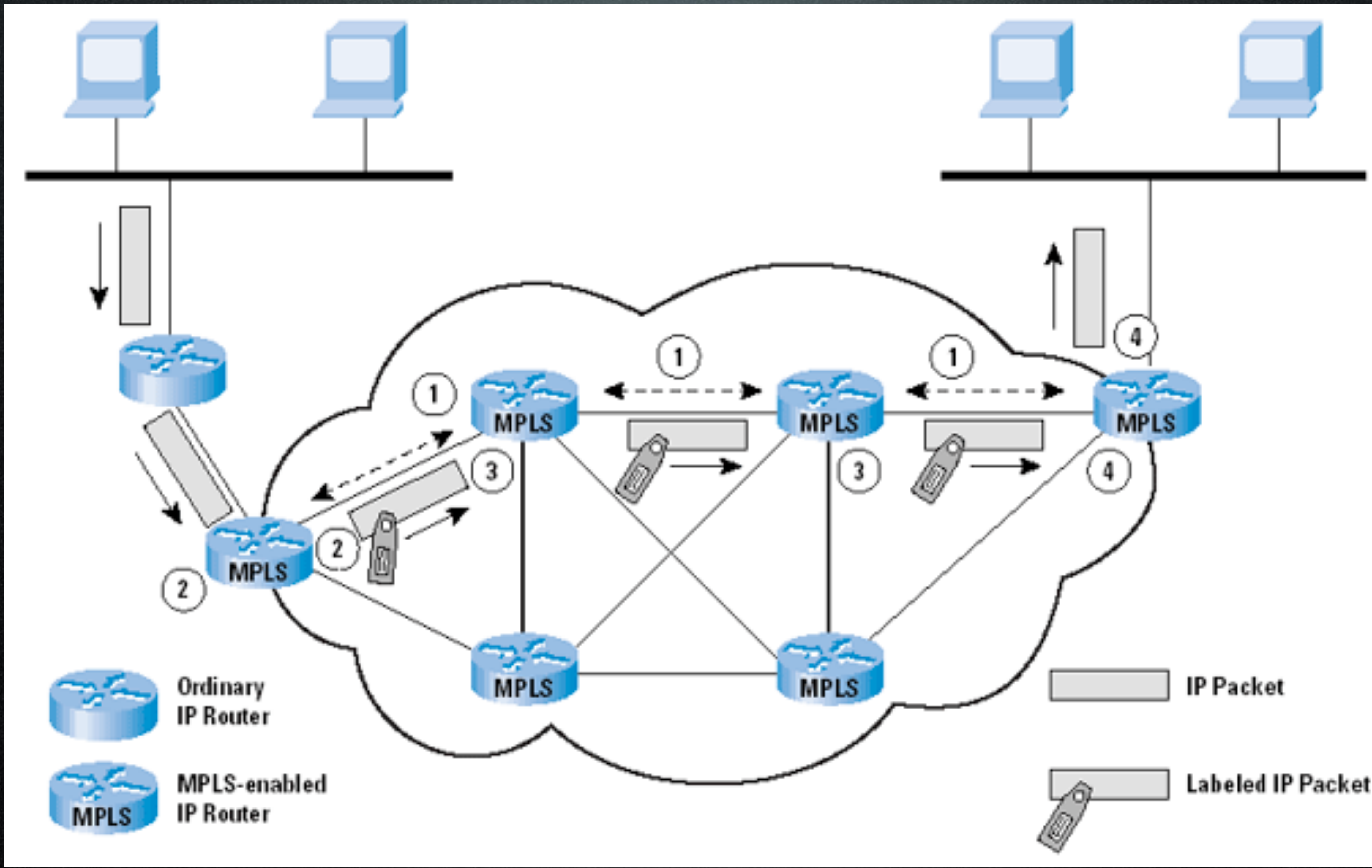
MPLS language

- LER: Label Edge Routers
- LSR: Label Switch Routers
- LDP: Label Distribution Protocol
- LSP: Label Switch Paths
- FEC: Forwarding Equivalence Class
- VRF: Virtual Routing and Forwarding table

Label Switch Routers

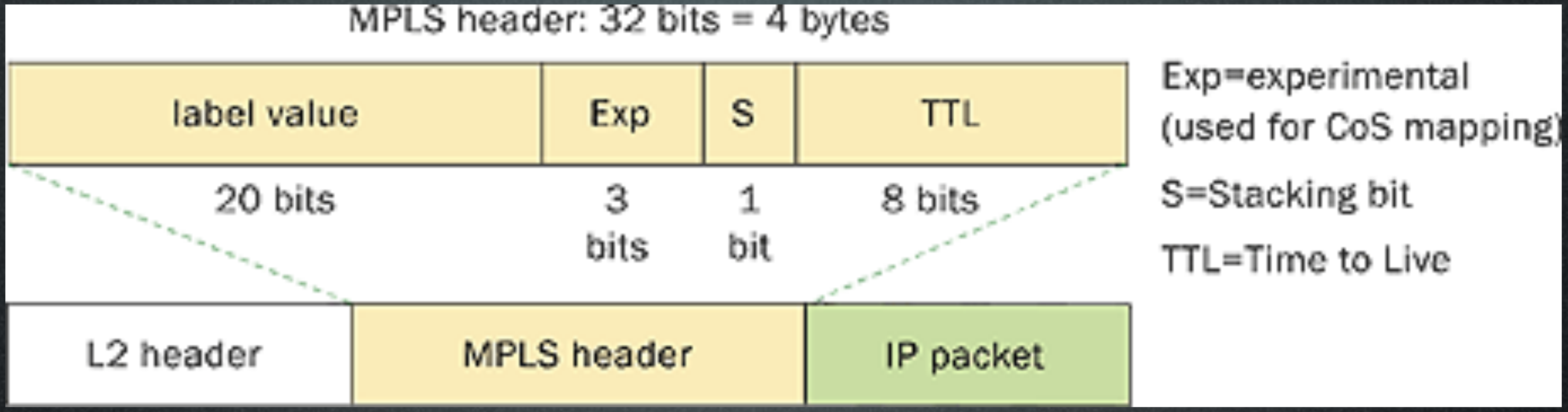


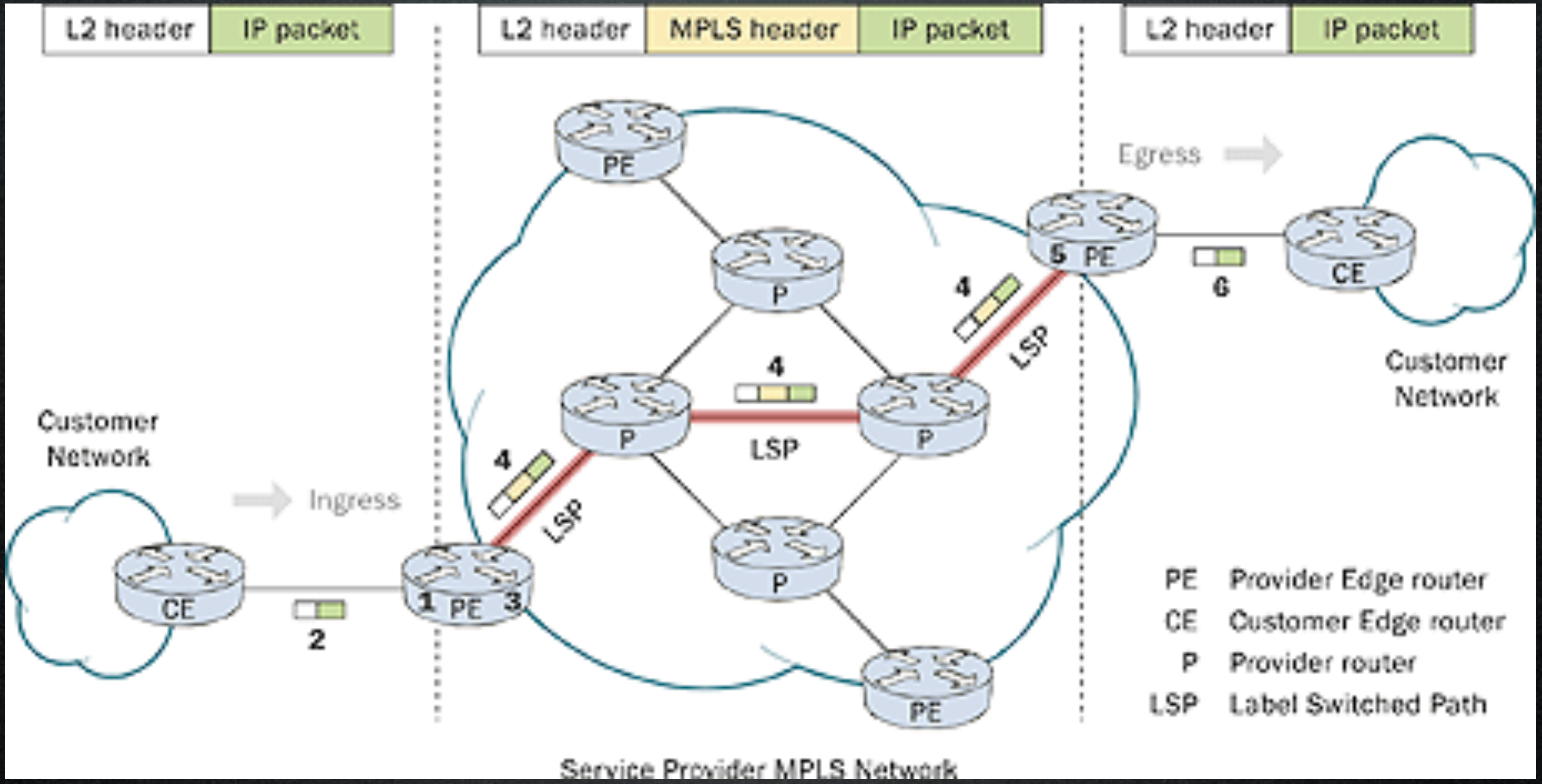
MPLS cloud



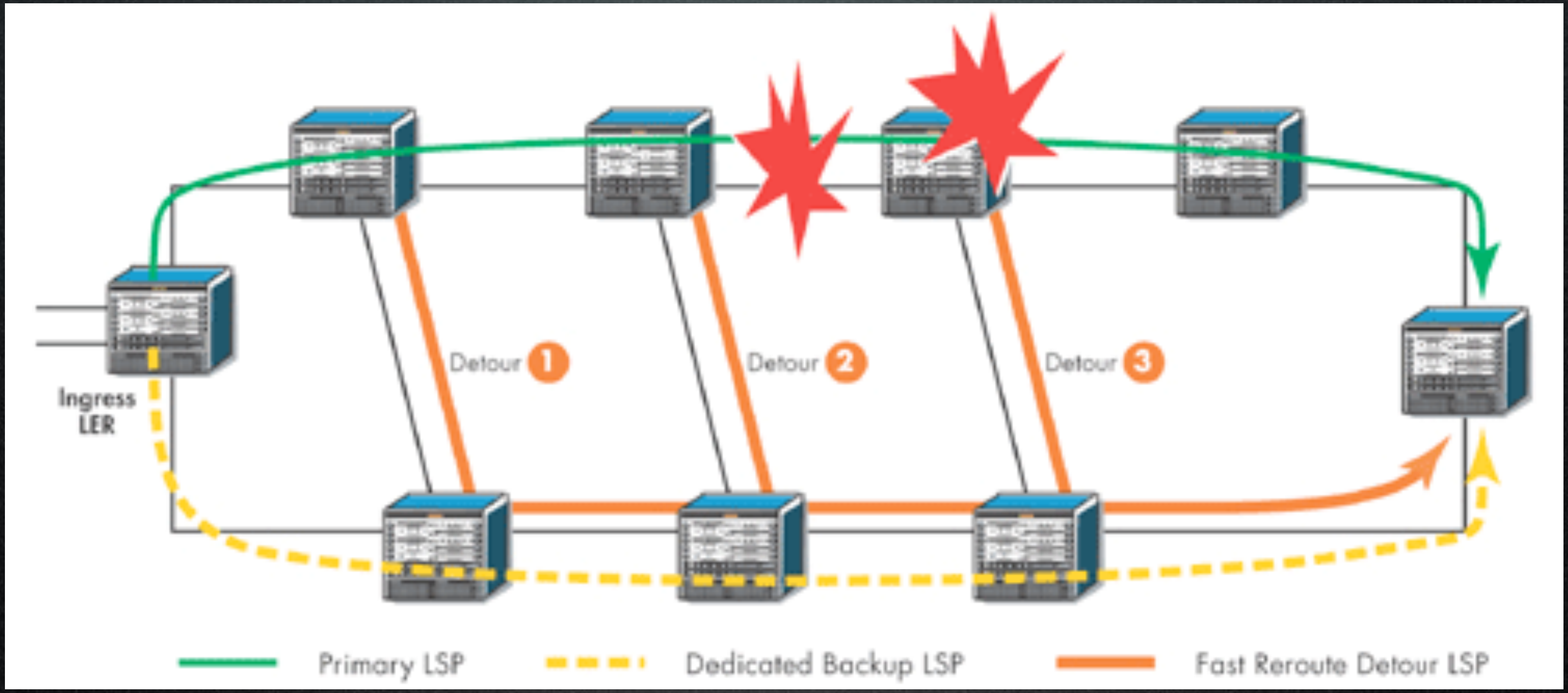
1
1
1
2
2
3
3
4
4
MPLS
MPLS
MPLS
MPLS
MPLS
MPLS
Ordinary
IP Router
MPLS
MPLS-enabled
IP Router
IP Packet
Labeled IP Packet

Label Switched Path

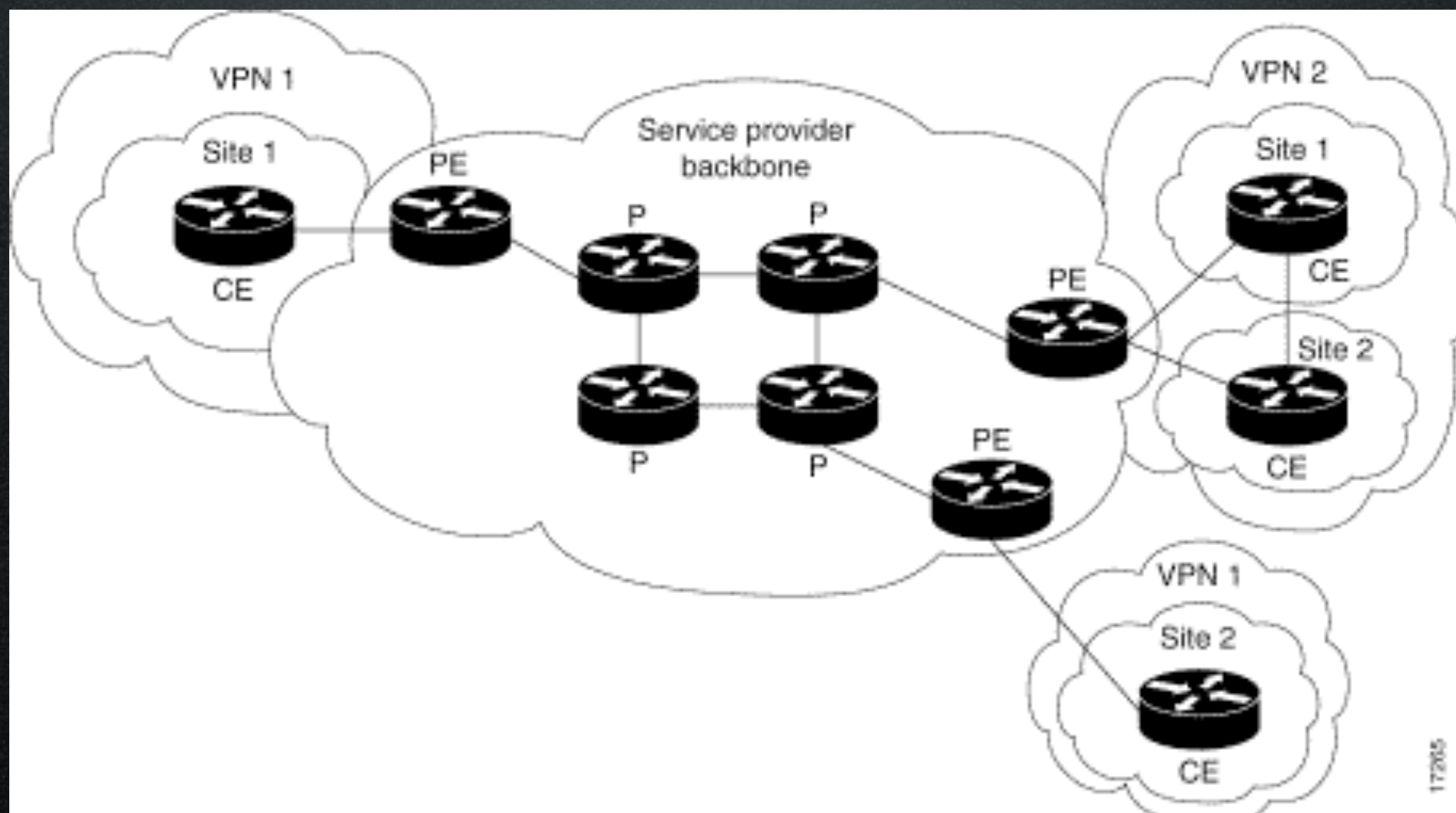




LSP - Fast Reroute

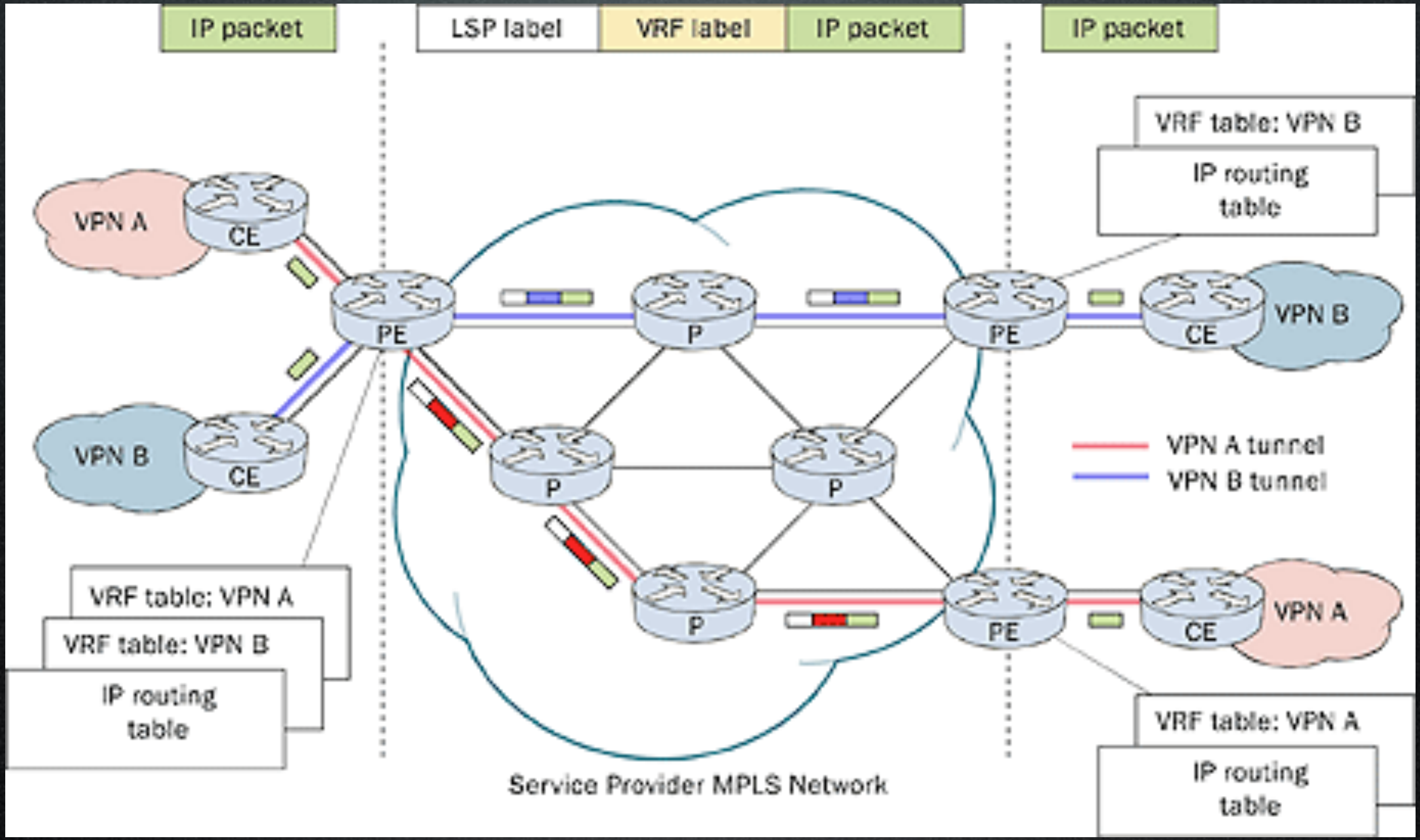


MPLS VPNs

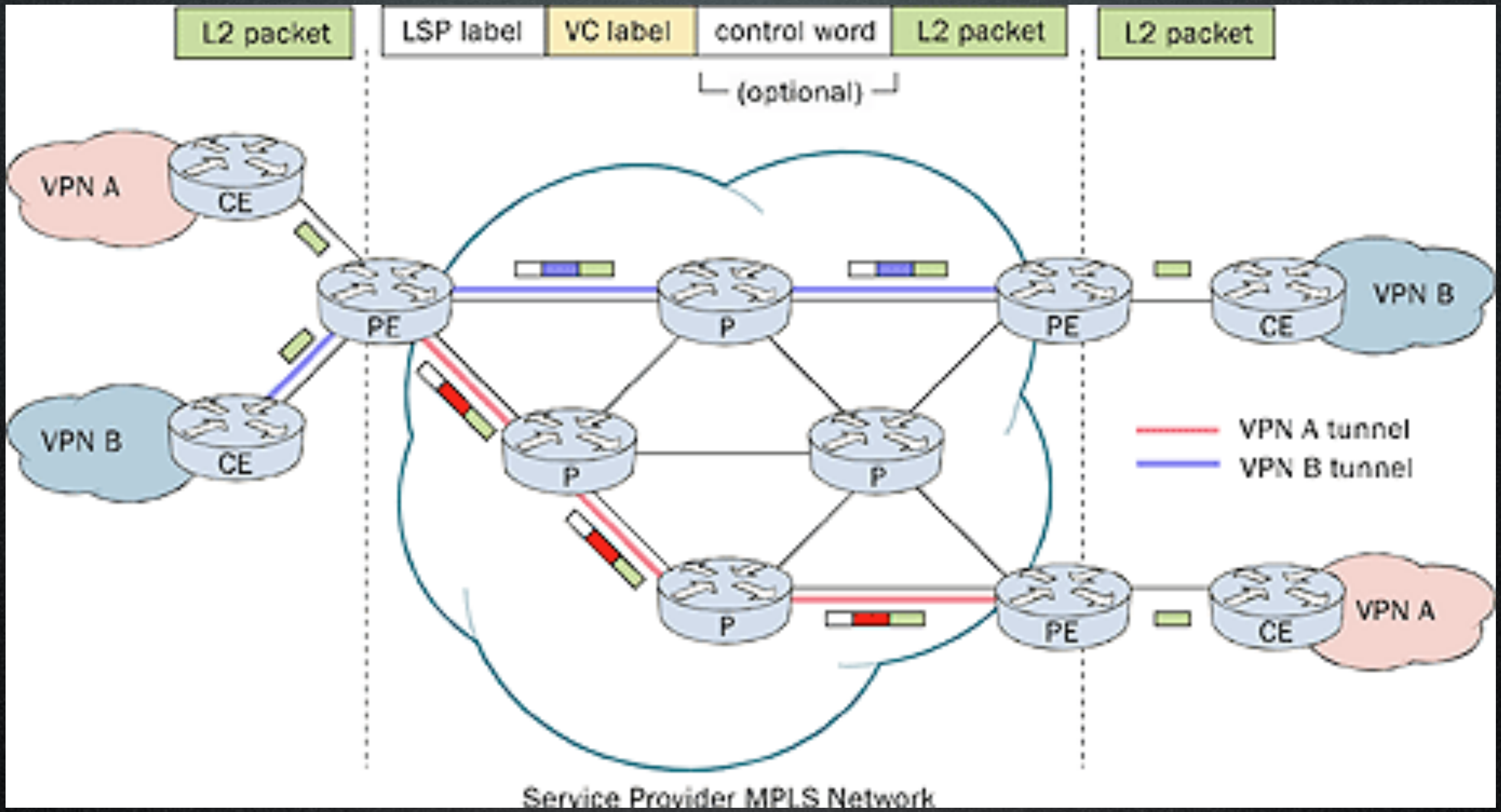


VPN 1
Site 1
CE
PE
Service provider backbone
P
P
P
P
PE
PE
VPN 2
Site 1
CE
Site 2
CE
VPN 1
Site 2
CE
17265

L3 VPNs

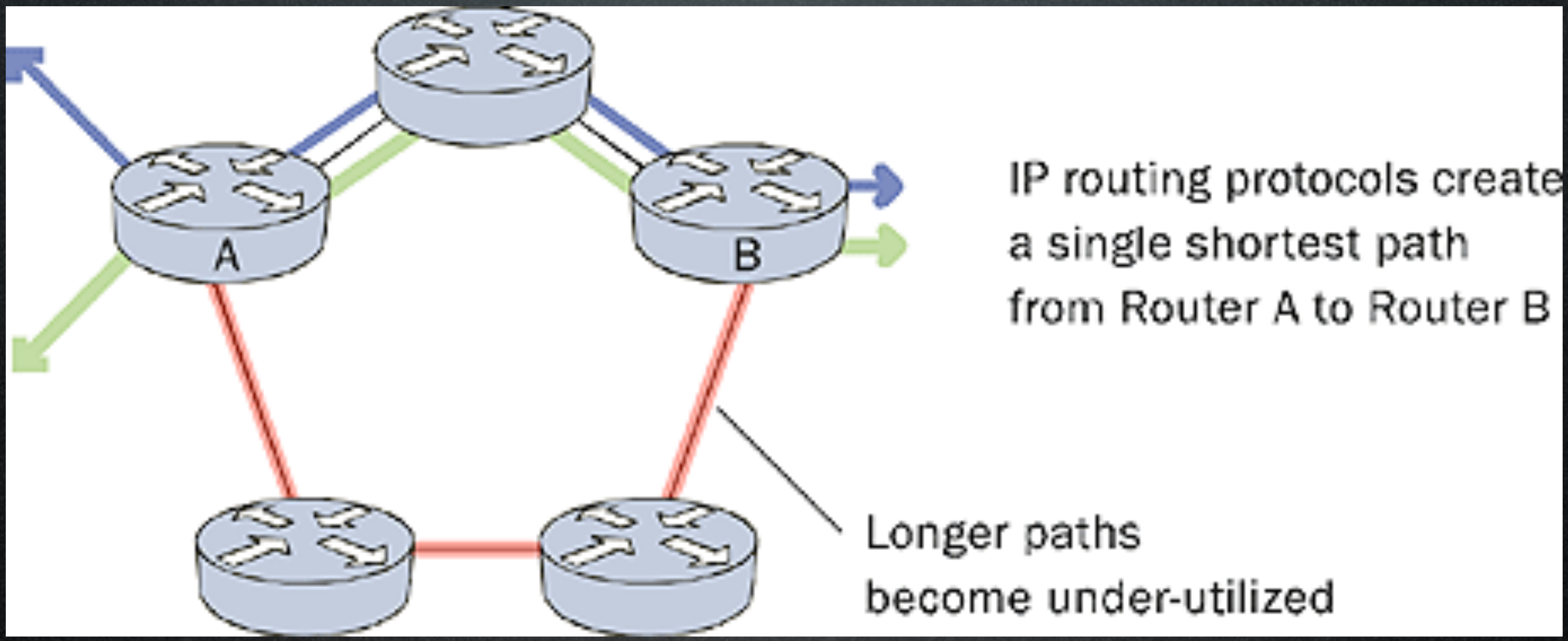

IP packet
LSP label
VRF label
IP packet
IP packet
VRF table: VPN B
IP routing table
VPN A
CE
PE
P
P
PE
CE
VPN B
VPN B
CE
P
P
VPN A tunnel
VPN B tunnel
P
PE
CE
VPN A
VRF table: VPN A
VRF table: VPN B
IP routing table
VRF table: VPN A
IP routing table
Service Provider MPLS Network

L2 VPNs

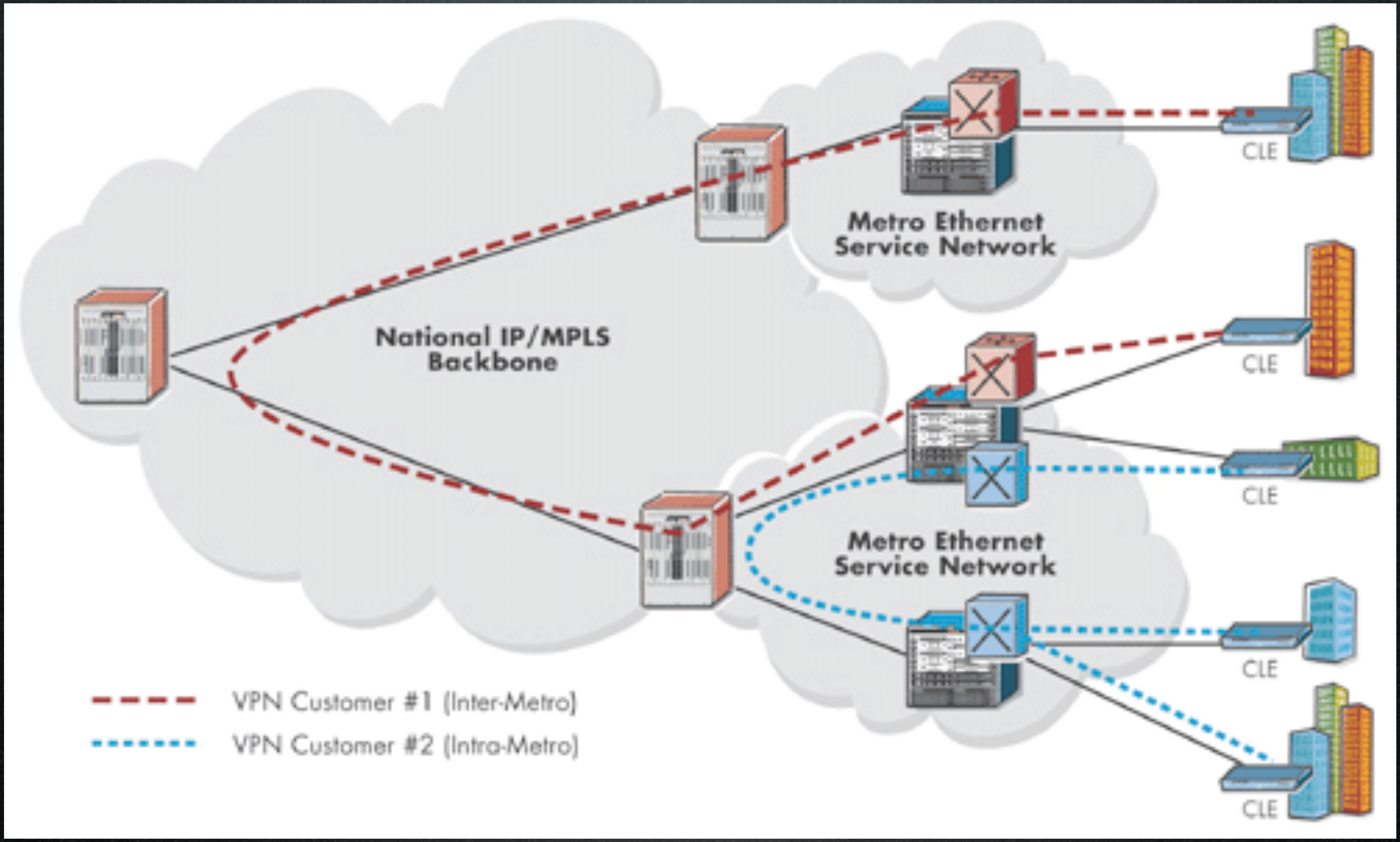


L2 packet
LSP label
VC label
control word
L2 packet
(optional)
L2 packet
VPN A
CE
PE
P
PE
CE
VPN B
VPN B
CE
P
P
VPN A tunnel
VPN B tunnel
P
PE
CE
VPN A
Service Provider MPLS Network

Traffic Engineering

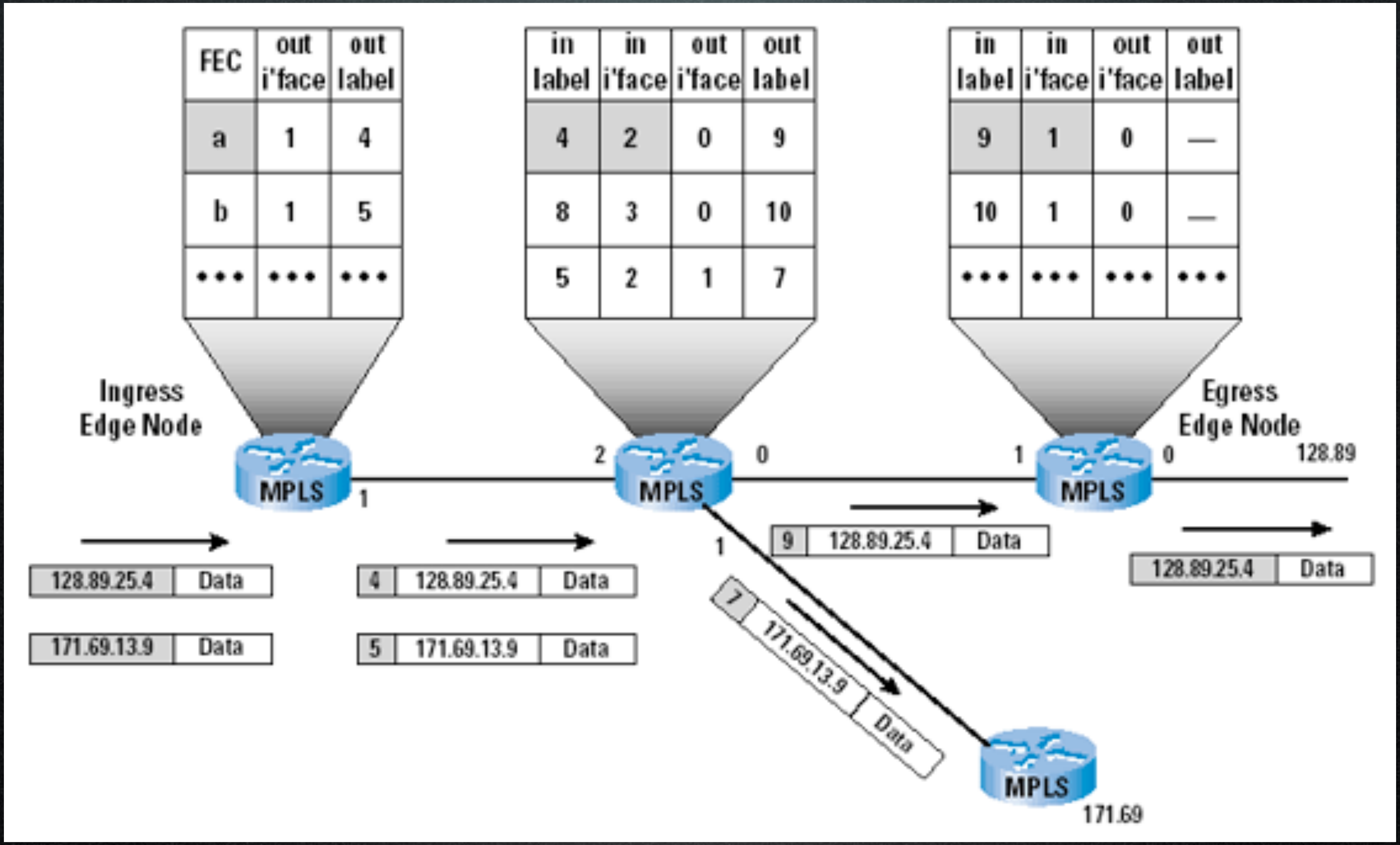


Example

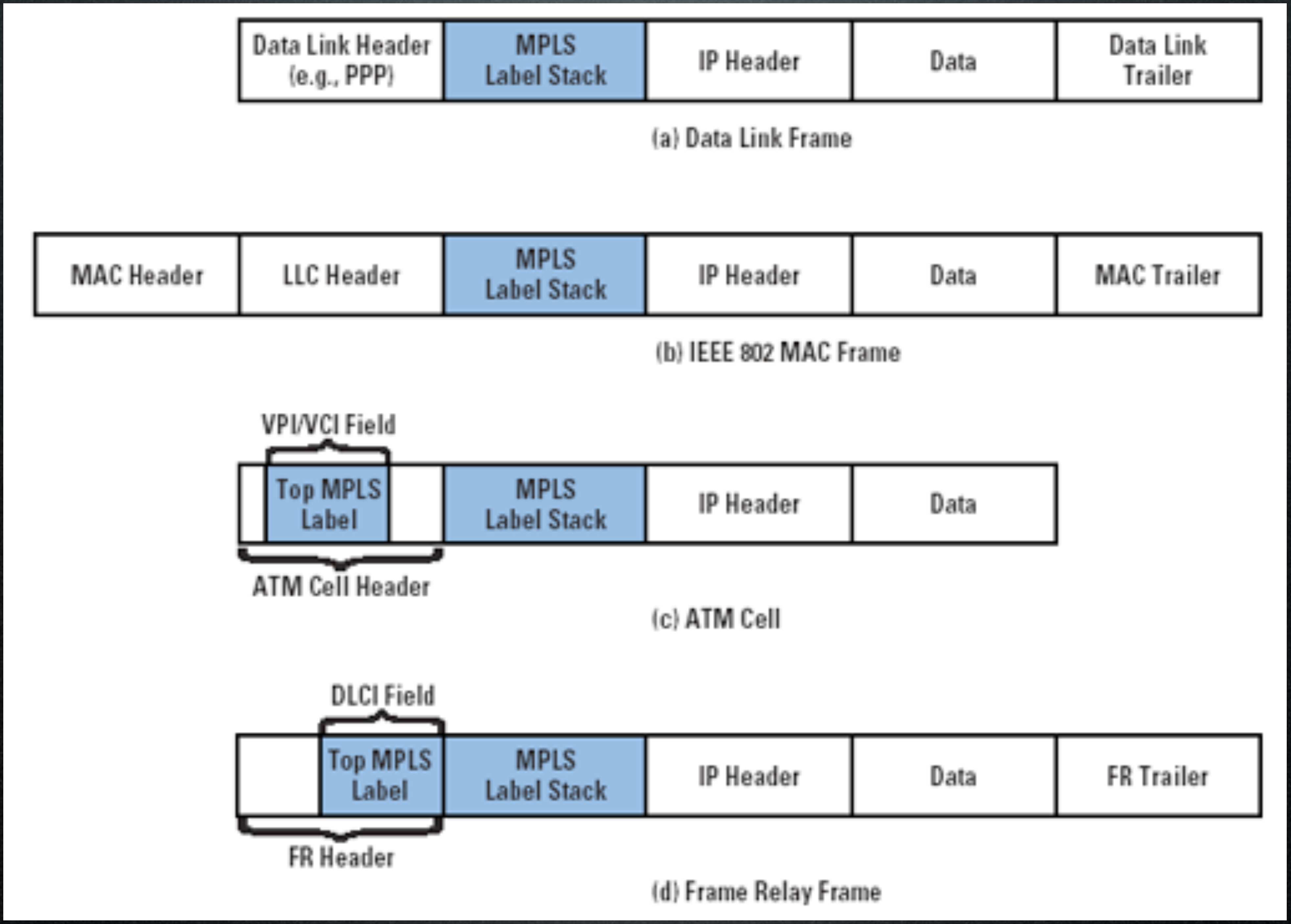


Metro Ethernet
Service Network
CLE
National IP/MPLS
Backbone
CLE
CLE
Metro Ethernet
Service Network
CLE
CLE
VPN Customer #1 (Inter-Metro)
VPN Customer #2 (Intra-Metro)

MPLS packet forwarding



MPLS over X


Data Link Header (e.g., PPP)
MPLS Label Stack
IP Header
Data
Data Link Trailer
(a) Data Link Frame
MAC Header
LLC Header
MPLS Label Stack
IP Header
Data
MAC Trailer
(b) IEEE 802 MAC Frame
VPI/VCI Field
Top MPLS Label
MPLS Label Stack
IP Header
Data
ATM Cell Header
(c) ATM Cell
DLCI Field
Top MPLS Label
MPLS Label Stack
IP Header
Data
FR Trailer
FR Header
(d) Frame Relay Frame