# MPLS and DiffServ

Sources:
MPLS Forum, Cisco
V. Alwayn, *Advanced MPLS Design and Implementation*, Cisco Press
E. W. Gray, *MPLS Implementing the Technology*, Addison Wesley
B. Davie and Y. Rekhter, *MPLS Technology and Applications*, Morgan Kaufmann
E. Osborne and A. Simha, *Traffic Engineering with MPLS*, CiscoPress

# Evolution of QoS Standards

- Best Effort Service: 1981
- Integrated Services (IntServ): 1997
- Differentiated Services (DiffServ): 1998
- DiffServ-Aware TE (DS-TE)

# What is IntServ ?

- An architecture allowing the delivery of the required level of QoS to **real-time applications**
- Introduces a **circuit-switched model** to IP
- A signalling-based system where the endsystem has to request the required service-level
- RSVP – one of the signaling protocols of choice
- A way of providing **end-to-end QoS**, state maintenance (for each RSVP flow and reservation), and admission control at each NE

# IntServ Characteristics

- Introduces the model of **connections** or flows
- Defines a traffic specification called **Tspec**, which specifies the kind of application traffic that ingresses the network.
- IntServ also defines a reservation spec called **Rspec**, which requests specific QoS levels and ther reservation of resources.
- Requires the following to verify that traffic conform to its Tspec:
  - ✓ Known QoS requirements
  - ✓ Signalling protocol (i.e., **RSVP**)
  - ✓ Significant enhancements on network element:
    - *Admission control*
    - *Policy control*
    - *Packet classification and marking*
    - *Packet scheduling and queuing*
    - *Packet dropping policy*

# IP Precedence

- Main problem with IntServ:
  - The IntServ RSVP **per-flow** approach to QoS is **not scalable** and adds complexity to implementation.
- Solution?:
  - IP precedence simplifies it by adopting an **aggregate model for flows** by classifying various flows into aggregated **classes** and providing the appropriate QoS for the classified flows.

# Differentiated Services (DiffServ)

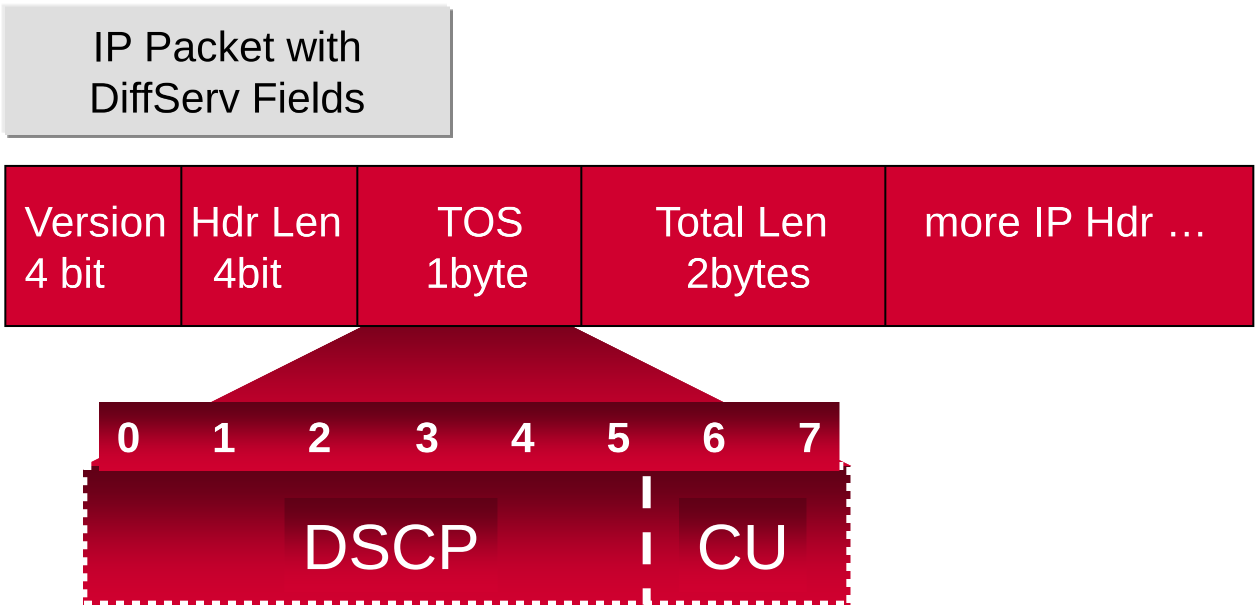
# What is DiffServ

- An architecture for implementing scalable, stateless service differentiation
- A service defines significant characteristics of packet transmission in one direction across a set of one or more paths in the network
- Examples of characteristics:
  - ✓ Delay
  - ✓ Jitter
  - ✓ Packet loss

# DiffServ Service Classes or Per Hop Behaviors (PHB)

- Describes the forwarding behavior applied to an **aggregate of flows**
- The means a network-node allocates resources to meet a behavior aggregate
- Per Hop Behaviors are implemented (on each router) via:
  - ✓ Queue management and scheduling
    - ➢ Buffer size, Queue depth, Over-subscription policy
  - ✓ Scheduling
    - ➢ Scheme to determine which queue to service when link is available
  - ✓ Congestion management and avoidance
    - ➢ Optimize resource utilization

# DiffServ Service Classes



IP Packet with DiffServ Fields

Version 4 bit | Hdr Len 4bit | TOS 1byte | Total Len 2bytes | more IP Hdr …

0 1 2 3 4 5 6 7

DSCP | CU

DiffServ Field (DSCP) defines Per-Hop Behavior (PHB) (i.e., marking)

*The remaining two unused bits in the TOS byte are used for TCP ECN which is defined in RFC3168.*

# DiffServ Service Classes

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | unused | |

Best Effort DSCP

- ➢The common best effort forwarding behavior available in all routers
- ➢Network will deliver these packets whenever resources available
- ➢Node should make sure that these packets don’t get ‘starved’
- ➢Packets with an unidentified DSCP should also receives this PHB

# DiffServ Service Classes

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Class | | | Drop Precedence | | | unused | |

Assured Forward (AF) DSCP

- Class – specifies the PHB that packet is to receive. AF is a method of providing low packet loss, but it makes minimal guarantees about latency.
    - AF1 – 001
    - AF2 – 010
    - AF3 – 011
    - AF4 – 100
- Drop Precedence - marks relative importance of a packet within a given class.
    - 010 low
    - 100 medium
    - 110 high

# DiffServ Service Classes

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 | 0 | unused | |

Expedited Forward (EF) DSCP

- These packets must be policed at ingress
- Non conforming packets are discarded
- These packets must be shaped on egress
- These packets should receive Priority Queuing or LLQ (Premium Service PHB)

# TOS and DSCP

- Issue on backward compatibility
- TOS octet and IP precentence were not widely used.
- IETF decided to reuse TOS as the DSCP for DiffServ networks:
  - ✓ IP Precedence is still used
  - ✓ DiffServ also defines the Class Selector
  - ✓

| DSCP | Binary | Decimal |
|---|---|---|
| CS0 | 000 000 | 0 |
| CS1 | 001 000 | 8 |
| CS2 | 010 000 | 16 |
| CS3 | 010 000 | 24 |
| CS4 | 010 000 | 32 |
| CS5 | 010 000 | 40 |
| CS6 | 010 000 | 48 |
| CS7 | 010 000 | 56 |

# DiffServ Service Classes Summary

| | | |
|---|---|---|
| Best Effort DSCP | Bronze Service | •Best Effort Service<br>•Client gets available Resources only |
| Assured Forward (AF) DSCP | Silver Service | •Specified Forwarding Behavior<br>•Specified Drop Precedence |
| Expedited Forward (EF) DSCP | Gold Service | •Priority Delivery<br>•Must adhere to “traffic contract” |

# DiffServ Characteristics

- DiffServ is a relatively simple and coarse method to provide differentiated Classes of Service.
- Offers a small well defined set of building blocks from which several services may be built.
- Flows (stream of packets with a common observable characteristics) are *conditioned* at the network ingress and receive a certain forwarding treatment *per hop behavior* within the network.
- Multiple queuing mechanisms offer differentiated forwarding treatments.

# DiffServ Summary

- Model consists of a set of Differentiated Services Domains (Policy / Management Domain)
- Interconnections of DS Domains require Traffic Classification and Conditioning
- DiffServ deals with aggregates of flows assigned to a PHB
- DiffServ operates stateless and does not require signalling
- DiffServ is a refined CoS mechanism

# MPLS and DiffServ

# MPLS Support of DiffServ

- **Backward compatibility**: Because MPLS is there primarily to transport IP, MPLS's primary QoS goal is to support existing IP QoS models
- **Scalability**: Because MPLS is there to support very large scale operations, MPLS should also be capable of supporting DiffServ.
- What Issues to consider?
  - ✓ Need to ensure that packets marked with various DiffServ Code Points (DSCPs) receive the appropriate QoS treatment at each LSR
    - ➢ DSCP is carried in IP header, but LSRs do not check IP header when forwarding packets
    - ➢ Hence, need some way to determine the appropriate PHB from the label header.
      - Exp bits in the shim header
      - ATM cell header

# Exp Bits

- The Exp field in the shim header
  - ✓ Original intent was to support marking of packets for DiffServ.
  - ✓ But only 3 bits (up to 8 values), DiffServ field is 6 bits (up to 64 DSCPs)
  - ✓ How to do the mapping between the two?

# Exp and DSCP Mapping

- How to map Exp and DSCPs?
  - ✓ If <= 8 PHBs, Exp field is sufficient. A LSR can maintain a mapping from Exp values to PHBs.
    - ➢ LSRs work similarly to conventional router.
    - ➢ Configure every LSR: Exp -> PHB mapping is configured on every router as per Diffserv
- Signaling?
  - ✓ Same as before, LDP, RSVP
- The **label** tells an LSR **where** to forward a packet, and the **Exp** bits tell it **what** **PHB** to treat the packet with.
- An LSP set up this way is called an **E-LSP**; E stands for Exp, meaning that the **PHB is inferred from the Exp bits.**

# Exp and DSCP Mapping

- If more than 8 PHBs?
  - ✓ Exp along is not enough.
  - ✓ Solution: use **label** to convey the PHB.
  - ✓ In this case, the LSP is called **L-LSP**; L stands for label, meaning the **PHB is inferred from the label**.
- If shim header is not used, such as ATM?
  - ✓ No Exp field
  - ✓ Again, the label field will be used in this case
  - ✓ **But, L-LSPs require signaling extension**

# Enhancement of Label Distribution/Signaling

- Why enhancement?
  - ✓ Because we want to convey information about the PHBs inside labels
- Label distribution mechanisms are used to advertise bindings between labels and FECs such as address prefixes
- Now need to expand the binding to both an FEC and a PHB (or PHBs)
- New DiffServ object/TLV added to RSVP/LDP to signal the “queue” in which to enqueue the label
- Meaning of Exp bits is well-known (i.e. standardised for each PSC (PHB Scheduling Class))
- <draft-ietf-mpls-diff-ext-03.txt>, by Francious Le Faucheur, et al

# Label Request Message

| Label Request | Message Length |
|---|---|
| Message ID | |
| LSPID TLV | |
| Explicit Route  TLV (optional) | |
| Traffic Parameters TLV (optional) | |
| Pinning TLV (optional) | |
| Resource Class TLV (optional) | |
| Pre-emption TLV (optional) | |
| **Diff-Serv TLV (optional)** | |

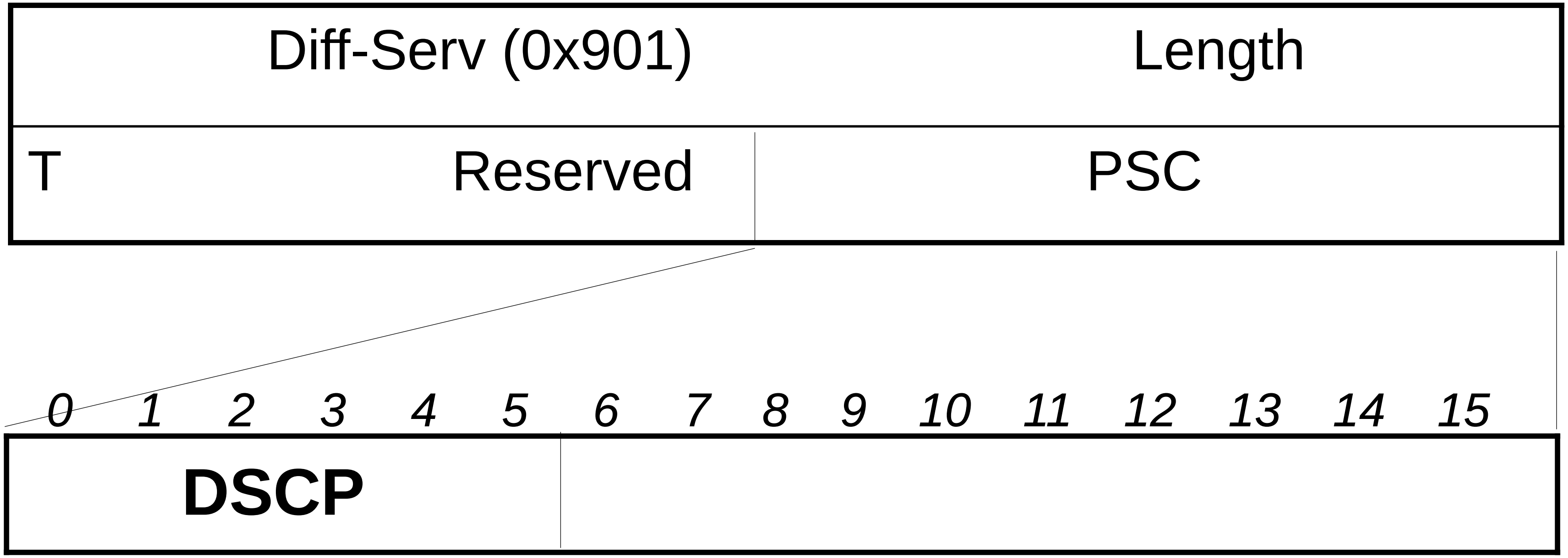
# DiffServ TLV for E-LSP CR-LDP

| | | |
|---|---|---|
| Diff-Serv (0x901) | | Length |
| T | Reserved | Mapnb(4) |
| | Map 1 | |
| | . | |
| | Mapnb | |

*Map Entry Format*

| | | |
|---|---|---|
| Reserved (13) | EXP (3) | PHBID (16) |

# DiffServ TLV for L-LSP CR-LDP



| Diff-Serv (0x901) | Length |
|---|---|
| T Reserved | PSC |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

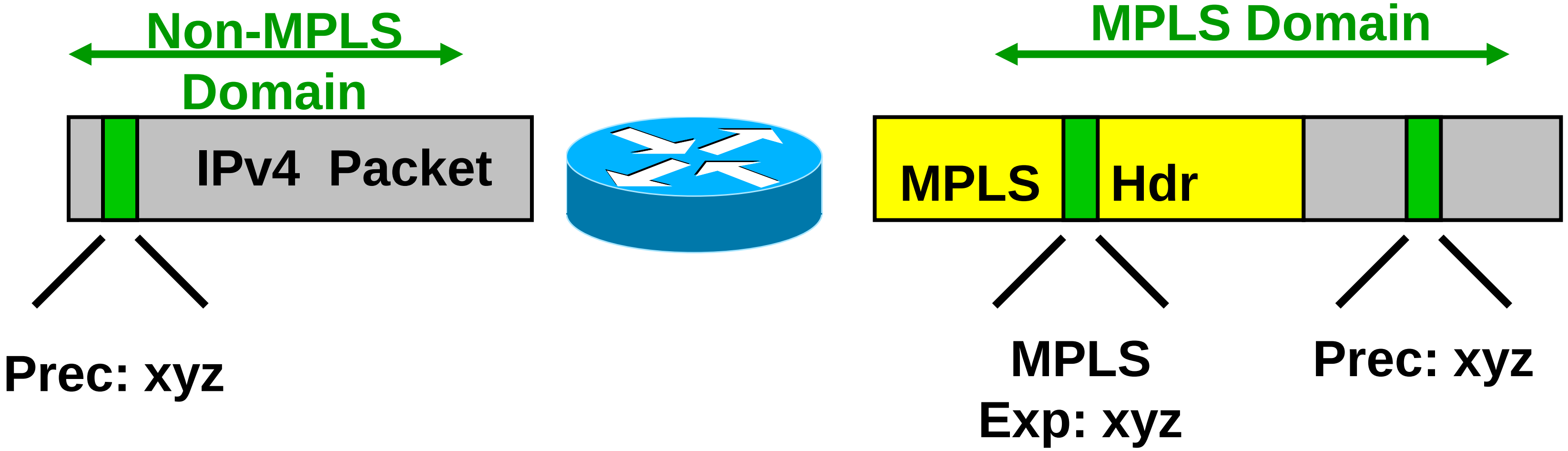
**DSCP**

PSC:PHB Scheduling Class

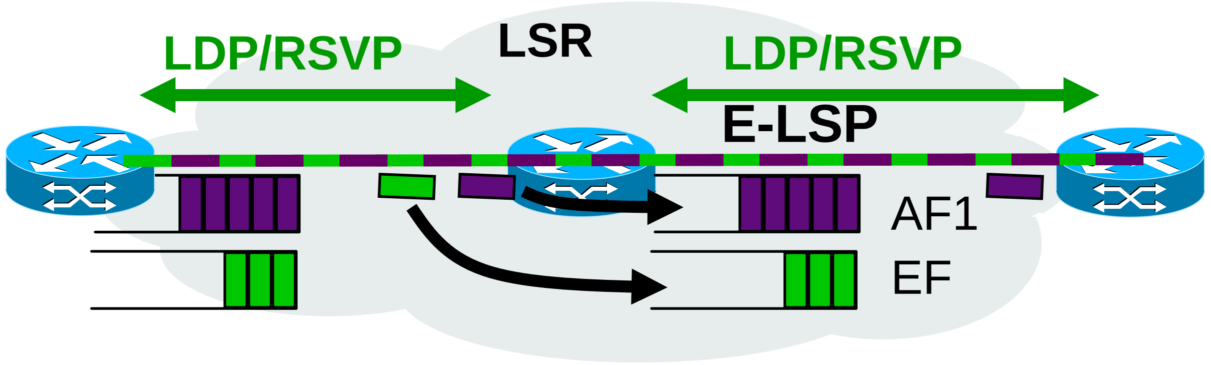
# MPLS QoS

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                Label                  | EXP |S|       TTL     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Copy of IP Precedence into MPLS EXP
- Each LSR along the LSP maps the Exp bits to a PHB
- Mapping of IP Precedence into MPLS Exp
- Also can use a different value in Exp field



# MPLS QoS
# E-LSP Example

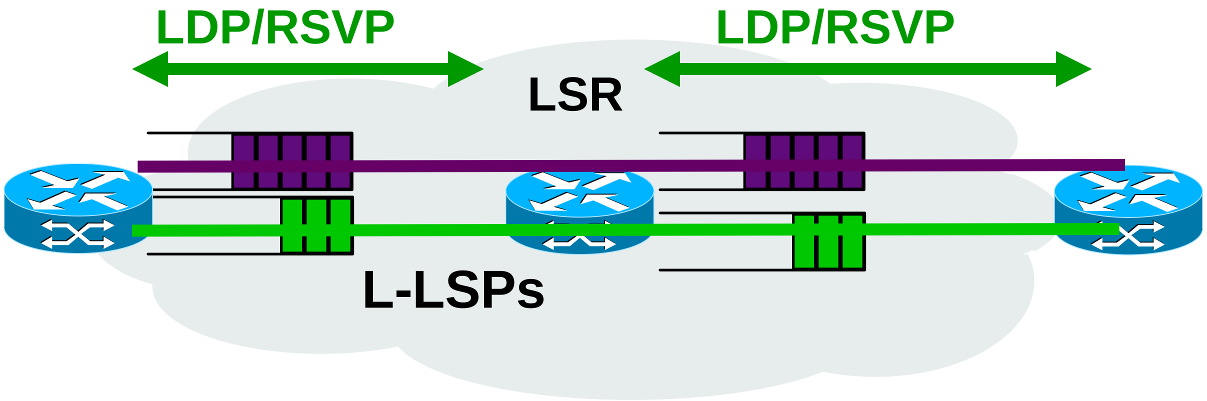


- E-LSPs can be established by various label binding protocols (LDP or RSVP)
- Example above illustrates support of EF and AF1 on single E-LSP

  Note: EF and AF1 packets travel on **single LSP** (single label) but are enqueued in **different queues** (different Exp values)

- Queue is selected based on **Exp**

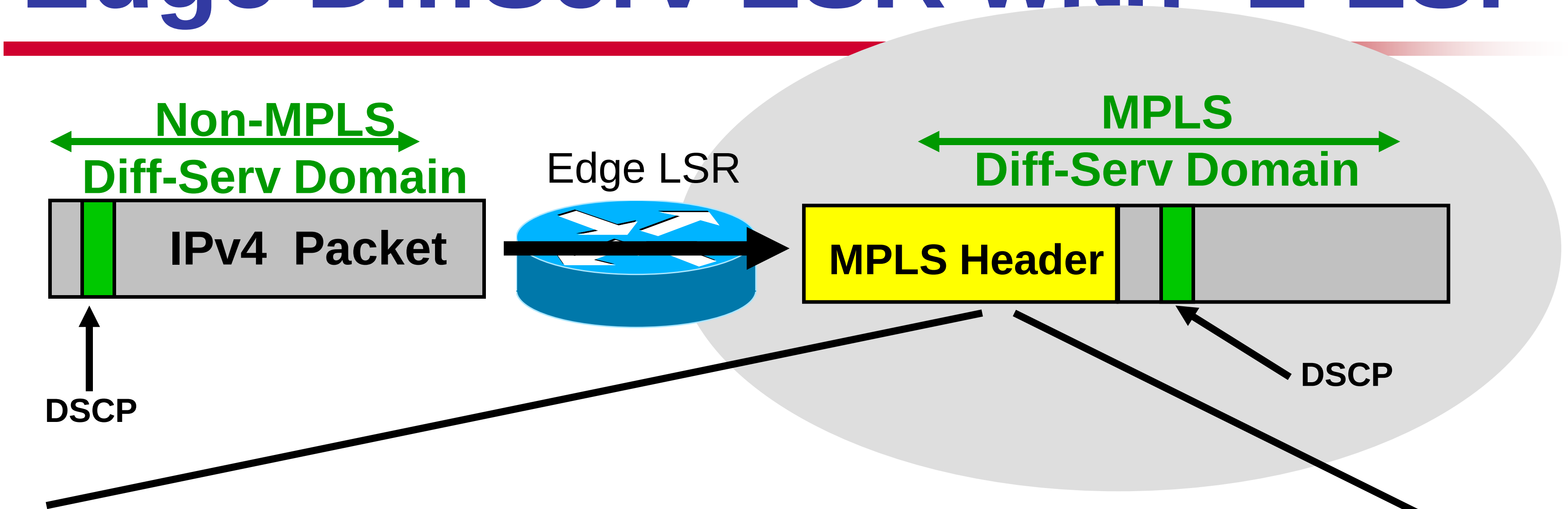
# MPLS QoS
# L-LSP Example



- L-LSPs can be established by various label binding protocols (LDP or RSVP)
- Only one PHB per L-LSP is possible, except for DiffServ AF.
- For DiffServ AF, packets sharing a common PHB can be aggregated into a FEC, which can be assigned to an LSP. This is known as a *PHB scheduling class*.
- Example above illustrates support of EF and AF1 **on separate L-LSPs**

  EF and AF1 packets travel on **separate LSPs** and are enqueued in **different queues (different label values)**
- Queue is selected based on **label**, drop precedence is based on **Exp**

# MPLS QoS
# Edge DiffServ LSR with  L-LSP



```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Label                   | EXP |S|      TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

1) identify incoming packet’s BA looking at incoming DSCP
2) pick the LSP/label which supports the right FEC and the right BA
3) mark the EXP field to reflect the packet’s BA (optional)

# Comparison of E-LSPs & L-LSPs

| E-LSPs | L-LSPs |
|---|---|
| PHB is determined from Exp | PHB is determined from label or from label plus Exp/CLP bits |
| No additional signaling required | PHB or PHB scheduling class is signaled at LSP setup |
| Exp & PHB mapping is configured | Label and PHB mapping is signaled<br>Exp/CLP and PHB mapping is standardised (only for AF) |
| Shim header required; not possible for ATM | Shim or link layer header may be used; L-LSPs are suitable for ATM links |
| Up to 8 PHBs per LSP | One PHB per LSP except for AF |

Advantages for E-LSPs:

Advantages for L-LSPs:

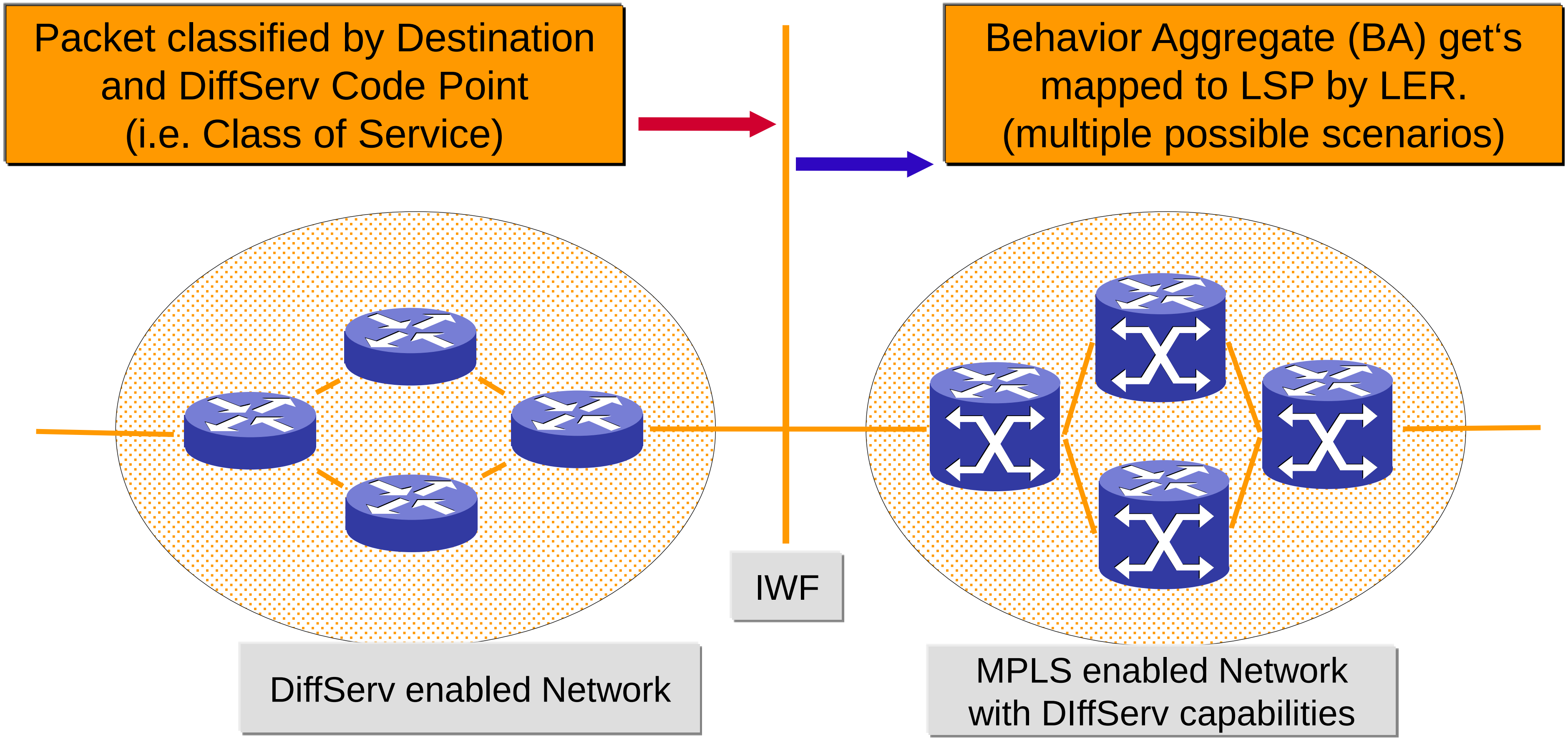
# MPLS QoS
# E-LSP & L-LSP Applicability

- MPLS over PPP and LAN:

  both E-LSPs and L-LSPs are applicable

- MPLS over ATM
  - only L-LSPs possible (Exp is not seen by ATM LSR)
  - PHB is inferred from the label carried in the VCI field
  - The label-to-PHB is signaled

# MPLS – DiffServ Interworking



Packet classified by Destination and DiffServ Code Point (i.e. Class of Service)

Behavior Aggregate (BA) get‘s mapped to LSP by LER. (multiple possible scenarios)

IWF

DiffServ enabled Network

MPLS enabled Network with DIffServ capabilities

# Label Stack Management

- Exp bits and IP Precedence bits or the DSCP bits mapping could involve three different cases:
  - ✓ ip-to-mpls or ip2mpls
  - ✓ mpls2mpls (label stack)
  - ✓ Mpls2ip
- Example
- But the Exp value or values (for mpls2mpls) and DSCP values could be different. How to treat the packet?
  - ✓ IP Precedence or MPLS Exp?
  - ✓ According to the label that was removed?
  - ✓ According to the outmost indicator in whatever remains after the POP?

# Tunnel Modes

- RFC 3270 defines three tunnel modes (still developing technology?):
  - ✓ **Uniform**
    - The network is **a single DiffServ domain**
    - Changes to the Exp values in transit are to be **propagated** to all labels underneath the packet and the underlying IP packet.
  - ✓ **Short-Pipe**
    - Useful for ISPs implementing their own QoS policy independent of their customer's QoS policy.
    - Change is propagated downward only within the label stack, not to the IP packet.
    - In the mpls2ip pop case, the PHB is decided based on the DSCP on the IP packet.
  - ✓ **Pipe**
    - The PHB on the mpls2ip link (for pop case) is selected based on the **removed Exp value**, not the DSCP value in IP packet.
    - The DSCP value in IP is not changed, but the mpls2ip path does not consider the DSCP for queuing on the egress link.

# DiffServ-Aware Traffic Engineering

# (DS-TE)

# DiffServ-Aware TE (DS-TE)

- DiffServ with MPLS packets is conceptually the same thing as with IP packets.
  - ✓ EXP setting vs. IP Precedence setting
- Why MPLS in the original motivation?
  - ✓ Make a headend resource-aware, so that it can intelligently pick paths through the network for its traffic to take.
  - ✓ **TE**: steer IP traffic way from the IGP shortest path or congested links.
- **Steering traffic per QoS**?
  - ✓ If there is traffic destined for a router, all that traffic follows the same path (per-src-dest), regardless of the DSCP/EXP settings.
  - ✓ Routing is limited by the routing table and how it decides to forward traffic.
  - ✓ So far (based on what we have covered),TE data forwarding doesn’t do admission control on a per-QoS class basis.

# DS-TE

- What's the problem from TE perspective?
  - ✓ If there is a congested link at a downstream node along the forwarding path, the congestion knowledge is localized at the downstream node and is not propagated back to the edge devices that send traffic down that path.
    - ➢ Gold traffic might be dropped.
  - ✓ Edges continue to send traffic to the same downstream router.
    - ➢ Gold traffic might continue to be dropped.
- Need **per-class admission control**.
- Combine DiffServ and TE (DS-TE).

# DS-TE (more)

- TE offers call admission control in addition to the PHB offered by DiffServ.
  - ✓ If more traffic is sent down a certain path than there is available bandwidth, queue higher-priority traffic ahead of low-priority traffic.
- How about the possible contention between different high-priority traffic streams?
  - ✓ Two voice pipes from customers, both with a low-latency requirement, if you forward both streams down the same congested paths, both streams might be affected.
- DS-TE allows to advertise more than one pool of available resources for a given link – a **global pool** and **subpools**.

# Subpools

- A subpool is a **subset of link bandwidth** that is available for a specific purpose.
  - ✓ A pool with which you can advertise resources for a separate queue.
  - ✓ Currently, DS-TE allows to advertise one subpool.
  - ✓ Recommended for low-latency queue (LLQ)
  - ✓ The actual queuing behavior at every hop is still controlled by the regular DiffServ mechanisms such as LLQ.
  - ✓ DS-TE has the ability to reserve **queue bandwidth**, rather than just link bandwidth in the control plane.
  - ✓ Let you build TE-LSPs that specifically reserve subpool bandwidth and carry only the specified traffic (e.g. LLQ).

# How to Make Use of Subpool?

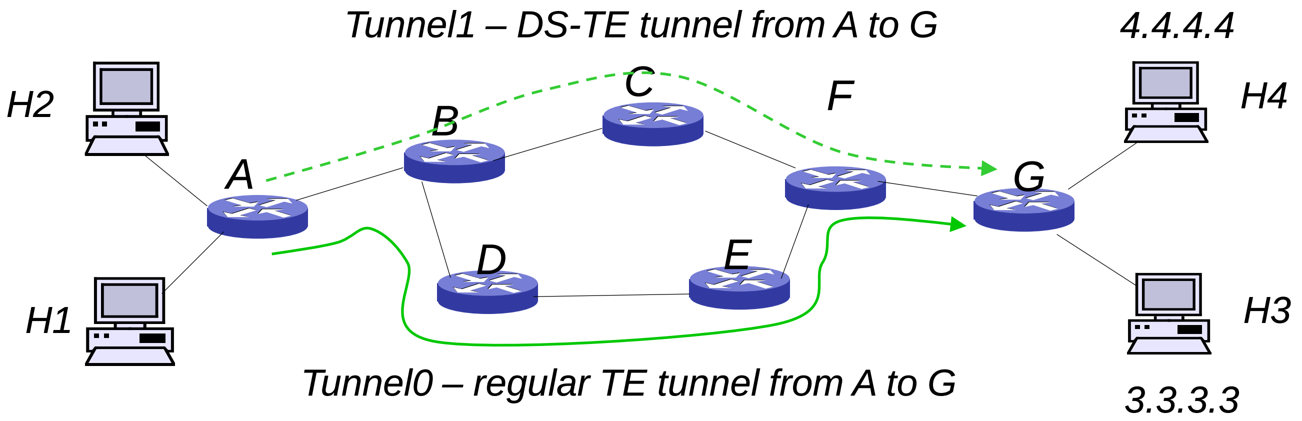
- Five steps involved:
  - ✓ Advertise a per-link subpool & its bandwidth availability
    - ➢ ip rsvp bandwidth 150000 **sub-pool** 45000
  - ✓ Specify per-link scheduling, LLQ

```
Class-map match-all voice
            match mpls experimental 5
policy-map llq
            class voice
                 priority percent 30
interface POS3/0
            service-policy output llq
```

  - ✓ Tell the headEnd subpool bandwidth requirement for **path calculation** and **bandwidth reservation**
    - ➢ tunnel mpls traffic-eng bandwidth sub-pool *kbps*
  - ✓ Perform headend tunnel admission control
    - ➢ Make sure that the only traffic to enter the DS-TE tunnel is traffic that belongs there
  - ✓ Enable tunnel preemption
- Example

# Forwarding DS-TE Traffic Down a Tunnel

- Forwarding DS-TE traffic down a tunnel
  - ✓ Static routes
  - ✓ Policy-based routing
  - ✓ Autoroute
    - ➢ Easiest. Requires only one command on the headend, and all the traffic destined for or behind the tail is sent down the tunnel.
    - ➢ But, if have both TE and DS-TE tunnels to the same destination, it may not do what you want.


Tunnel1 – DS-TE tunnel from A to G
4.4.4.4
H2
C
F
H4
B
A
G
D
E
H1
H3
Tunnel0 – regular TE tunnel from A to G
3.3.3.3

# Forwarding DS-TE Traffic Down a Tunnel

- H2 has voice traffic destined for H4, and H1 has regular IP traffic destined for H3.
- If enable autoroute on both tunnels, what will happen?
  - ✓ **Load sharing**, i.e., both H3 and H4 are reachable over both tunnels.
  - ✓ Need to forward ONLY the voice traffic down Tunnel1
- Need to use static route, so that H3 is only reachable over Tunnel0
  - ✓ Example: ip route 3.3.3.3 255.255.255.0 Tunnel0
- What if there are many hosts that receive voice traffic?
  - ✓ Static routes are reasonable for a small-scale problem
  - ✓ Need to aggregate devices into subnets.

# Modular QoS CLI (MQC) and Example

# MQC

- Basic commands
  - ✓ Class map – defines a traffic class, or how you define what traffic you're interested in
  - ✓ Policy map – what you do to the traffic defined in a class map. Associate a class map with one or more QoS policies (bandwidth, police,  queue-limit, random detect, shape, set prec, set DSCP, set mpls exp).
  - ✓ Service policy – how you enable a policy map on an interface. Associate the policy map with an input or output interface.

# Example

- Create a simple LLQ policy matching MPLS Exp 5 traffic and assume it is VoIP traffic and Exp 4 for Business

```
Class-map match-all VOICE
     match ip dscp ef
Class-map match-all BUSINESS
   match ip dscp af31 af32 af33

policy-map llq
   class VOICE
         set mpls experimental 5
         priority percent 30
policy-map
   class BUSINESS
         set mpls experimental 4

interface POS3/0
   ip address 10.10.10.10 255.255.255.0
   service-policy output llq
….
```

# Explicit Congestion Notification

# (ECN)

# Explicit Congestion Notification

- ECN is classified as an “experimental” protocol by the IETF, has been specified but not standardized until more experience is gained with it.
- Congestion control in today’s IP networks (implicit)
  - ✓ Congestion avoidance mechanisms of TCP: timeout and packet losses are an indication of congestion.
  - ✓ TCP senders reduce their sending rates when they experience packet loss and slowly increase rates during periods when no packets have been lost.
- Drawbacks:
  - ✓ Dropped packets need to be retransmitted and will arrive later, degradation of the response time and quality
  - ✓ Dropped packets still consume resources, better not to send the packet at all.

# ECN Overview

- ECN introduces a way to explicitly signal congestion **to the sender** without dropping a packet.
- How to know congestion?
  - ✓ Need some form of queue management such as RED to monitor congestion rather than just dropping packets when the queue becomes full.
- What to do?
  - ✓ A router sets a bit (congestion experienced CE) in a packet header when it detects congestion, and then forwards the packet rather than dropping it.
- How does the sender know it and respond?
  - ✓ When a packet with the CE bit arrives at its destination, the receiver sends a signal back to the sender to reduce rate
  - ✓ The way the sender responds to it is dependent on end-to-end protocol used.
  - ✓ For TCP, ECN-echo bit in the TCP header is set and sent to the sender via ACK packet. When the sender receives it, it responds exactly as if a packet had bee dropped.
- Compatibility and deployment issue
  - ✓ Some routers are ECN-capable; some, non-ECN-capable. Sender may not reduce traffic.
  - ✓ ECN defines two new bits (2 unused bits in the ToS byte) to be carried in the IP header: CE bit and ECT bit (ECN-capable transport). If congestion:
    - ➢ If ECT bit is set, set CE bit
    - ➢ If ECT bit is not set, drop packets

# MPLS Support of ECN

- ECN should be supported in the MPLS header. Where in the MPLS header?
  - ✓ Use one bit in the Exp field
- Is it enough?  How to represent ECN states?
  - ✓ Not ECN capable
  - ✓ ECN capable AND not CE
  - ✓ ECN capable AND CE
- Rules for setting the ECN bit in the MPLS header
  - ✓ When we add the MPLS header to IP header
    - ➢ 0 ECN capable AND not CE
    - ➢ 1 Not ECN capable OR CE
  - ✓ When we remove the MPLS header

| IP ECT bit on input | MPLS ECN bit value | IP ECN bits on output |
|---|---|---|
| Not ECN capable Transport (ECT=0) | 1 | ECT=0, CE=0 |
| ECN capable (ECT=1) | 0 | ECT=1,CE=0 |
| ECN capable (ECT=1) | 1 | ECT=1,CE=1 |

- Why wait until it reaches the destination and send a notification? BECN vs. FECN