IP and MPLS

Quality of Service

Agenda

- What is Quality of Service
- Different QoS Approaches
- Quality of Service Toolset
  - TOS, Traffic Class, IP Precedence and DSCP
  - BE, CS, AF and EF PHBs
- Classification and Marking
- Policing and Shaping

<u>Congestion Management:</u>

Queuing and Different Queuing Techniques
- FIFO, CQ, PQ, WFQ, CBWFQ, LLQ

<u>Congestion Avoidance:</u>

- TCP and UDP Reactions to Packet Loss
- TCP Slow-Start and Congestion Avoidance Processes
- Tail Drop, Global Synchronization, Saw-teeth effect and TCP Starvation
- RED, WRED, AQM

Agenda

- QoS Design Best Practices – RFC 4594 Analysis
- QoS Design Models – 4 Class, 8 Class and 12 Class QoS Models
- MPLS Diffserv Tunneling Models
  - Uniform, Short-Pipe and Pipe Model Design
- Diffserv-Aware MPLS-TE Quality of Service
  - Different Bandwidth Allocation Models – MAM – RDM
- QoS Frequently Asked Questions
- QoS Summary

Quality of Service (QoS)

- Quality of service (QoS) is the overall performance of a telephony or computer network, particularly the performance seen by the users of the network

- QoS is the ability of the network to give a higher preference to one piece of data over another

Quality of Service (QoS)

- Providing Network resources to satisfy application demands
- Managing fairness!

Different QoS Approaches

Two Quality Of Service approaches have been defined by the standard organizations:

- Intserv (Integrated Services) and Diffserv (Differentiated Services)

Different QoS Approaches

- Intserv requires each and every flow to request a bandwidth from the network and network would reserve the required bandwidth for the user during a conversation

Different QoS Approaches

- Think this is an on-demand circuit switching, each flows of each user would be remembered by the network

- This clearly would create a resource problem (CPU, Memory , Bandwidth) on the network thus never widely adopted

Different QoS Approaches

- The second Quality of Service Approach is Diffserv (Differentiated Services) doesn’t require reservation but instead flows are aggregated and place into the classes

- Then each and every node can be controlled by the network operator to treat differently for the aggregated flows

Different QoS Approaches

- It is scalable approach compare to the Intserv Quality of Service model

Different QoS Approaches

Design Requirement	Integrated Services	Diffentiated Services
What is it?	QoS Architecture that specifies the elements to guarantee QoS on the Networks	QoS Architecture that specifies a simple and scalable mechanim for managing network traffic to provide QoS on the Networks
Scalablability	Low, each flow requires reservation on each and every nework hop	Good scalability as the flows are aggregated in an application classes and different application classes get different treatment, not per flow
Signalling	RSVP	Not Applicable, No reservation/no state entry
Known as	It is known as Intserv and Hard QoS	It is known as Diffserv and Soft QoS
Widely Deployed	No, not deployed	Yes, widely deployed
Complexity	Complex since it requires randevous point, Anycast RP for the Randevous point redundancy, Randevous point engineering for the optimal multicast routing	Easy, it requires source information only. There is no Randevous Point in Source Specific Multicast, no RP Engineering , no Anycast RP
Resource Requirement	Too much, each flow requires bandwidth reservation	Resource reservation is not applicable

QoS Toolset

- For Diffserv purpose, there are many tools to categorize the traffic, also traffic may need to be prioritize or punish in case of congestion, based on business and application requirements

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

- Layer 3 packets are marked with IP Precedence or Differentiated Services Code Points (DSCP) in the Type-Of-Service (TOS) byte that is in the IP Header

- In order to understand QoS we must take a look at the TOS byte and understand what the eight bits are doing within this byte

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

- IP Precedence – RFC 1812

- IP Precedence (IPP) is viewed by many as a legacy technology but must still be observed for backwards compatibility

- The second byte in an IPv4 packet is the TOS byte. The first 3 bits are referred to as the IP Precedence bits

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

- IP Precedence – RFC 1812

- The IP Precedence bit only allows for eight values (0-7), generally 6 and 7 are reserved for network control traffic such as routing protocols

- The value of 0 is normally reserved for default behavior, leaving only 5 values for traffic other than best effort behavior

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

- IPP Value of 5 is recommended for voice
- IPP Value of 4 is recommended for interactive and streaming video
- IPP Value of 3 is recommended for call control and signaling

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

- The IPP value of 1 and 2 are remaining markings for all data applications

- This is commonly found to be too restrictive resulting in a move to the more scalable 6 bit 64 value Differentiated Services Code Point (DSCP)

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

- The IPP bits are mainly used to classify packets at the edge of the network into one of the eight possible categories

- Packets of lower precedence (lower values) can be dropped in favor of higher precedence when there is congestion on the network

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

RFC 1812-IP Precedence

TOS	IP Prec Value	4	2	1
Routine - Best Effort Data	0	0	0	0
Priority - Medium Priority Data	1	0	0	1
Immediate - High Priority Data	2	0	1	0
Flash- call control/signaling	3	0	1	1
Flash override- video	4	1	0	0
Critical-VOIP	5	1	0	1
Internetworking-routing	6	1	1	0
Network	7	1	1	1

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

Differentiated Services Code Point (DSCP) – RFC 2474

- DSCP uses the same three bits as IP Precedence uses as well as the next three bits for a total of six bits

- Six bits provides for a range of 64 different DSCP values

QoS Toolset – TOS, Traffic Class, IP Precedence and DSCP

- These values can be expressed in numeric form or by keyword names, called per-hop behaviors (PHB)

- A collection of packets that has the same DSCP value in the TOS byte and crossing in a particular direction is called a Behavior Aggregate (BA)/Traffic Class

QoS Toolset – DSCP and PHB

- In computer networking, per-hop behavior (PHB) is a term used in differentiated services (DiffServ) or multiprotocol label switching (MPLS). It defines the policy and priority applied to a packet when traversing a hop (such as a router) in a DiffServ network

- For vendor interoperability purpose, IETF defined 4 different PHBs – Per Hob Behavior to specify treatment for the packets from QoS perspective.

- These are namely BE – Best Effort, CS – Class Selector , AF - Assured Forwarding and EF – Expedite Forwarding PHBs.

- All are standard PHBs, not Cisco or any other specific

QoS Toolset – DSCP and PHB

The Class Selector PHB and DSCP Values

Class Selector – RFC 2474

Class Selector (CS) is used to provide for backward compatibility with IP Precedence. Devices that don't understand DSCP values, can still have standard based QoS policy, thanks to CS PHBs

Just like IPP, CS has 0s in the 4^{th}, 5^{th} and 6^{th} bits of the TOS byte

TOS byte in IP header is 8 bits, 6 bits are used and names DSCP bits in QoS

QoS Toolset – DSCP and PHB

The Class Selector PHB and DSCP Values

Class Selector – RFC 2474

For example, if you are sending packets to a router that only understands IPP markings you could send CS marked packets of 101000

This value is 40 in DSCP values but is interpreted as IPP 5 in the router that only understands IPP

QoS Toolset – DSCP and PHB

The Class Selector PHB and DSCP Values

DSCP Class Selector Names	Binary DSCP Values	IPP Binary Values	IPP Names
Default/CS0*	000000	000	Routine
CS1	001000	001	Priority
CS2	010000	010	Immediate
CS3	011000	011	Flash
CS4	100000	100	Flash Override
CS5	101000	101	Critical
CS6	110000	110	Internetwork Control
CS7	111000	111	Network Control

QoS Toolset – DSCP and PHB

The AF PHB and DSCP Values

Assured Forwarding (AF) – RFC 2597

- AF defines a method by which packets can be given different forwarding assurances
- Traffic can be divided into different classes and then each class given a certain percentage of bandwidth
- For example, one class could have 50% of the available link bandwidth; one class could have 30% and another 20% of the bandwidth

QoS Toolset – DSCP and PHB

The AF PHB and DSCP Values

- Assured forwarding is denoted by the letters AF and then two digits
- The first digit denotes the AF class and can range from 1–4
- These first 3 bits of the AF correspond to IPP. The second digit refers to the level of drop probability within the AF class

QoS Toolset – DSCP and PHB

The AF PHB and DSCP Values

- Something interesting to note about AF is the first 3 bits are the same for the three drop probabilities for each group
- Also notice that a Class 1 AF would correspond to an IPP of 1, and a Class 2 AF would correspond to an IPP of 2 and so on

  - Class 1 AF PHB = 001
  - Class 2 AF PHB = 010
  - Class 3 AF PHB = 011
  - Class 4 AF PHB = 100

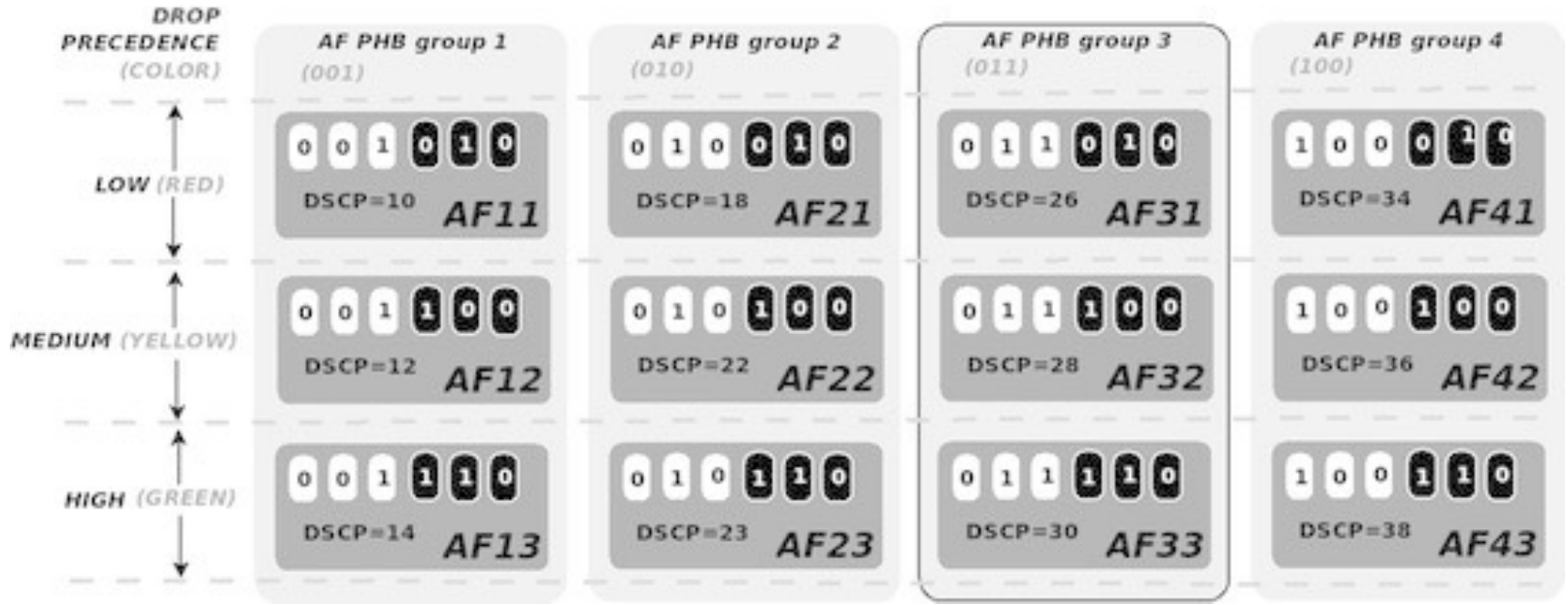
QoS Toolset – DSCP and PHB

The AF PHB and DSCP Values

- The second digit, or the drop probability, functions in the following way during periods of congestion: the higher the number, the more likely the packet is to be dropped
- For example, packets assigned AF13 will be dropped before packets in the AF12 class
- RED and WRED can be applied to AF classes

QoS Toolset – DSCP and PHB

The AF PHB and DSCP Values



QoS Toolset – DSCP and PHB

The BE PHB and DSCP Values

- Best effort traffic should be marked with DSCP 0
- Adequate bandwidth should be assigned to the Best-Effort class as a whole because the majority of applications default to this class

QoS Toolset – DSCP and PHB

The BE PHB and DSCP Values

- It is recommended to reserve at least 25% for best effort traffic
- In most networks there are hundreds, if not thousands, of applications that assign their IP packets to a default of DSCP 0
- Consequently, adequate bandwidth needs to be provisioned to allow for the sheer volume of packets that will be placed in the default class.

QoS Toolset – DSCP and PHB

Expedited Forwarding – RFC 2598

- EF PHB provides a low-loss, low-latency, low-jitter, and assured bandwidth service
- Applications such as VoIP, Video, and other time sensitive applications require a robust network treatment like EF

QoS Toolset – DSCP and PHB

The EF – Expedited Forwarding PHB and DSCP Values

- EF can be implemented using priority queuing, along with rate limiting for these time sensitive packets
- EF should only be used for only the most critical applications
- If congestion exists it is possible to treat too much traffic as EF and oversubscribe the queues anyway.

QoS Toolset – Classification and Marking

- First action that we have to perform in overall QoS design, is to identify/categorize one packet or packet flow is different from another, this is called classification
- An action that organizes packets into different traffic types, to which different policies can then be applied
- Classification of packets can happen without marking

QoS Toolset – Classification and Marking

- Writes a value into the packet header
- When it is done once, you don’t need to do the Deep Packet inspection at every hop again, so you can reduce te performance impact

QoS Toolset – Classification and Marking

Classification can be done on:

- Layer 1 – such as ingress physical or sub-interface
- Layer 2 – IEEE 802.1Q/p COS bits
- Layer 3 – TOS Byte/DSCP
- Layer 4 – TCP/UDP Ports
- Layer 7 – Application Based, for example using NBAR

QoS Toolset – Classification and Marking

- Layer 1 classification is not scalable, it is hard to manage
- Layer 2 classification and marking may not be end to end, if packet travels from Ethernet to Non-Ethernet, marking values get lost
- Layer 3 classification and marking is end to end, thus recommended option

QoS Toolset – Classification and Marking

- If more granular classification and marking is necessary, Layer 4 and Layer 7 might be needed

- Layer 7 would be necessary if there are applications tunnelled through HTTP, so they look like same, using port 80, but you may need to threat the applications differently

QoS Toolset – Classification and Marking

Marking can be done on:

- Layer 2 – IEEE 802.1Q/p COS
- Layer 2.5 – MPLS EXP
- Layer 3 – IP DSCP
- Internal in Router – QoS Group

QoS Toolset – Policing and Shaping

- QoS policing and shaping mechanisms are used to identify traffic violation and make responses
- Policing and shaping adopt the same algorithms for identifying traffic violation, but they make different responses

QoS Toolset – Policing and Shaping

- The policing mechanism checks traffic in real time, and takes immediate actions according to the settings when it discovers violation
- For example, the policing mechanism can identify if the traffic payload exceeds the defined traffic flow rate, and then decide to re-mark or drop the excessive parts
- It can control the traffic of both inbound and outbound directions

QoS Toolset – Policing and Shaping

- The shaping mechanism works together with queuing mechanism

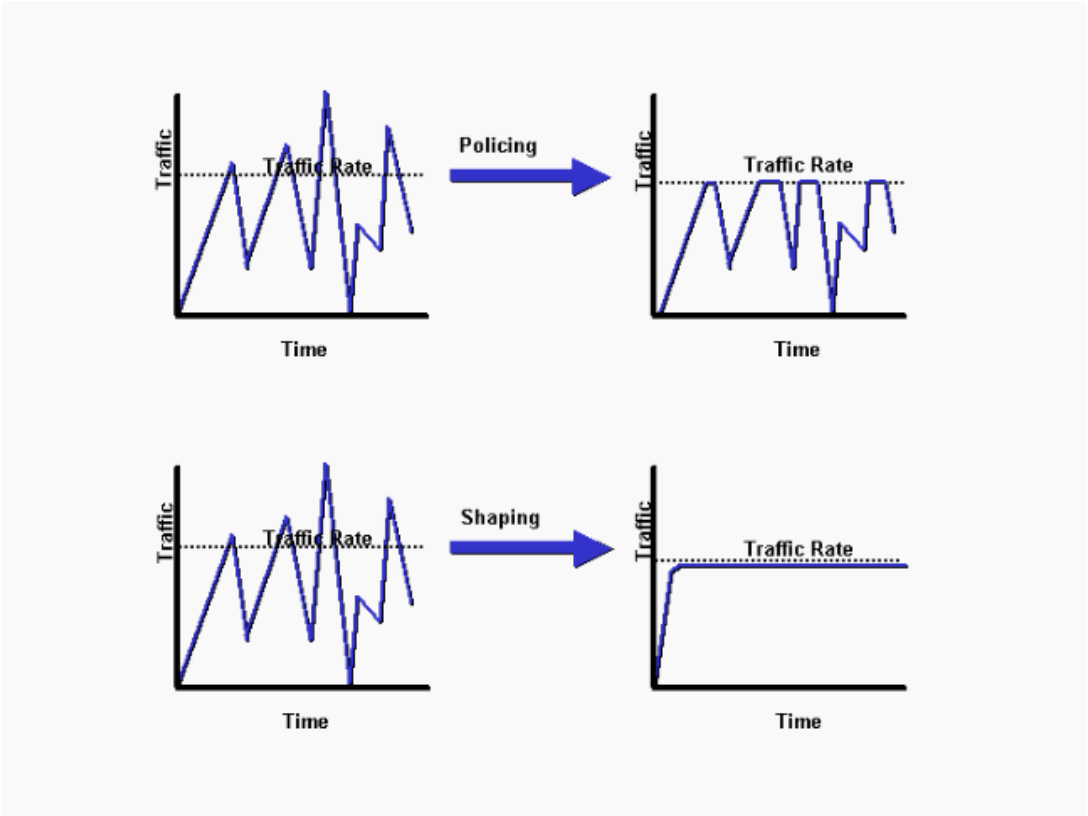
QoS Toolset – Policing and Shaping

- Policing perform checks for traffic violations agains a configured rate and take immediate actions – such as dropping or remarking
- Policing don’t delay the traffic
- Policing can be done at Data plane and control plane

QoS Toolset – Policing and Shaping

- Shaping smoot out the bursty traffic for helping infrastructure to deal with them more easily
- Smoot out the peaks of traffic arrival, so that it never exceeds the configured rate
- If the offered traffic momentarily spikes above the contracted rate, the excess traffic is buffered and delayed, until the offered traffic once agains goes below the configured rate

QoS Toolset – Policing and Shaping



QoS Toolset – Queuing

- Congestion in Internet occurs when the link bandwidth exceeds the capacity of available routers or when the arrival rate of packets is greater than the departure rate due to one of the following two reasons:

1. Input interface is faster than the output interface
2. Output interface is receiving packets coming in from multiple other interfaces

QoS Toolset – Queuing

- Primary role of router is to switch packets from the input links to output links through buffer
- Apart from forwardign the packets, routers involve controlling the congestion in the network
- Congestion management is done with Queue management and scheduling

QoS Toolset – Queuing

- Congestion management features allow you to control congestion by determining the order in which packets are sent out an interface based on priorities assigned to those packets

- Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission

QoS Toolset – Queuing

- If you use congestion management features, packets accumulating at an interface are queued until the interface is free to send them; they are then scheduled for transmission according to their assigned priority and the queueing mechanism configured for the interface

- The router determines the order of packet transmission by controlling which packets are placed in which queue and how queues are serviced with respect to each other

QoS Toolset – Queuing

FIFO Queuing:

- The first reason that a router or switch needs output queues is to hold a packet while waiting for the interface to become available for sending the packet

- Whereas the other queuing tools in this course also perform other functions, like reordering packets, FIFO Queuing just provides a means to hold packets while they are waiting to exit an interface

QoS Toolset – Queuing

FIFO Queuing:

- In FIFO queuing, classification and scheduling is not necessary
- FIFO Queuing uses a single queue for the interface
- Because there is only one queue, there is no need for classification to decide the queue into which the packet should be placed

QoS Toolset – Queuing

FIFO Queuing:

- Also, there is no need for scheduling logic to pick which queue from which to take the next packet
- The only really interesting part of FIFO Queuing is the queue length, which is configurable, and how the queue length affects delay and loss

QoS Toolset – Queuing

FIFO Queuing:

- FIFO Queuing uses tail drop to decide when to drop or enqueue packets
- If you configure a longer FIFO queue, more packets can be in the queue, which means that the queue will be less likely to fill
- If the queue is less likely to fill, fewer packets will be dropped

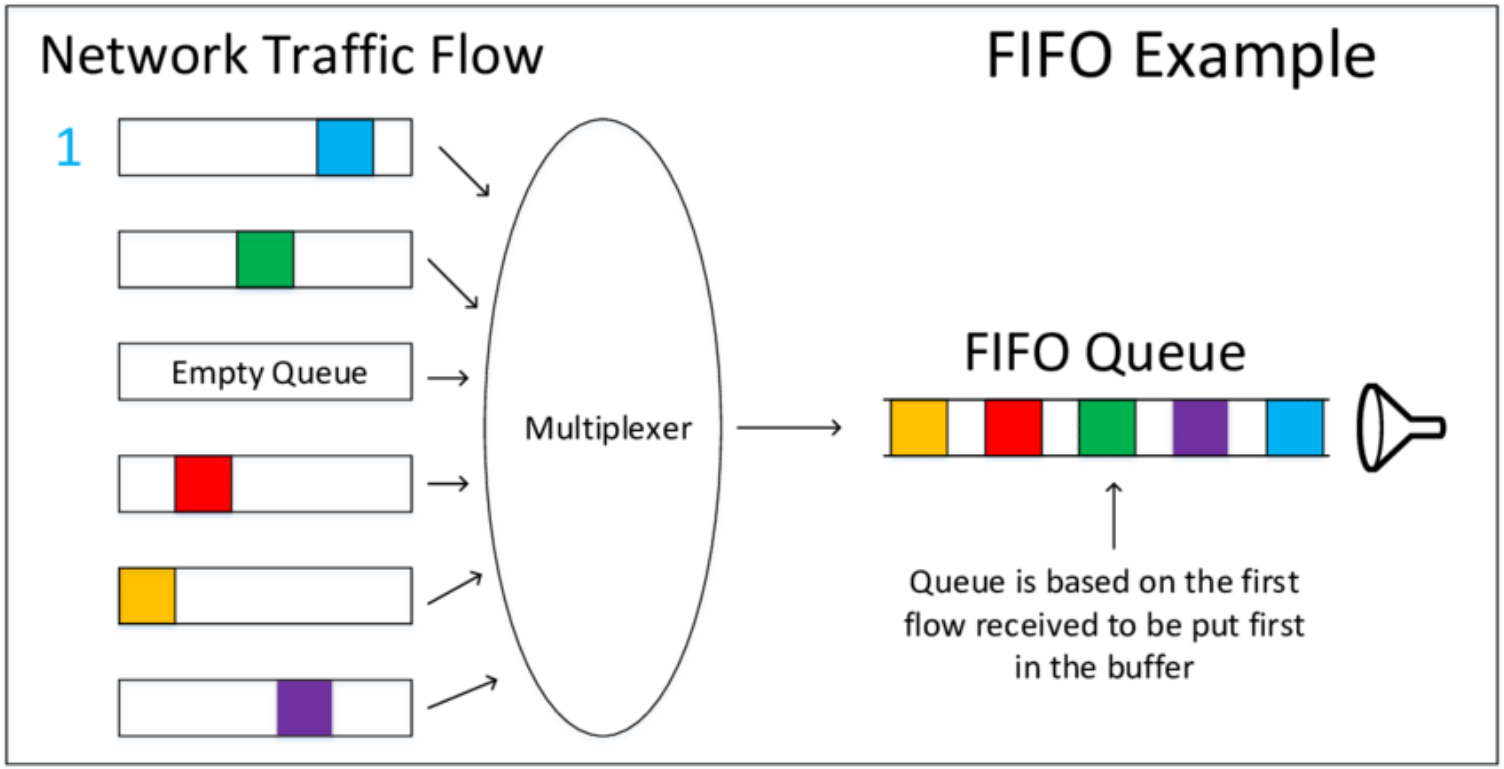
QoS Toolset – Queuing

FIFO Queuing:

- However, with a longer queue, packets may experience more delay and jitter
- With a shorter queue, less delay occurs, but the single FIFO queue fills more quickly, which in turn causes more tail drops of new packets
- These facts are true for any queuing method, including FIFO

QoS Toolset – Queuing

FIFO Queuing:



QoS Toolset – Queuing

Priority Queuing:

- Priority Queuing's most unique feature is its scheduler.
- PQ schedules traffic such that the higher-priority queues always get serviced, with the side effect of starving the lower-priority queues
- With a maximum of four queues, called High, Medium, Normal, and Low

QoS Toolset – Queuing

Priority Queuing:

- If there is a packet in the High Queue the scheduler will always take the packets in the High queue
- If the High queue does not have a packet waiting, but the Medium queue does, one packet is taken from the Medium queue—and then the process always starts over at the High queue
- The Low queue only gets serviced if the High, Medium, and Normal queues do not have any packets waiting

QoS Toolset – Queuing

Priority Queuing:

- The PQ scheduler has some obvious benefits and drawbacks
- Packets in the High queue can claim 100 percent of the link bandwidth, with minimal delay, and minimal jitter
- The lower queues suffer!
- In fact, when congested, packets in the lower queues take significantly longer to be serviced than under lighter loads

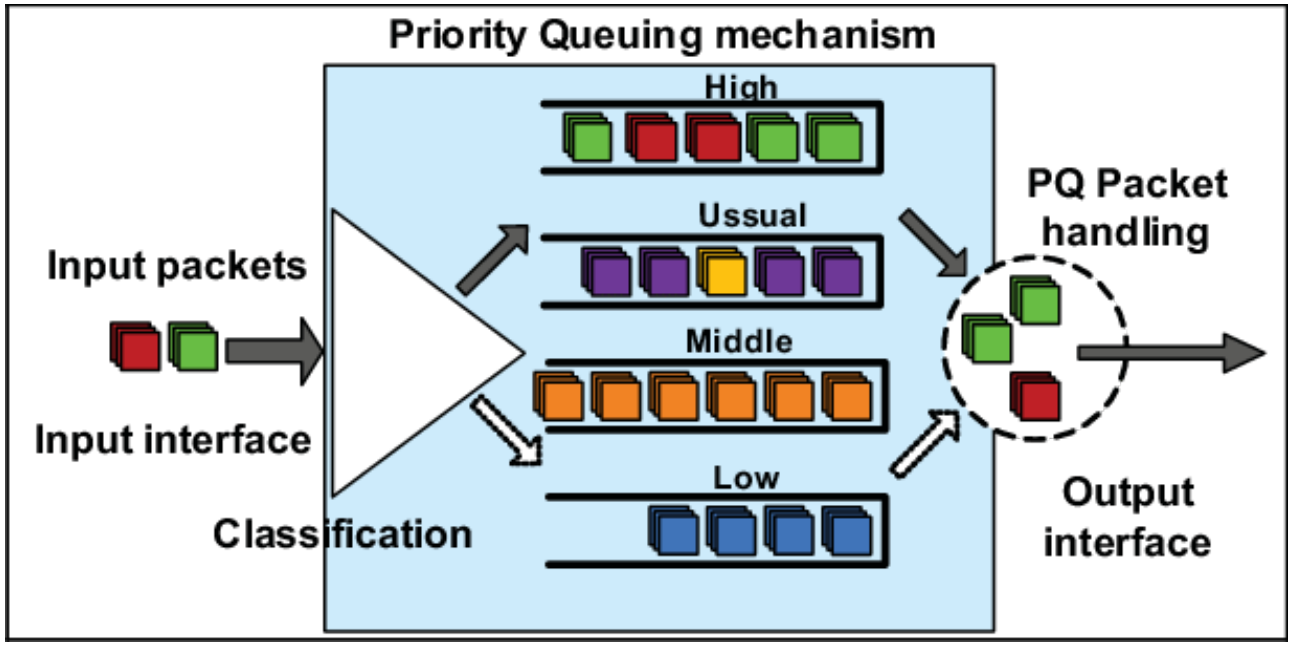
QoS Toolset – Queuing

Priority Queuing:

- Priority Queuing classification is done based on matching an ACL for all Layer 3 protocols, incoming interface, packet size, whether the packet is a fragment, and TCP and UDP port numbers
- PQ Drop is done based on Tail Drop
- Scheduling inside a single queue, for example in High Queue is still done based on FIFO

QoS Toolset – Queuing

Priority Queuing:



QoS Toolset – Queuing

Custom Queuing:

- CQ addresses the biggest drawback of PQ by providing a queuing tool that does service all queues, even during times of congestion
- It has 16 queues available, implying 16 classification categories, which is plenty for most applications
- The negative part of CQ, as compared to PQ, is that CQ's scheduler does not have an option to always service one queue first—like PQ's High queue— so CQ does not provide great service for delay- and jitter-sensitive traffic

QoS Toolset – Queuing

Custom Queuing:

- The CQ scheduler performs round-robin service on each queue, beginning with Queue 1
- CQ takes packets from the queue, until the total byte count specified for the queue has been met or exceeded
- After the queue has been serviced for that many bytes, or the queue does not have any more packets, CQ moves on to the next queue, and repeats the process

QoS Toolset – Queuing

Custom Queuing:

- The CQ scheduler essentially guarantees the minimum bandwidth for each queue, while allowing queues to have more bandwidth under the right conditions

- If any given time, one of the configured queue doesn’t have a traffic, scheduler moves to the. Next queue immediately and time slot is spent for the remaining queues

QoS Toolset – Queuing

Custom Queuing:

- Unlike PQ, CQ does not name the queues, but it numbers the queues 1 through 16
- No single queue has a better treatment by the scheduler than another, other than the number of bytes serviced for each queue
- If one queue is configured with 1000 byte and the second queue is 5000 byte, the second the gets 5 times more scheduler time

QoS Toolset – Queuing

Custom Queuing:

- CQ Classifies based on matching an ACL for all Layer 3 protocols, incoming interface, packet size, whether the packet is a fragment, and TCP and UDP port numbers
- Default drop policy for CQ is Tail Drop
- Withing a single queue, scheduling is done based on FIFO

QoS Toolset – Queuing

Custom Queuing:

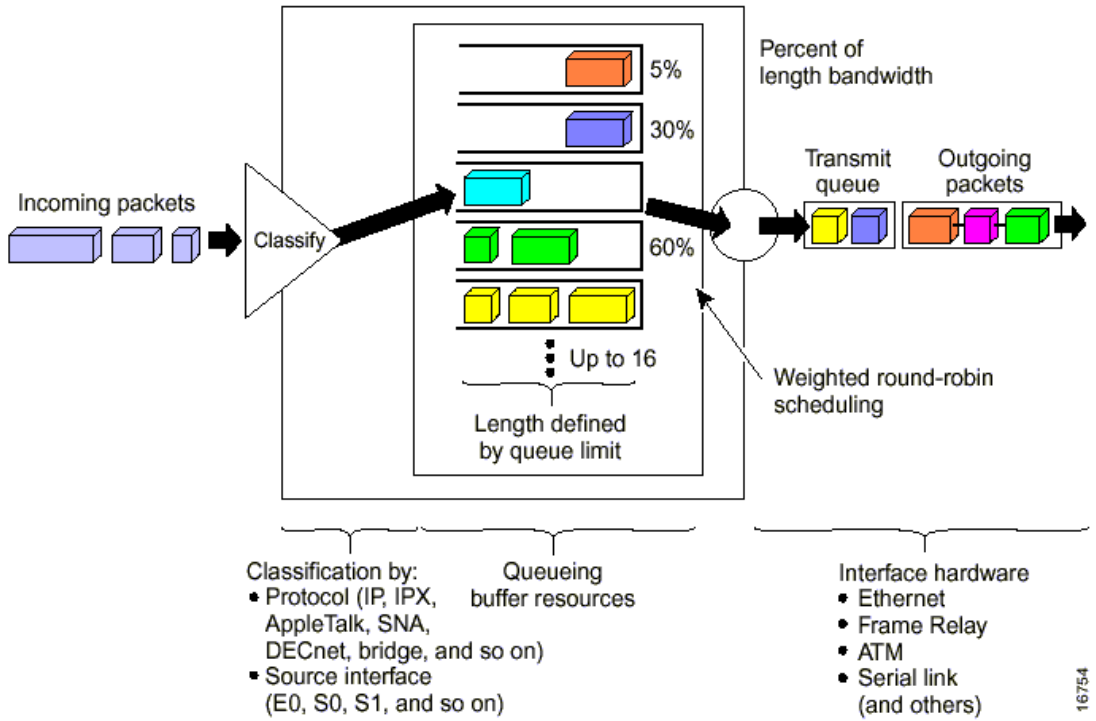


Figure #7 (stolen from Cisco)

QoS Toolset – Queuing

Weighted Fair Queuing:

- Weighted fair queuing differs from PQ and CQ in several significant ways
- The most obvious difference is that WFQ does not allow classification options to be configured
- WFQ automatically classifies packets based on flows, with each flow being placed into a separate queue

QoS Toolset – Queuing

Weighted Fair Queuing:

- Flow is defined by using, Source IP, Destination IP, Source Port, Destination Port and IP Precedence value for WFQ purpose
- Because WFQ puts packets of different flows in different queues, it necessarily has a much larger number of queues than any of the non-flow-based queuing tools
- The WFQ scheduler uses logic that is quite different from the logic of other queuing tools so that it can deal with the larger number of queues

QoS Toolset – Queuing

Weighted Fair Queuing:

Goal of the WFQ Scheduler:

- Flows with the same IPP should be given the same amount of bandwidth, regardless of how many bytes are sent in each flow
- For flows with different IPP values, give flows with higher IPP a proportionally higher amount of bandwidth
- The result: WFQ favors lower-volume, higher IPP flows

QoS Toolset – Queuing

Weighted Fair Queuing:

How WFQ gives more bandwidth to higher IP Precedence value flows?

One of the goals of the WFQ scheduler is to provide more bandwidth to flows with higher IPP values

To do so, the flows are weighted based on IPP plus 1

In other words, precedence 7 flows get eight times more bandwidth than precedence 0 flows, because (7 + 1) / (0 + 1) = 8. For another example, if you compare precedence 3 to precedence 0, the ratio is (3 + 1) / (0 + 1) = 4

QoS Toolset – Queuing

Weighted Fair Queuing:

Scheduler works in WFQ mechanism:

- The WFQ scheduler takes the packet with the lowest sequence number (SN) (also sometimes called finish time, or FT) when it needs to move the next packet to the hardware queue
- WFQ assigns each packet an SN when the packet is added to a WFQ flow queue
- The SN assignment process is actually the more interesting part of the scheduler

QoS Toolset – Queuing

Weighted Fair Queuing:

- The WFQ scheduler includes both the packet length and IPP when calculating the SN. The formula for calculating the SN for a packet is as follows:

Previous_SN + (weight * new_packet_length) Where weight is calculated as follows:

- The formula considers the length of the new packet, the weight of the flow, and the previous SN
  - By considering the packet length, the SN calculation results in a higher number for larger packets, and a lower number for smaller packets
  - By including the SN of the previous packet enqueued into that queue, the formula assigns a larger number for packets in queues that already have a larger number of packets enqueued. And by putting the weight (IPP + 1) in the denominator, packets with higher IPP values end up with lower SNs

QoS Toolset – Queuing

Weighted Fair Queuing:

- The drawback of WFQ is that it is very resource-intensive because of the bit computations
- The original WFQ idea also consumes many resources because the flows are not aggregated into classes with limited queues
- Instead, each flow or stream gets its own queue or buffer quota, existing implementations aggregate flows into the traffic classes, thus computationally less intensive

QoS Toolset – Queuing

Class-Based Weighted Fair Queuing:

- CBWFQ addresses some of the limitations of PQ, CQ, and WFQ
- CBWFQ allows creation of user-defined classes, each of which is assigned to its own queue
- Each queue receives a user-defined (minimum) bandwidth guarantee, but it can use more bandwidth if it is available

QoS Toolset – Queuing

Class-Based Weighted Fair Queuing:

- In contrast to PQ, no queue in CBWFQ is starved
- Unlike PQ and CQ, you do not have to define classes of traffic to different queues using complex access lists
- WFQ does not allow creation of user-defined classes, but CBWFQ does; moreover, defining the classes for CBWFQ is done with class maps, which are flexible and user friendly, unlike access lists

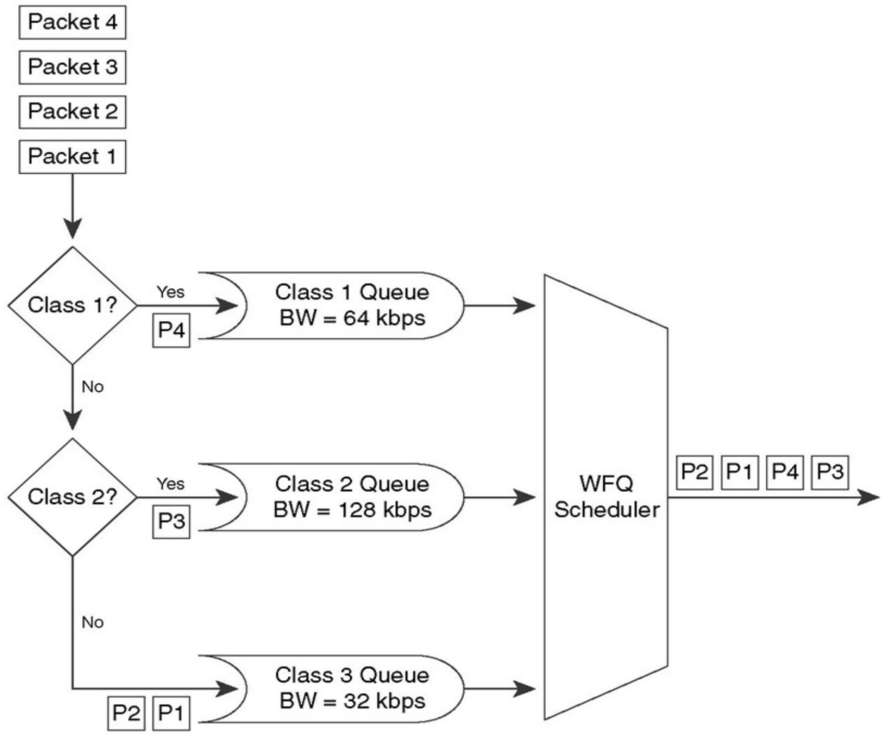
QoS Toolset – Queuing

Class-Based Weighted Fair Queuing:

- Similar to WFQ and CQ, CBWFQ does not address the low-delay requirements of real-time applications such as VoIP
- **CBWFQ is used to provide minimum guaranteed bandwidth!**

QoS Toolset – Queuing

Class-Based Weighted Fair Queuing:



QoS Toolset – Queuing

Class-Based Weighted Fair Queuing:

- CBWFQ can create up to 64 queues, one for each user-defined class
- Each queue is a FIFO queue with a defined bandwidth guarantee and If a queue reaches its maximum packet limit, tail drop occurs
- To avoid tail drop, you can apply WRED to a queue

QoS Toolset – Queuing

Class-Based Weighted Fair Queuing:

- **The main benefits of CBWFQ are as follows:**

1. It allows creation of user-defined traffic classes. These classes can be defined using MQC class maps

2. It allows allocation/reservation of bandwidth for each traffic class based on user policies and preferences

3. Defining a few (up to 64) fixed classes based on the existing network applications and user policies, rather than relying on automatic and dynamic creation of flow-based queues (as WFQ does), provides for finer granularity and scalability

QoS Toolset – Queuing

Class-Based Weighted Fair Queuing:

- **The drawback of CBWFQ is that it** does not offer a queue suitable for real-time applications such as voice or video over other IP applications

- Real-time applications expect low-delay guarantee in addition to bandwidth guarantee, which CBWFQ does not offer

QoS Toolset – Queuing

Low-Latency Queuing:

- Neither WFQ nor CBWFQ can provide guaranteed bandwidth and low-delay guarantee to selected applications such as VoIP; that is because those queuing models have no priority queue

- Certain applications such as VoIP have a small end-to-end delay budget and little tolerance to jitter (delay variation among packets of a flow)

QoS Toolset – Queuing

Low-Latency Queuing:

- LLQ includes a strict-priority queue that is given priority over other queues, which makes it ideal for delay and jitter-sensitive applications
- Unlike the plain old PQ, whereby the higher-priority queues might not give a chance to the lower-priority queues and effectively starve them, the LLQ strict-priority queue is policed
- This means that the LLQ strict-priority queue is a priority queue with a minimum bandwidth guarantee, but at the time of congestion, it cannot transmit more data than its bandwidth permits

QoS Toolset – Queuing

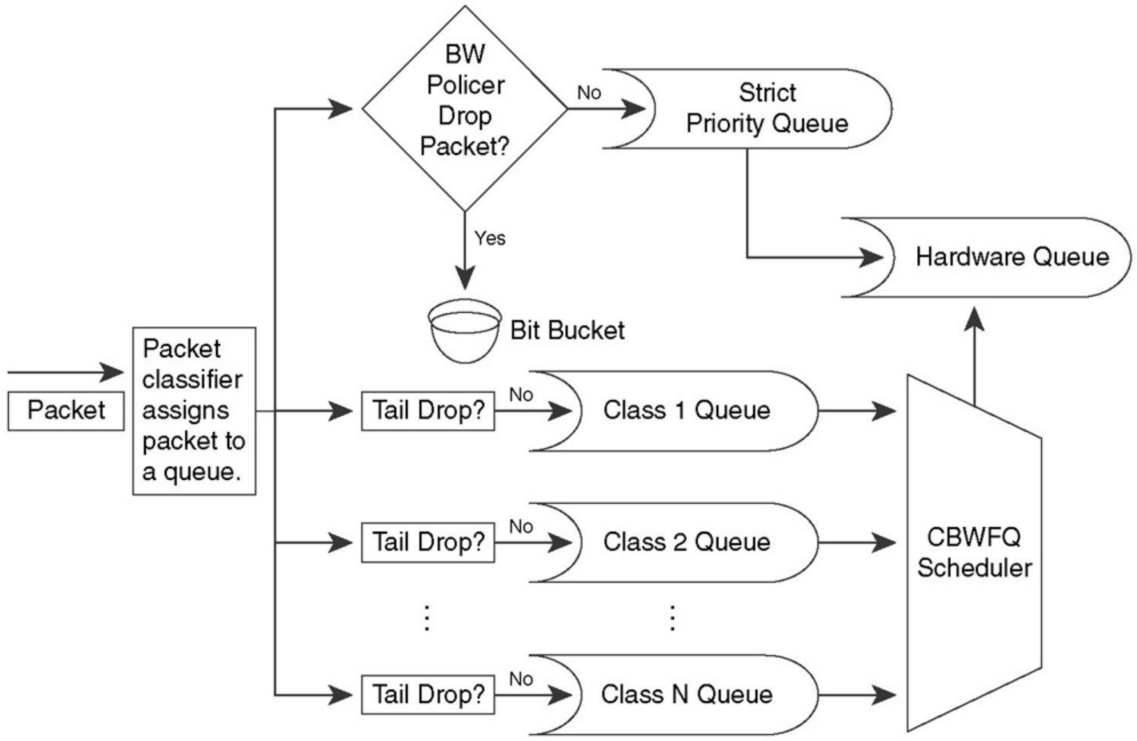
Low-Latency Queuing:

- If more traffic arrives than the strict-priority queue can transmit (due to its strict bandwidth limit), it is dropped

- Thus, at times of congestion, other queues do not starve, and get their share of the interface bandwidth to transmit their traffic

QoS Toolset – Queuing

Low-Latency Queuing:



QoS Toolset – Queuing

Low-Latency Queuing:

As you can see from the previous figure, LLQ is effectively a CBWFQ with one or more strict-priority queues added

Please note that it is possible to have more than one strict priority queue

This is usually done so that the traffic assigned to the two queues—voice and video traffic, for example—can be separately policed

However, after policing is applied, the traffic from the two classes is not separated; it is sent to the hardware queue based on its arrival order (FIFO)

QoS Toolset – Queuing

Low-Latency Queuing:

- LLQ offers all the benefits of CBWFQ, including the ability of the user to define classes and guarantee each class an appropriate amount of bandwidth and to apply WRED to each of the classes (except to the strict-priority queue) if needed

- In the case of LLQ and CBWFQ, the traffic that is not explicitly classified is considered to belong to the class-default class

QoS Toolset – Queuing

Low-Latency Queuing:

- The benefit of LLQ over CBWFQ is the existence of one or more strict-priority queues with bandwidth guarantees for delay- and jitter-sensitive traffic
- The advantage of LLQ over the traditional PQ is that the LLQ strict-priority queue is policed
- That eliminates the chance of starvation of other queues, which can happen if PQ is used

QoS Toolset – Queuing

Low-Latency Queuing:

- LLQ is one of the most commonly used Queuing mechanism today!
- It is basically a CBWFQ + PQ

QoS Toolset – Queuing

Queuing Discipline	**Default on Some Router Interfaces**	**Number of Queues**	**Allows User-Defined Classes**	**Allows User-Definable Interface Bandwidth Allocation**	**Provides a High-Priority Queue for Delay-Sensitive Traffic**	**Adequate for Both Delay-Sensitive and Mission-Critical Traffic**	**Configured Using** MQC
FIFO	Yes	1	No	No	No	No	No
PQ	No	4	Yes	No	Yes	No	No
WRR (CQ)	No	User defined	Yes	Yes	No	No	No
WFQ	Yes	Number of active flows	No	No	No	No	No
CBWFQ	No	User defined	Yes	Yes	No	No	Yes
LLQ	No	User defined	Yes	Yes	Yes	Yes	Yes

Congestion Avoidance

- When congestion occurs, queuing mechanisms kick in and queue started to be full
- When it is completely full, all the packets drop, this is called Tail Drop
- It may not be the desired behavior, we will see why

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- UDP and TCP behave very differently when packets are lost
- UDP, by itself, does not react to packet loss, because UDP does not include any mechanism with which to know whether a packet was lost
- TCP senders, however, slow down the rate at which they send after recognizing that a packet was lost

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- Unlike UDP, TCP includes a field in the TCP header to number each TCP segment (sequence number), and another field used by the receiver to confirm receipt of the packets (acknowledgment number)

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- When a TCP receiver signals that a packet was not received, or if an acknowledgment is not received at all, the TCP sender assumes the packet was lost, and resends the packet

- More importantly, the sender also slows down sending data into the network

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- TCP uses two separate window sizes that determine the maximum window size of data that can be sent before the sender must stop and wait for an acknowledgment

- The first of the two different windowing features of TCP uses the Window field in the TCP header, which is also called the receiver window or the advertised window

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- The receiver grants the sender the right to send x bytes of data before requiring an acknowledgment, by setting the value x into the Window field of the TCP header

- The receiver grants larger and larger windows as time goes on, reaching the point at which the TCP sender never stops sending, with acknowledgments arriving just before a complete window of traffic has been sent

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- The second window used by TCP is called the congestion window, or CWND, as defined by RFC 2581

- Unlike the advertised window, the congestion window is not communicated between the receiver and sender using fields in the TCP header

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- Instead, the TCP sender calculates CWND

- CWND varies in size much more quickly than does the advertised window, because it was designed to react to congestion in networks

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- The TCP sender always uses the lower of the two windows to determine how much data it can send before receiving an acknowledgment

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- The receiver window is designed to let the receiver prevent the sender from sending data faster than the receiver can process the data
- The CWND is designed to let the sender react to network congestion by slowing down its sending rate
- It is the variation in the CWND, in reaction to lost packets, which RED relies upon

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- To understand how RED works, you need to understand the processes by which a TCP sender lowers and increases the CWND

Congestion Avoidance

TCP and UDP Reactions to Packet Loss

- CWND is lowered in response to lost segments
- CWND is raised based on the logic defined as the TCP slow start and TCP congestion-avoidance algorithms
- In fact, most people use the term "slow start" to describe both features together, in part because they work closely together

Congestion Avoidance

TCP Slow Start Process

1. A TCP sender fails to receive an acknowledgment in time, signifying a possible lost packet
2. The TCP sender sets CWND to the size of a single segment
3. Another variable, called slow start threshold (SSTHRESH) is set to 50 percent of the CWND value before the lost segment
4. After CWND has been lowered, slow start governs how fast the CWND grows up until the CWND has been increased to the value of SSTHRESH
5. After the slow start phase is complete, congestion avoidance governs how fast CWND grows after CWND > SSTHRESH

Congestion Avoidance

TCP Slow Start Process

- Therefore, when a TCP sender fails to receive an acknowledgment, it reduces the CWND to a very low value (one segment size of window)

- The sender progressively increases CWND based first on slow start, and then on congestion avoidance.

Congestion Avoidance

TCP Slow Start Process

- Slow start increases CWND by the maximum segment size for every packet for which it receives an acknowledgment

Congestion Avoidance

TCP Slow Start Process

- Because TCP receivers may, and typically do, acknowledge segments well before the full window has been sent by the sender, CWND grows at an exponential rate during slow start—a seemingly contradictory concept

- Slow start gets its name from the fact that CWND has been set to a very low value at the beginning of the process, meaning it starts slowly, but slow start does cause CWND to grow very quickly

Congestion Avoidance

TCP Slow Start Process

- Congestion avoidance is the second mechanism that dictates how quickly CWND increases after being lowered

- As CWND grows, it begins to approach the original CWND value

Congestion Avoidance

TCP Slow Start Process

- If the original packet loss was a result of queue congestion, letting this TCP connection increase back to the original CWND may then induce the same congestion that caused the CWND to be lowered in the first place

Congestion Avoidance

TCP Slow Start Process

- Congestion avoidance just reduces the rate of increase for CWND as it approaches the previous CWND value

Congestion Avoidance

TCP Slow Start Process

- Once slow start has increased CWND to the value of SSTHRESH, which was set to 50 percent of the original CWND, congestion-avoidance logic replaces the slow start logic for increasing CWND

- Congestion avoidance uses a formula that allows CWND to grow more slowly, essentially at a linear rate

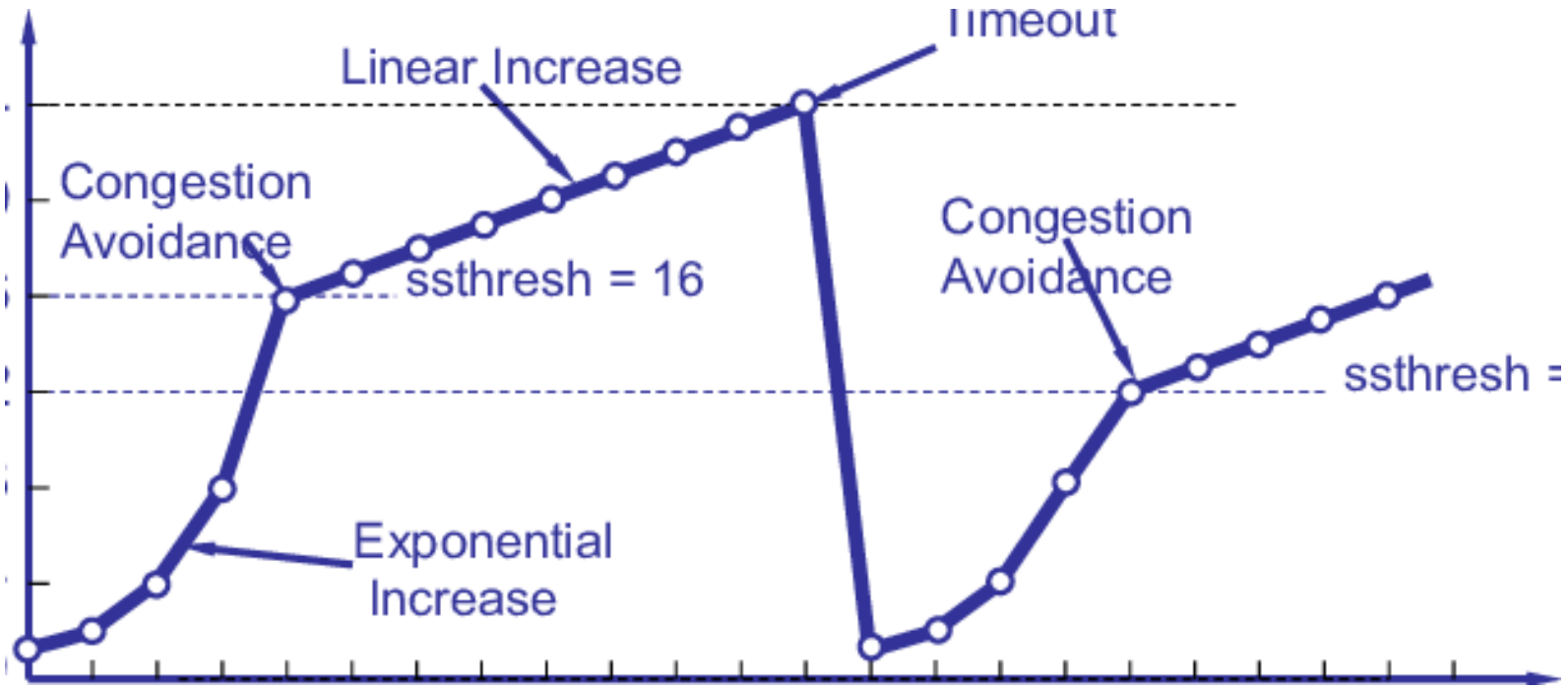
Congestion Avoidance

Slow Start and Congestion Avoidance

- Many people do not realize that the slow start process consists of a combination of the slow start algorithm and the congestion-avoidance algorithm
- With slow start, CWND is lowered, but it grows quickly
- With congestion avoidance, the CWND value grows more slowly as it approaches the previous CWND value

Congestion Avoidance

Slow Start and Congestion Avoidance



Congestion Avoidance

TCP and UDP Reaction to Packet Loss Summary

- UDP senders do not reduce or increase sending rates as a result of lost packets
- TCP senders do reduce their sending rates as a result of lost packets
- TCP senders decide to use either the receiver window or the CWND, based on whichever is smaller at the time
- TCP slow start and congestion avoidance dictate how fast the CWND rises after the window was lowered due to packet loss

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- Tail drop occurs when a packet needs to be added to a queue, but the queue is full
- However, tail drop results in some interesting behavior in real networks, particularly when most traffic is TCP based, but with some UDP traffic
- Of course, the Internet today delivers mostly TCP traffic, because web traffic uses HTTP, and HTTP uses TCP

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- In the TCP and UDP reaction to packet loss section described the behavior of a single TCP connection after a single packet loss
- Now imagine an Internet router, with 100,000 or more TCP connections running their traffic out of a high-speed interface
- The amount of traffic in the combined TCP connections finally exceeds the output line rate, causing the output queue on the interface to fill, which in turn causes tail drop

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- What happens to those 100,000 TCP connections after many of them have at least one packet dropped?

- The TCP connections reduce their CWND; the congestion in the queue abates; the various CWND values increase with slow start, and then with congestion avoidance

Congestion Avoidance

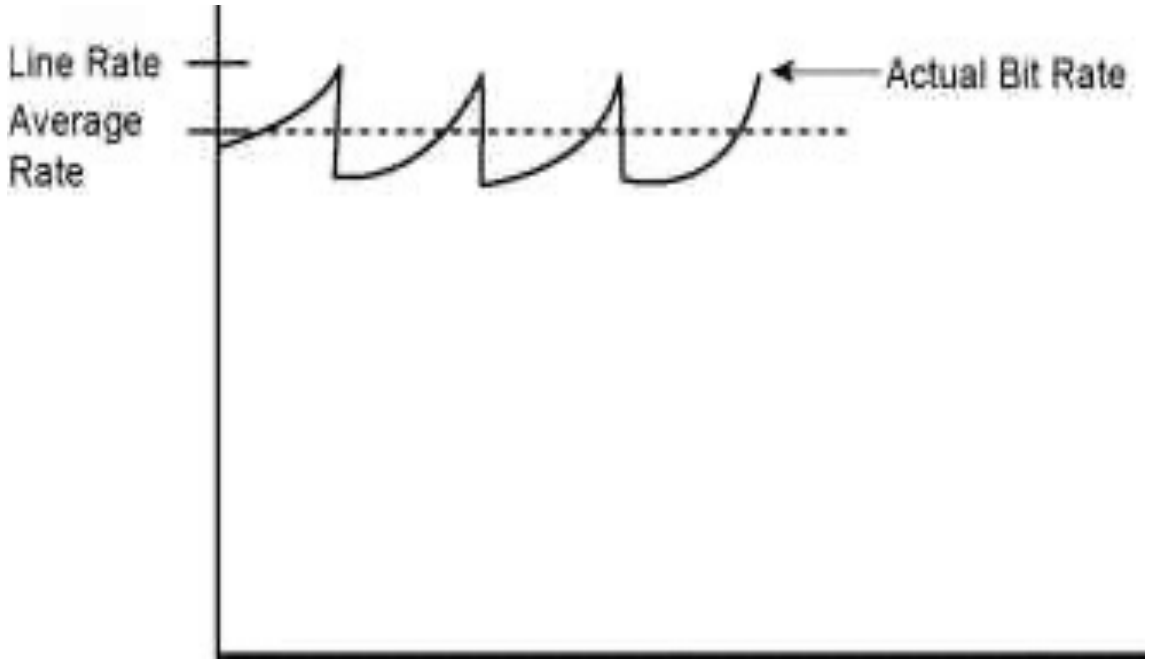
Tail Drop, Global Synchronization and TCP Starvation

- Eventually, however, as the CWND values of the collective TCP connections approach the previous CWND value, the congestion occurs again, and the process is repeated
- When a large number of TCP connections experience near simultaneous packet loss, the lowering and growth of CWND at about the same time causes the TCP connections to synchronize
- The result is called global synchronization

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- The graph shows the results of global synchronization. The router never fully utilizes the bandwidth on the link because the offered rate keeps dropping as a result of synchronization



Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- Note that the overall rate does not drop to almost nothing because not all TCP connections happen to have packets drop when tail drop occurs, and some traffic uses UDP, which does not slow down in reaction to lost packets

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- Weighted RED (WRED), when applied to the interface that was tail dropping packets, significantly reduces global synchronization

- WRED allows the average output rates to approach line rate, with even more significant throughput improvements, because avoiding congestion and tail drops decreases the overall number of lost packets

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- Another problem can occur if UDP traffic competes with TCP for bandwidth and queue space

- Although UDP traffic consumes a much lower percentage of Internet bandwidth than TCP does, UDP can get a disproportionate amount of bandwidth as a result of TCP's reaction to packet loss.

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- Imagine that on the same Internet router, 20 percent of the offered packets were UDP, and 80 percent TCP

- Tail drop causes some TCP and UDP packets to be dropped; however, because the TCP senders slow down, and the UDP senders do not, additional UDP streams from the UDP senders can consume more and more bandwidth during congestion

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- The interface output queue on this Internet router could fill with UDP packets

- If a few high-bandwidth UDP applications fill the queue, a larger percentage of TCP packets might get tail dropped—resulting in further reduction of TCP windows, and less TCP traffic relative to the amount of UDP traffic

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- The term "TCP starvation" describes the phenomena of the output queue being filled with larger volumes of UDP, causing TCP connections to have packets tail dropped

Congestion Avoidance

Tail Drop, Global Synchronization and TCP Starvation

- Tail drop does not distinguish between packets in any way, including whether they are TCP or UDP, or whether the flow uses a lot of bandwidth or just a little bandwidth

- TCP connections can be starved for bandwidth because the UDP flows behave poorly in terms of congestion control

Congestion Avoidance

RED – Random Early Detection

- Random Early Detection (RED) reduces the congestion in queues by dropping packets so that some of the TCP connections temporarily send fewer packets into the network

Congestion Avoidance

RED – Random Early Detection

- Instead of waiting until a queue fills, causing a large number of tail drops, RED purposefully drops a percentage of packets before a queue fills

- This action attempts to make the computers sending the traffic reduce the offered load that is sent into the network

Congestion Avoidance

RED – Random Early Detection

- The name "Random Early Detection" itself describes the overall operation of the algorithm

- RED randomly picks the packets that are dropped after the decision to drop some packets has been made

Congestion Avoidance

RED – Random Early Detection

- RED detects queue congestion early, before the queue actually fills, thereby avoiding tail drops and synchronization

- In short, RED discards some randomly picked packets early, before congestion gets really bad and the queue fills

Congestion Avoidance

RED – Random Early Detection

- RED is a congestion avoidance mechanism that randomly drops packets before congestion can occur

- RED uses TCP's congestion control mechanisms by dropping packets and letting TCP reduce the sender's window size

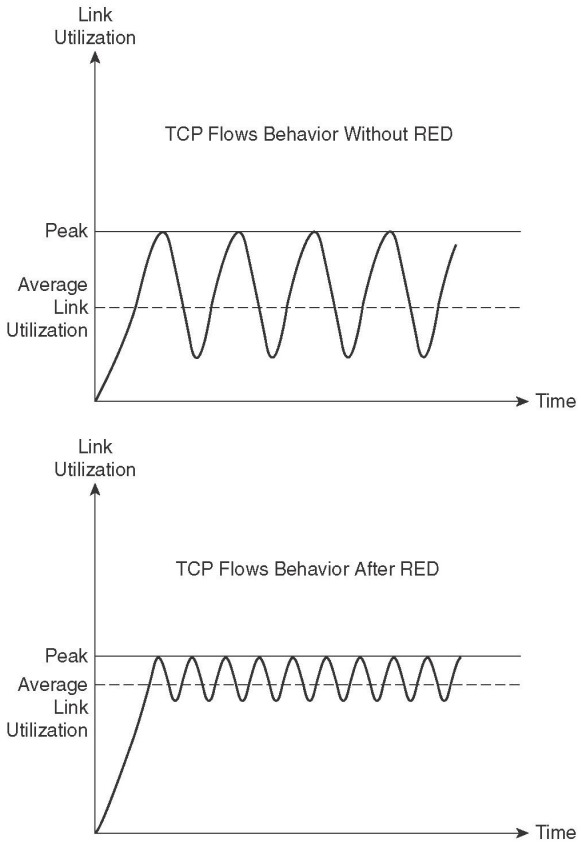
Congestion Avoidance

RED – Random Early Detection

- The disadvantage of RED is that dropped packets can affect UDP that doesn't implement windowing flow control mechanism

Congestion Avoidance

RED – Random Early Detection



Congestion Avoidance

WRED – Weighted Random Early Detection

- WRED is a Cisco implementation of RED that implements a preferential treatment of packets when determining which packets to drop when congestion occurs
- WRED uses the IP Precedence bits to determine which packets to drop
- The higher the IP Precedence is in a packet, the less likely the packet might be dropped

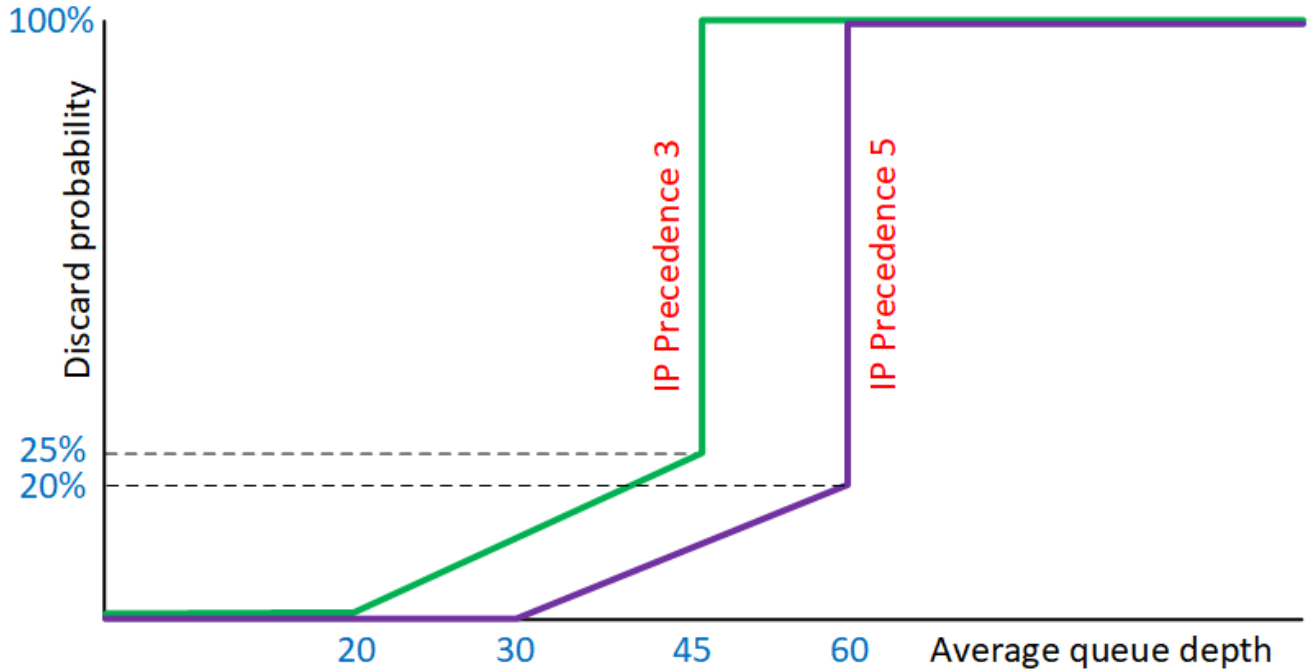
Congestion Avoidance

WRED – Weighted Random Early Detection

- Both RED and WRED avoid global TCP synchronization by randomly dropping packets
- When the sending rates of some TCP sessions slow down after their packets are dropped, other TCP sessions remain at high sending rates
- Link bandwidth is efficiently used, because TCP sessions at high sending rates always exist

Congestion Avoidance

WRED – Weighted Random Early Detection



QoS Best Practices

- Always classify and mark applications as close their source as possible
- Implement QoS always at the hardware if it is possible to avoid performance impact
- Switches support QoS in the the hardware, so for example in the Campus and Datacenter, classify and mark the traffic at the switches

QoS Best Practices

- Police flows as close to source as possible
- Not just classification, but policing also should be done to avoid performance impact of software QoS
- Police unwanted traffic flows as close to their sources as possible

QoS Best Practices

- Enable queuing at every node that has potential for congestion
- Limit LLQ to 33% of link capacity
- Use AC – Admission Control mechanism for LLQ

QoS Best Practices

- Do not enable WRED for LLQ

- Provision at least 25% for Best Effort traffic (Most of the application goes here)

QoS Best Practices

- Mark the packets with DSCP if it is possible
- Because 802.1p bit get lost when the packet enter to the IP or MPLS domain, mapping is needed
- Also, there is no standard for COS and MPLS EXP, but there is for IP DSCP (RFC 4594), thus when deploying the policy based on DSCP, we can have a better chance to have end to end consistent policy

QoS Best Practices

- Mark down traffic according to standards-based rules if possible
- If traffic exceed the CIR or PIR and if it will be markdown, follows standard-based marking values
- For example, if the application class for the conforming traffic is AF31 , exceeding traffic should be markdown with AF32 and violating traffic should be markdown with AF33

QoS Best Practices

- QoS design should support minimum 3 classes; EF (Expedited Forwarding), DF (Default Forwarding/Best Effort) and AF(Assured Forwarding)

- If company policy allows YouTube, gaming and other non-business applications, Scavenger class is created and CS1 PHB is implemented

- CS1 is defined as less than best effort service in the standard RFC

QoS Best Practices

- On AF queues, DSCP-based WRED should be enabled
- Otherwise, TCP synchronization occurs
- WRED allows the packet to be dropped randomly and DSCP functionality provides packet to be dropped based on their priority

QoS Best Practices

LLQ:

- Limit the sum of all LLQs to 33%
- Use an admission control mechanism
- Do not enable WRED

QoS Best Practices

Multimedia/Data:

- Provision guaranteed bandwidth according to application requirements
- Enable DSCP-based WRED

QoS Best Practices

Control:

- Provision guaranteed bandwidth according to control traffic requirements
- Do not enable WRED

QoS Best Practices

Scavenger:

- Provision with a minimum bandwidth allocation such as 1%
- Do not enable WRED

QoS Best Practices

Default/Best effort:

- Allocate at least 25% for the default/Best effort queue
- Enable WRED

RFC 4594

Application Class	Per-Hop Behavior	Admission Control	Queuing & Dropping	Application Examples
VoIP Telephony	EF	Required	Priority Queue (PQ)	Cisco IP Phones (G.711, G.729)
Broadcast Video	CS5	Required	(Optional) PQ	Cisco IP Video Surveillance / Cisco Enterprise T
Realtime Interactive	CS4	Required	(Optional) PQ	Cisco TelePresence
timedia Conferencing	AF4	Required	BW Queue + DSCP WRED	Cisco Jabber, Cisco WebEx
ultimedia Streaming	AF3	Recommended	BW Queue + DSCP WRED	Cisco Digital Media System (VoDs)
Network Control	CS6		BW Queue	EIGRP, OSPF, BGP, HSRP, IKE
Signaling	CS3		BW Queue	SCCP, SIP, H.323
/ Admin / Mgmt (OAM)	CS2		BW Queue	SNMP, SSH, Syslog
Transactional Data	AF2		BW Queue + DSCP WRED	ERP Apps, CRM Apps, Database Apps
Bulk Data	AF1		BW Queue + DSCP WRED	E-mail, FTP, Backup Apps, Content Distributio
Best Effort	DF		Default Queue + RED	Default Class
Scavenger	CS1		Min BW Queue (Deferential)	YouTube, iTunes, BitTorent, Xbox Live

Voice Best Practices

- Voice traffic should be marked to DSCP EF per the QoS Baseline and RFC 3246 , 4594
- Loss should be no more than 1 %
- One-way Latency (mouth-to-ear) should be no more than 150 ms
- Average one-way Jitter should be targeted under 30 ms
- 21–320 kbps of guaranteed priority bandwidth is required per call (depending on the sampling rate, VoIP codec and Layer 2 media overhead)

- Voice quality is directly affected by all three QoS quality factors: loss, latency and jitter

Video QoS Requirements

In general we are interested in two type of video traffic. Interactive Video and Streaming Video. Interactive Video :

When provisioning for Interactive Video (IP Videoconferencing) traffic, the following guidelines are recommended:

- ➢ Interactive Video traffic should be marked to DSCP AF41; excess Interactive- Video traffic can be marked down by a policer to AF42 or AF43.
- ➢ Loss should be no more than 1 %.
- ➢ One-way Latency should be no more than 150 ms.
- ➢ Jitter should be no more than 30 ms.
- ➢ Overprovision Interactive Video queues by 20% to accommodate bursts

Streaming Video:

Video Best Practices:

- Streaming Video (whether unicast or multicast) should be marked to DSCP CS4 as designated by the QoS Baseline.
- Loss should be no more than 5 %.
- Latency should be no more than 4–5 seconds (depending on video application buffering capabilities).
- There are no significant jitter requirements.
- Guaranteed bandwidth (CBWFQ) requirements depend on the encoding format and rate of the video stream.
- Streaming video is typically unidirectional and, therefore, Branch routers may
- not require provisioning for Streaming Video traffic on their WAN/VPN edges (in the direction of Branch-to-Campus).

➢ **Data Applications QoS Requirements**

❖ Best Effort Data

❖ Bulk Data

❖ Transactional/Interactive Data

Best Effort Data

- The Best Effort class is the default class for all data traffic. An application will be removed from the default class only if it has been selected for preferential or deferential treatment

- Best Effort traffic should be marked to DSCP 0. Adequate bandwidth should be assigned to the Best Effort class as a whole, because the majority of applications will default to this class; reserve at least 25 percent for Best Effort traffic

Bulk Data

- The Bulk Data class is intended for applications that are relatively non- interactive and drop-insensitive and that typically span their operations over a long period of time as background occurrences. Such applications include the following:
- FTP
- E-mail
- Backup operations
- Database synchronizing or replicating operations
- Content distribution

- Any other type of background operation

- Bulk Data traffic should be marked to DSCP AF11; excess Bulk Data traffic can be marked down by a policer to AF12; violating bulk data traffic may be marked down further to AF13 (or dropped)

- Bulk Data traffic should have a moderate bandwidth guarantee, but should be constrained from dominating a link

Transactional/Interactive Data

- The Transactional/Interactive Data class, also referred to simply as Transactional Data, is a combination to two similar types of applications:

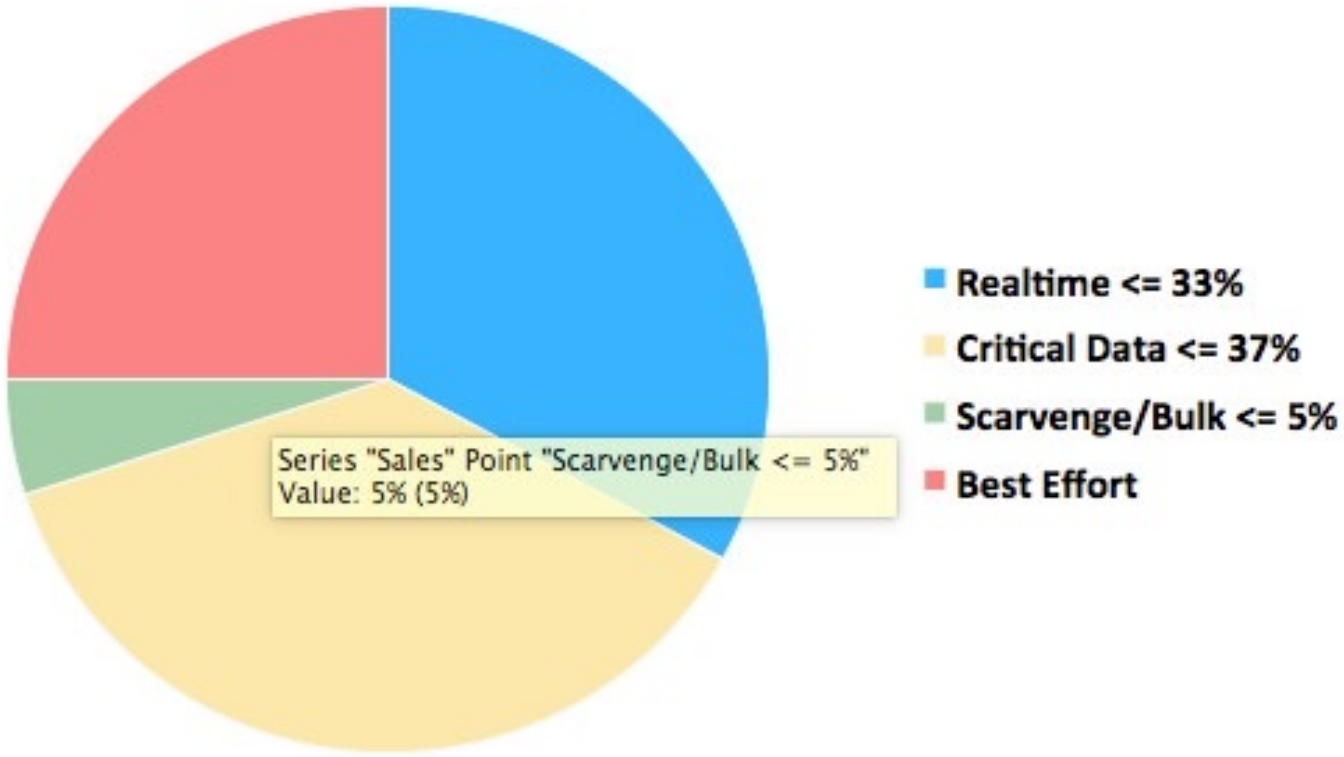
Transactional Data client-server applications and Interactive Messaging applications

- The response time requirement separates Transactional Data client-server applications from generic client-server applications
- For example, with Transactional Data client-server applications such as SAP, PeopleSoft
- Transaction is a foreground operation; the user waits for the operation to complete before proceeding

- E-mail is not considered a Transactional Data client-server application, as most e-mail operations occur in the background and users do not usually notice even several hundred millisecond delays in mail operations

- Transactional Data traffic should be marked to DSCP AF21; excess Transactional Data traffic can be marked down by a policer to AF22; violating Transactional Data traffic can be marked down further to AF23 (or dropped)

- Real time, Best Effort ,Critical Data and Scavenger Queuing Rule – 4 class QoS deployment



QoS Models

Four-Class Model:

- Voice
- Control
- Transactional Data
- Best Effort

QoS Models

Eight-Class Model:

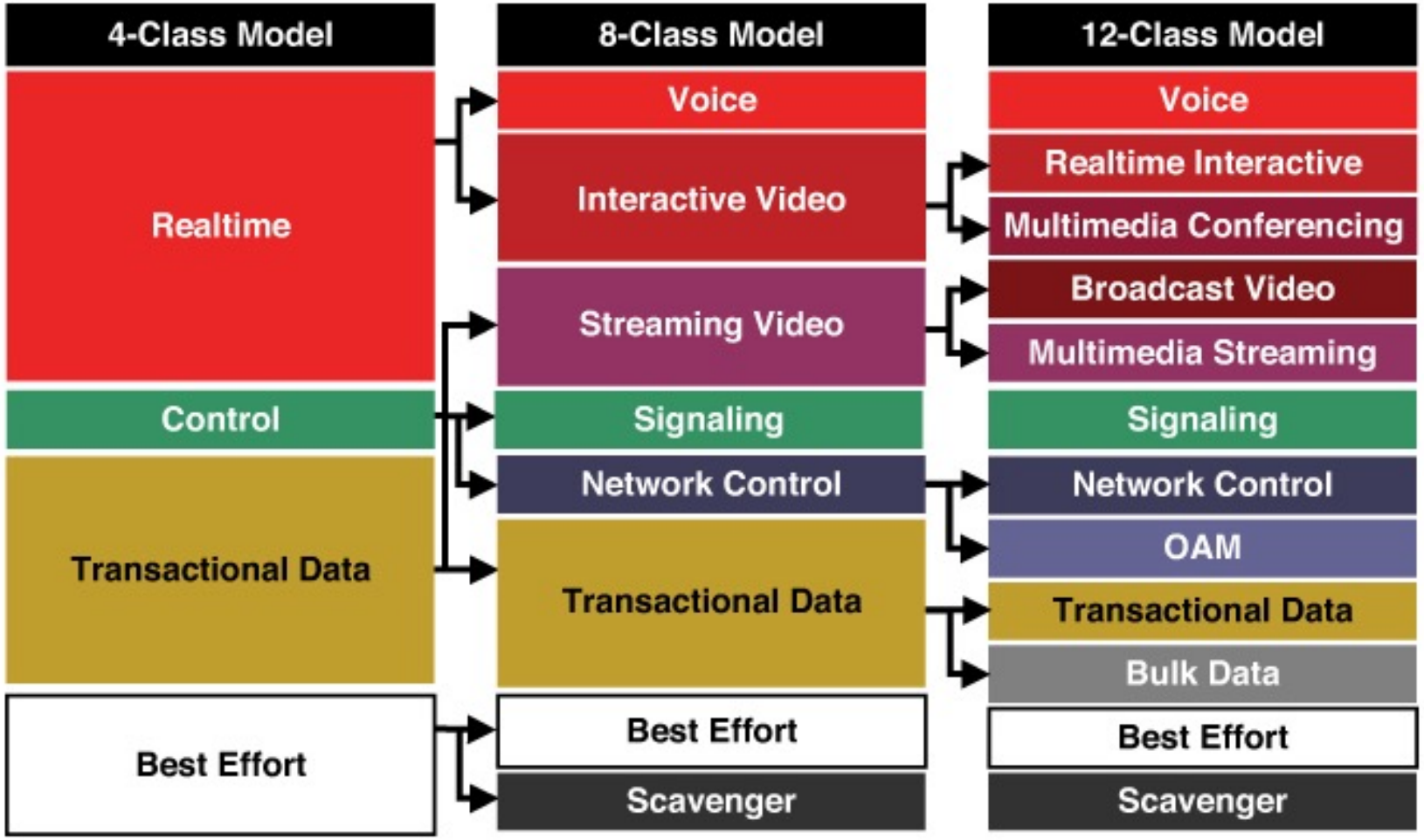
- Voice
- Multimedia-conferencing
- Multimedia-streaming
- Network Control
- Signaling
- Transactional Data
- Best Effort
- Scavenger

QoS Models

Twelve-Class Model:

- Voice
- Broadcast Video
- Real-time interactive
- Multimedia-conferencing
- Multimedia-streaming
- Network Control
- Signaling
- Management/OAM
- Transactional Data
- Bulk Data
- Best Effort
- Scavenger

QoS Models



MPLS Diffserv Tunneling Models

- When an IP Packet enters to the MPLS domains, packet QoS marking can be either carried intact or can be changed in the Core of the network
- MPLS doesn’t define new QoS architecture
- It uses Diffserv architecture that we use in IP networks

MPLS Diffserv Tunneling Models

- MPLS Diffserv is defined in RFC 3270
- RFC 3270 defines three distinct modes of MPLS DiffServ tunneling
- These are Uniform Mode, Short-Pipe and Pipe Models

MPLS Diffserv Tunneling Models

- Tunneling is the ability of QoS to be transparent from one edge of a network to the other edge of the network

MPLS Diffserv Tunneling Models

- A tunnel starts where there is label imposition

- A tunnel ends where there is label disposition; that is, where the label is popped off of the stack and the packet goes out as an MPLS packet with a different PHB layer underneath or as an IP packet with the IP PHB layer

MPLS Diffserv Tunneling Models

- Pipe mode and Short Pipe mode provide QoS transparency

- With QoS transparency, the customer’s IP marking in the IP packet is preserved

MPLS Diffserv Tunneling Models

- The only difference between Pipe mode and Short Pipe mode is which PHB is used on the service provider’s egress edge router

MPLS Diffserv Tunneling Models

- In Pipe mode with an explicit NULL Label, QoS is done on the PE-to-CE link based on the service provider's PHB markings

- The egress LSR still uses the marking that was used by intermediate LSRs

MPLS Diffserv Tunneling Models

- All three tunneling modes affect the behavior of edge and penultimate label switching routers (LSRs) where labels are pushed (put onto packets) and popped (removed from packets)
- They do not affect label swapping at intermediate routers
- A service provider can choose different types of tunneling modes for each customer

MPLS Diffserv Tunneling Models

- **Uniform Mode:**

- In Uniform mode, packets are treated uniformly in the IP and MPLS networks; that is, the IP Precedence value and the MPLS EXP bits always are identical

- Whenever a router changes or recolors the PHB of a packet, that change must be propagated to all encapsulation markings

MPLS Diffserv Tunneling Models

Uniform Mode:

- The propagation is performed by a router only when a PHB is added or exposed due to label imposition or disposition on any router in the packet's path

- For example, if a packet's QoS marking is changed in the MPLS network, the IP QoS marking reflects that change

MPLS Diffserv Tunneling Models

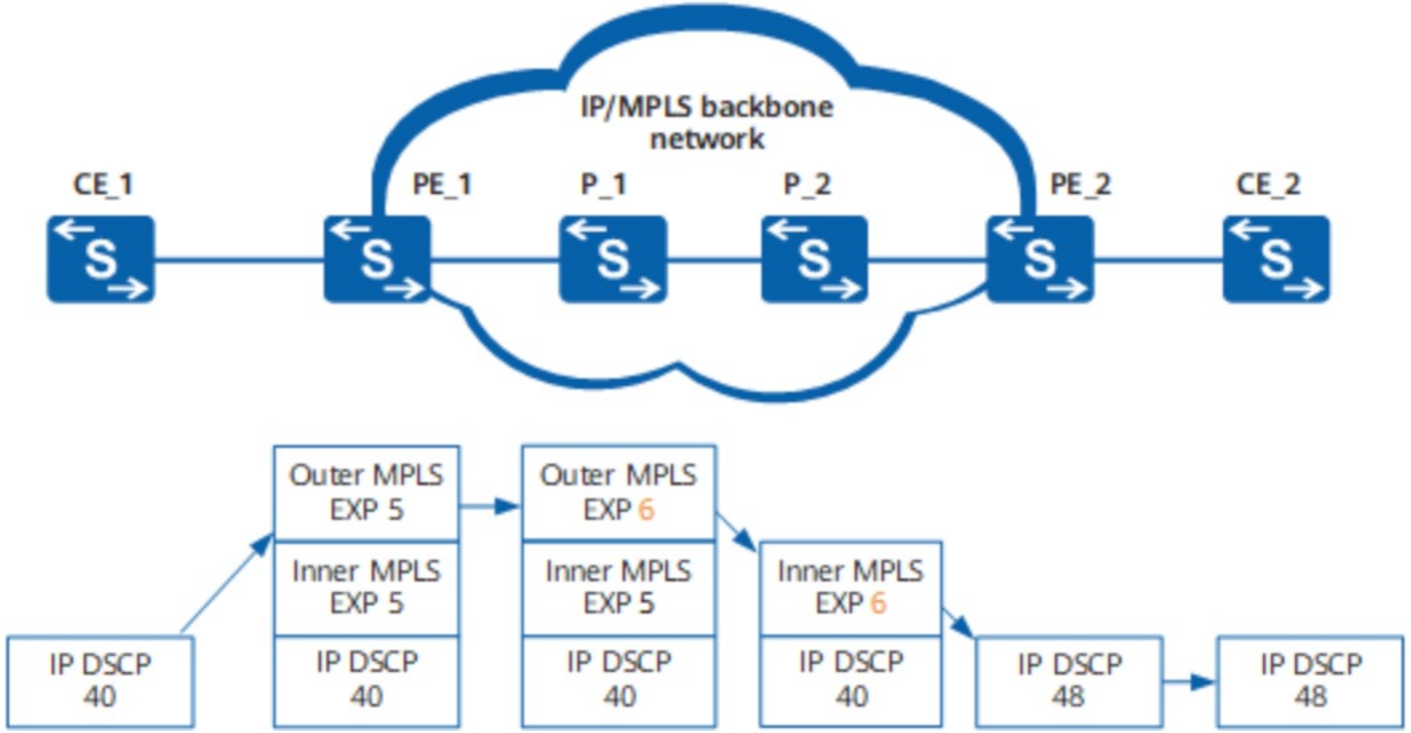
Uniform Mode:

- When an IP packet enters an MPLS network, the ingress node maps the DSCP value of the IP packet to the EXP value of the MPLS packet

- When the MPLS packet leaves the MPLS network, the Egress node maps the EXP value to the DSCP value

MPLS Diffserv Tunneling Models

Uniform Mode:



MPLS Diffserv Tunneling Models

Short-Pipe Mode:

- On the Ingress PE, Service Provider might use EXP value different than customer DSCP value, for example customer might mark its Voice traffic with DSCP 24, this can be mapped to EXP 5, another customer might mark voice traffic with DSCP 46, this can be marked to EXP5 as well

- It is better Service Provider to know the Customer policy

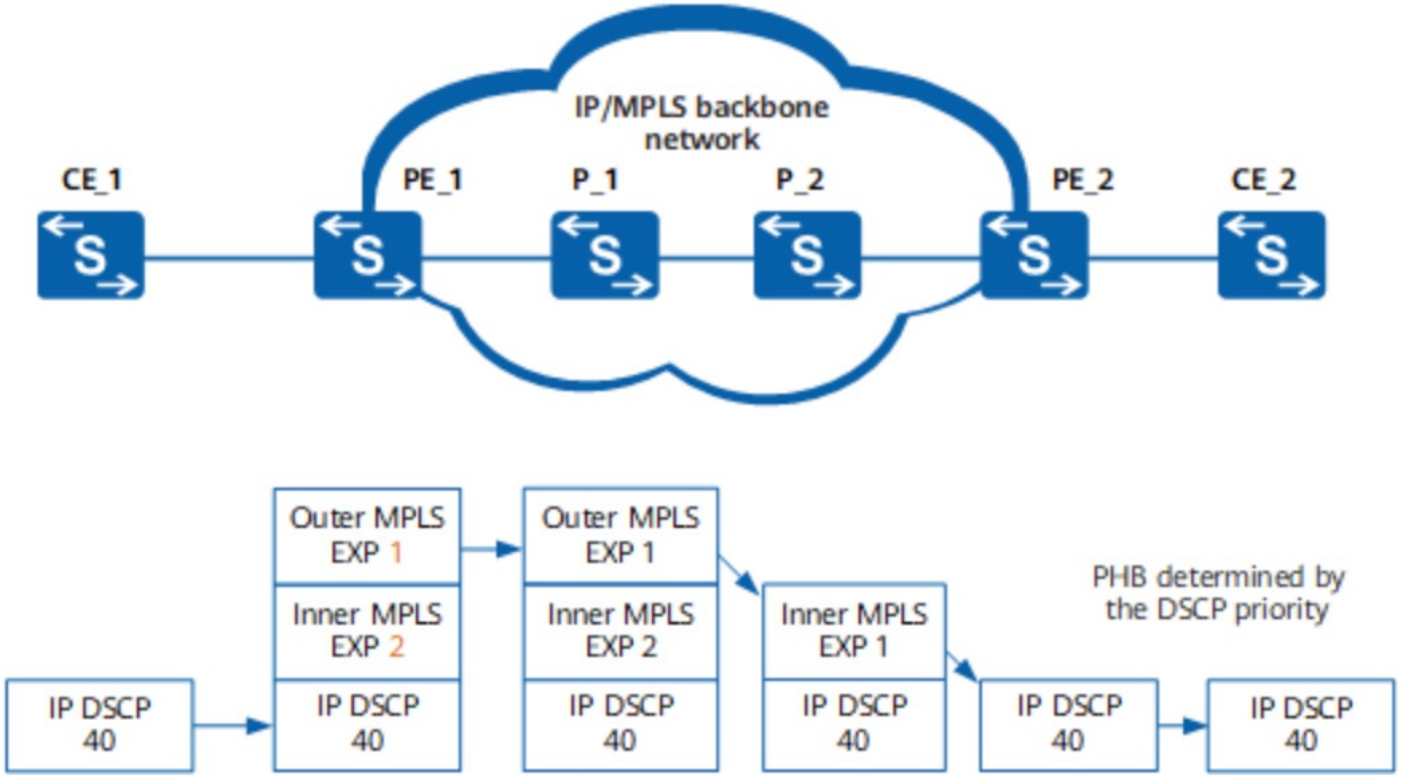
MPLS Diffserv Tunneling Models

Short-Pipe Mode:

- When an IP packet enters and leaves an MPLS network, the packet is processed in the same way as in pipe mode
- On nodes from the ingress node to the penultimate node, the packet is scheduled based on the configured EXP value
- On the egress node, the top label is popped, and then the packet is scheduled based on its DSCP value

MPLS Diffserv Tunneling Models

Short-Pipe Mode:



MPLS Diffserv Tunneling Models

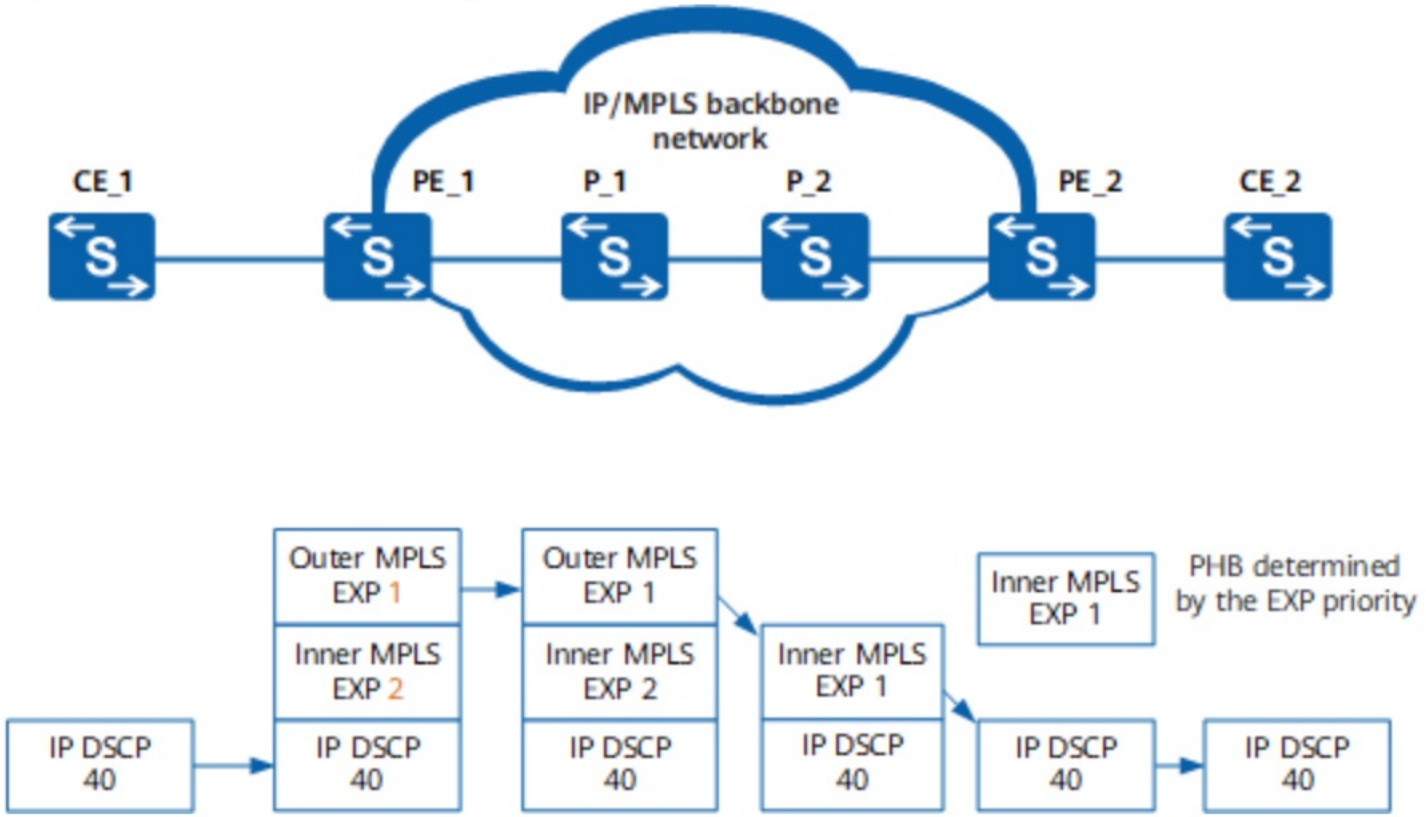
Pipe Mode:

- When an IP packet enters an MPLS network, the ingress node ignores the DSCP value of the IP packet and uses the configured value as the EXP value of the MPLS packet

- When the MPLS packet leaves the MPLS network, the egress node does not modify the original DSCP value. In the MPLS network, the packet is scheduled based on the configured EXP value

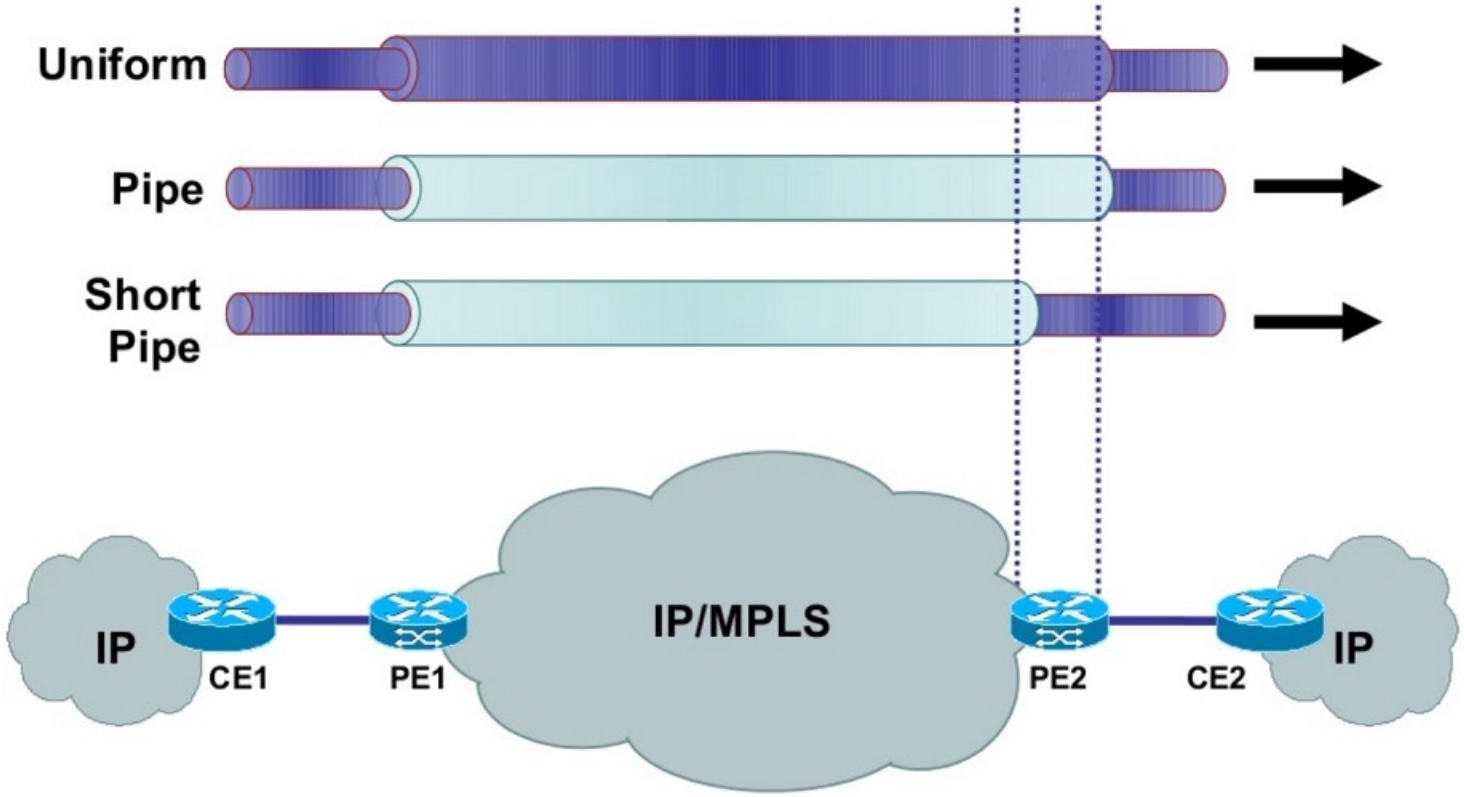
MPLS Diffserv Tunneling Models

Pipe Mode:



MPLS Diffserv Tunneling Models

Summary



MPLS Diffserv Tunneling Models

Summary

- If there are managed customer edge (CE) routers, it is recommended that you use Pipe mode with so that there is service provider PHB on the PE-to-CE link
- If there is no managed CE router, it is recommended that you use Short Pipe mode
- If there are no markings or few markings, customers are likely to use Uniform mode

MPLS Traffic Engineering with Diffserv - DS-TE

Why DS-TE?

- MPLS TE sets up an LSP along links with available resources, thus ensuring bandwidth is always available for a particular flow and avoiding congestion
- Optimization is achieved by allowing LSPs not to follow the shortest paths, if resources are not sufficient along the shortest path
- However, MPLS TE does not take into account QoS (Class of Service CoS or IP DSCP)

MPLS Traffic Engineering with Diffserv - DS-TE

Why DS-TE?

- MPLS DiffServ TE (DS-TE) makes MPLS TE (also called aggregate MPLS TE) QoS aware, allowing resource reservation with QoS granularity
- DS-TE delivers QoS guarantees for sensitive traffic like Voice
- To achieve QoS optimization, traffic engineering is done at per-class level instead of at aggregate (global) level

MPLS Traffic Engineering with Diffserv - DS-TE

- By mapping the traffic from a given DiffServ class-of-service on a separate LSP, it allows this traffic to utilize resources available to the given class on paths that meet constraints which are specific to the given class

MPLS Traffic Engineering with Diffserv - DS-TE

- The fundamental requirement of DS-TE is to be able to enforce different bandwidth constraints for different sets of DS-TE tunnels

MPLS Traffic Engineering with Diffserv - DS-TE

- MPLS DS-TE combines MPLS TE and Diff-Serv to provide QoS guarantee
- A disadvantage of the basic MPLS-TE model is that it is not aware of the different Diffserv classes, operating at an aggregate level across all of them
- In the case of Diffserv aware MPLS-TE, it refines the MPLS-TE model by allowing bandwidth reservations to be carried out on a per-class basis
- MPLS DS-TE uses the Class Type (CT) so that MPLS TE can allocate resources based on the type of traffic and provide differentiated services

MPLS Traffic Engineering with Diffserv - DS-TE

- DS-TE must be able to keep track of the available bandwidth for each classes of traffic
- For the purpose of keeping track of available bandwidth for each type of traffic, Class Types are defined
- There are no rules that govern what traffic maps to which CT. A given DS-TE tunnel belongs to the same CT on all links

MPLS Traffic Engineering with Diffserv - DS-TE

- Eight CTs are defined CT0 through CT7
- A DS-TE LSP can only carry traffic from one CT. Suppose a network supports voice and data traffic with voice being EF PHB (EF queue) and data being best-effort (BE queue), CT1 can be mapped to EF queue while CT0 can be mapped to BE queue
- Separate TE LSPs are established with separate bandwidth requirements from CT0 and from CT1

MPLS Traffic Engineering with Diffserv - DS-TE

- **CSPF in DS-TE**

- In aggregate MPLS TE, CSPF computes a path based on user-defined constraints such as bandwidth and link attributes (setup and hold priority)

- DS-TE adds available bandwidth at each of the 8 CTs as a constraint that can be applied to a path

- Therefore, CSPF is enhanced to take into account a CT-specific bandwidth at a given priority as a constraint when computing a path

MPLS Traffic Engineering with Diffserv - DS-TE

- **TE Class in DS-TE**
- This means, ideally, the IGP must carry bandwidth information of 8 CTs at 8 priority level (i.e. 64 values) for each link in LSAs
- However, only 8 values are advertised
- A TE class is defined which is a combination of CT and setup priority
- The IGPs advertise the available bandwidth for each of the TE classes defined

MPLS Traffic Engineering with Diffserv - DS-TE

TE Class in DS-TE

- DS-TE supports a maximum of 8 TE classes, TE0 through TE7, which are chosen from a possible 64 different CT-priority combination through configuration

- The combinations chosen depends on the classes and priorities the network support

MPLS Traffic Engineering with Diffserv - DS-TE

- **TE Class in DS-TE**
- DS-TE specifies a preemption priority for each CT
- A TE-class is a combination of a CT and a priority
- Each of eight CTs can be combined with any of eight priorities, so there are 64 TE-classes
- On the device, eight TE-classes can be configured manually

MPLS Traffic Engineering with Diffserv - DS-TE

- **TE Class in DS-TE**
- A TE-class mapping table consists of a set of TE-classes.
- You are advised to configure all the LSRs with the same TE-class mapping table over an MPLS network
- The device has the default TE-class mapping table

MPLS Traffic Engineering with Diffserv - DS-TE

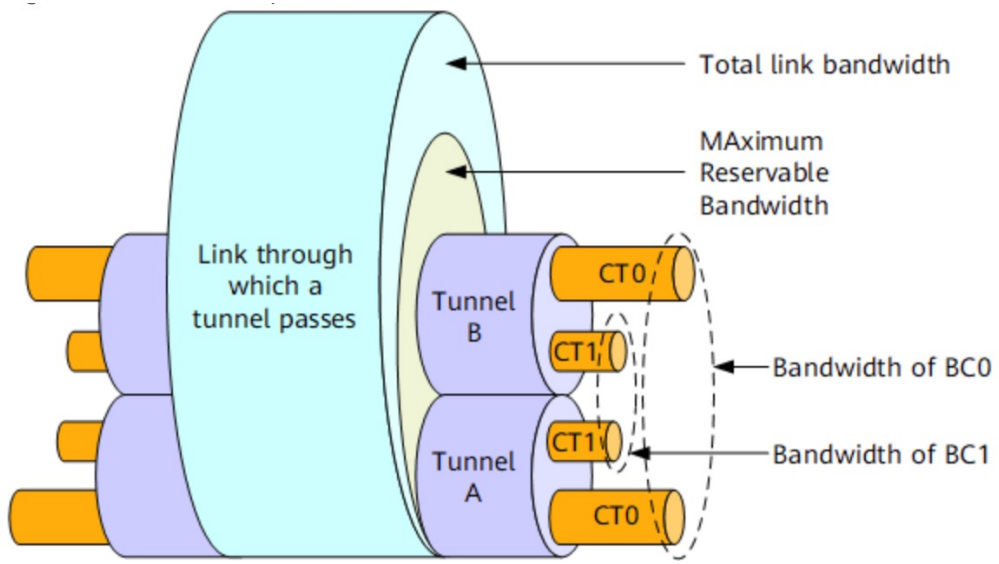
- **TE Class in DS-TE**

TE-Class	CT	Priority
TE-Class[0]	0	0
TE-Class[1]	1	0
TE-Class[2]	2	0
TE-Class[3]	3	0
TE-Class[4]	0	7
TE-Class[5]	1	7
TE-Class[6]	2	7
TE-Class[7]	3	7

MPLS Traffic Engineering with Diffserv - DS-TE

Bandwidth in DS-TE

- MPLS DS-TE involves the following types of bandwidth:
- Total link bandwidth
  - Bandwidth of a physical link
- Maximum reservable bandwidth
  - Maximum bandwidth that a link can reserve for an MPLS TE tunnel. The maximum reservable bandwidth must be lower than or equal to the total link bandwidth
- CT bandwidth: is the bandwidth of service traffic of each type on each DS-TE tunnel.
- BC bandwidth: is the bandwidth reserved for all CTs along a link



MPLS Traffic Engineering with Diffserv - DS-TE

Bandwidth Constraints Model

- Bandwidth constraint model defines the maximum number of bandwidth constraints and which CTs each bandwidth constraint applies to and how to use BC bandwidth

- The IETF defines the following bandwidth constraints models : RDM – Russian Dolls Model and MAM – Maximum Allocation Model

MPLS Traffic Engineering with Diffserv - DS-TE

Bandwidth Constraints Model

- **Russian Dolls Model (RDM):**

- CTs can share bandwidth

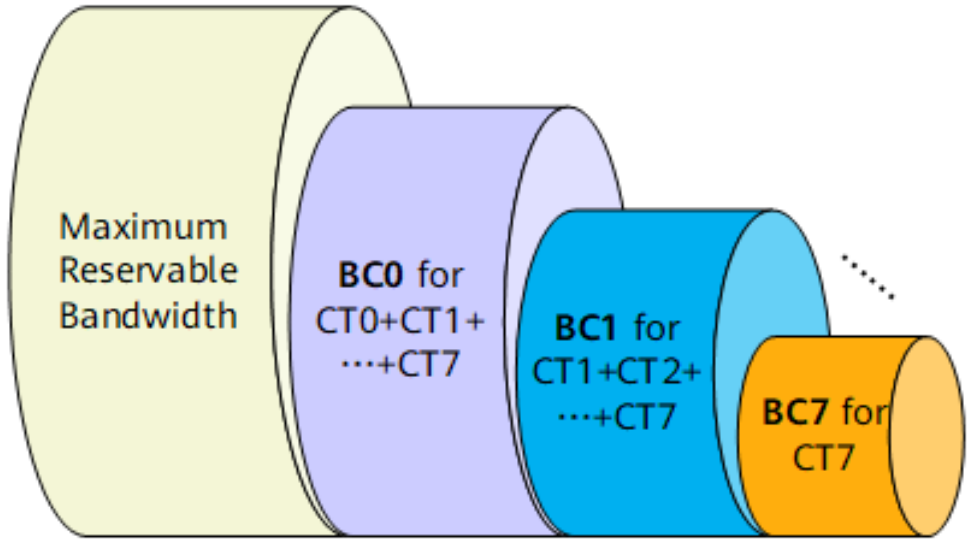
- The bandwidth of BC0 is less than or equal to the maximum reservable bandwidth of a link

MPLS Traffic Engineering with Diffserv - DS-TE

Bandwidth Constraints Model

- **Russian Dolls Model (RDM):**

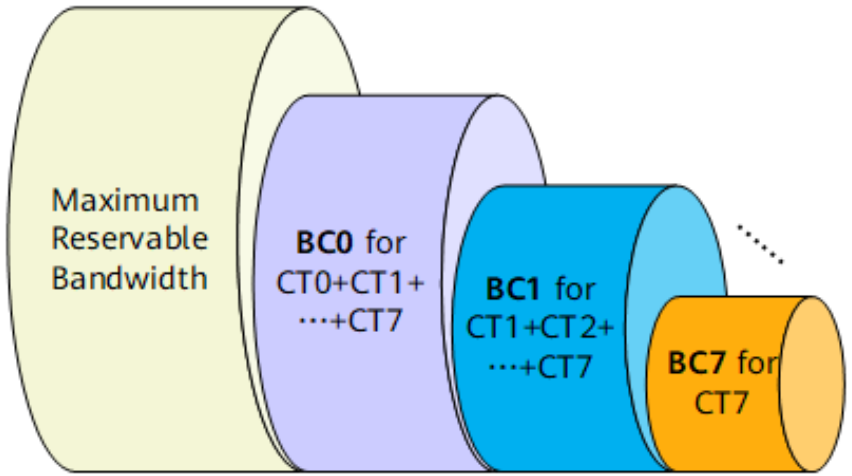
- Total bandwidth of all LSPs from CT0, CT1, ... CT7 ≤ Bandwidth of BC0 ≤ Maximum reservable bandwidth
- Total bandwidth of all LSPs from CT1, CT2, and CT7 ≤ Bandwidth of BC1
- ...
- Total bandwidth of all LSPs from CT7 ≤ Bandwidth of BC7



MPLS Traffic Engineering with Diffserv - DS-TE

Russian Dolls Model (RDM):

- For example, the bandwidth of a link is 100 Mbit/s, RDM is used, and three CTs are supported, that is, CT0, CT1, and CT2
- CT0, CT1, and CT2 transmit BE, AF, and EF traffic respectively
- The bandwidths of BC0, BC1, and BC2 are 100 Mbit/s, 50 Mbit/s, and 20 Mbit/s respectively
- The total bandwidth of all LSPs transmitting EF traffic cannot be larger than 20 Mbit/s; the total bandwidth of all LSPs transmitting AF and EF traffic cannot be larger than 50 Mbit/s; the total bandwidth of all LSPs cannot be larger than 100 Mbit/s



MPLS Traffic Engineering with Diffserv - DS-TE

Bandwidth Constraints Model

- **Russian Dolls Model (RDM):**
- The RDM allows bandwidth preemption between CTs
- CT0 traffic with lower setup priority can preempt CT0 traffic with higher setup priority

MPLS Traffic Engineering with Diffserv - DS-TE

Bandwidth Constraints Model

- **Maximum Allocation Model:**

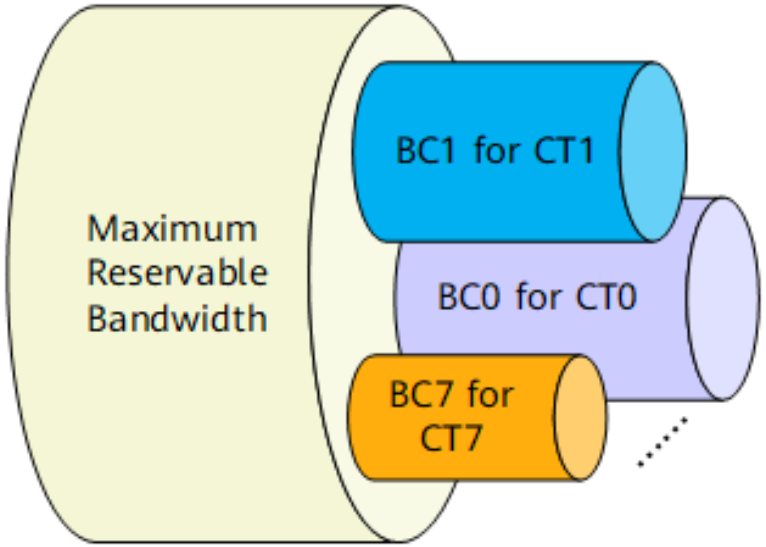
- One BC is mapped to one CT, and CTs cannot share bandwidth

- The total bandwidth of BCs cannot be larger than the maximum reservable bandwidth

MPLS Traffic Engineering with Diffserv - DS-TE

Russian Dolls Model (RDM):

- For example, the bandwidth of a link is 100 Mbit/s, MAM is used, and three CTs are supported, that is, CT0, CT1, and CT2
- BC0 is 20 Mbit/s and transmits CT0 traffic (for example, BE traffic); BC1 is 50 Mbit/s and transmits CT1 traffic (for example, AF traffic); BC2 is 30 Mbit/s and transmits CT2 traffic (for example, EF traffic)
- The total bandwidth of all LSPs transmitting BE traffic cannot be larger than 20 Mbit/s; the total bandwidth of all LSPs transmitting AF traffic cannot be larger than 50 Mbit/s; the total bandwidth of all LSPs transmitting EF traffic cannot be larger than 30 Mbit/s



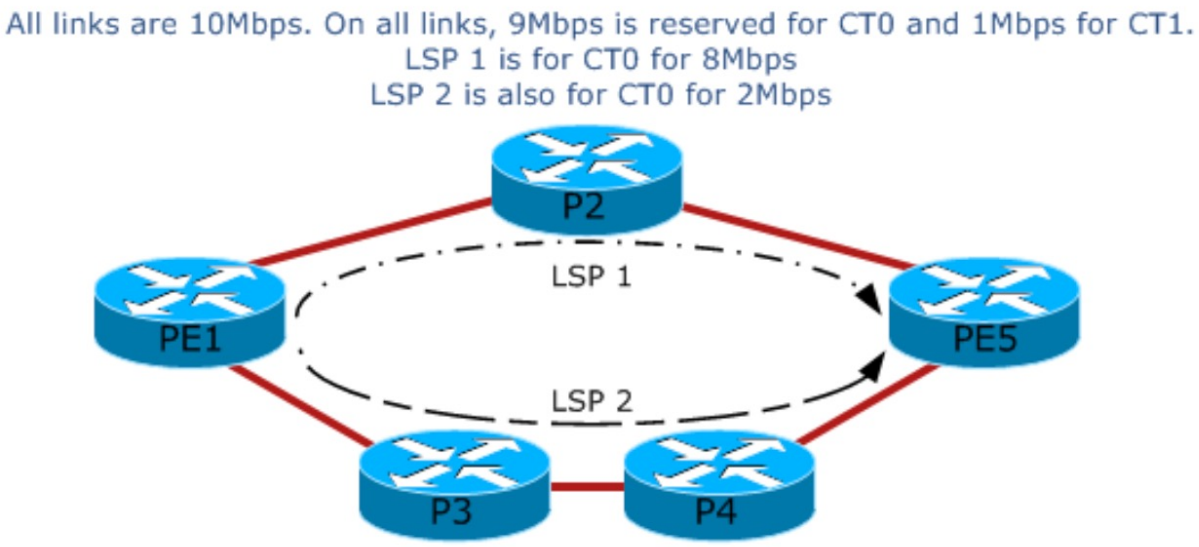
MPLS Traffic Engineering with Diffserv - DS-TE

DS-TE Example - MAM Model

All links are 10Mbps and hence the maximum reservable bandwidth is 10Mbps

Suppose 9Mbps is reserved for CT0 and 1 Mbps for CT1

If LSP1 is setup for 8Mbps from CT0, it follows the shortest path **PE1-P2-PE5**

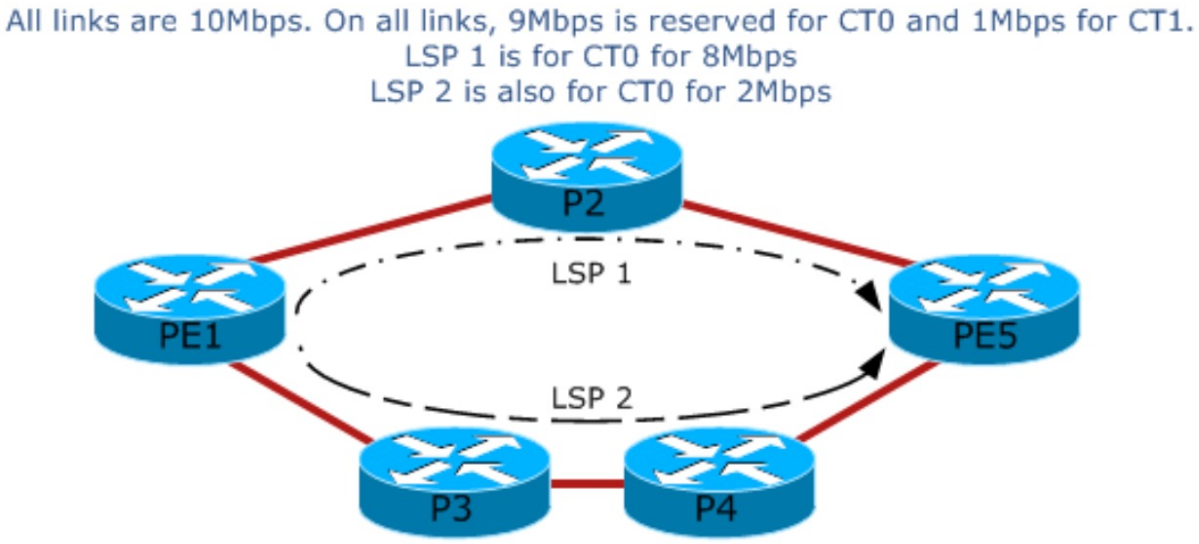


MPLS Traffic Engineering with Diffserv - DS-TE

DS-TE Example - MAM Model

Now, if another LSP2 is to be setup from CT0 for 2Mbps, it does not follow the shortest path as 1Mbps of the remaining bandwidth is reserved for CT1

So, LSP2 is forced to follow the non-optimal path **PE1-P3-P4-PE5**



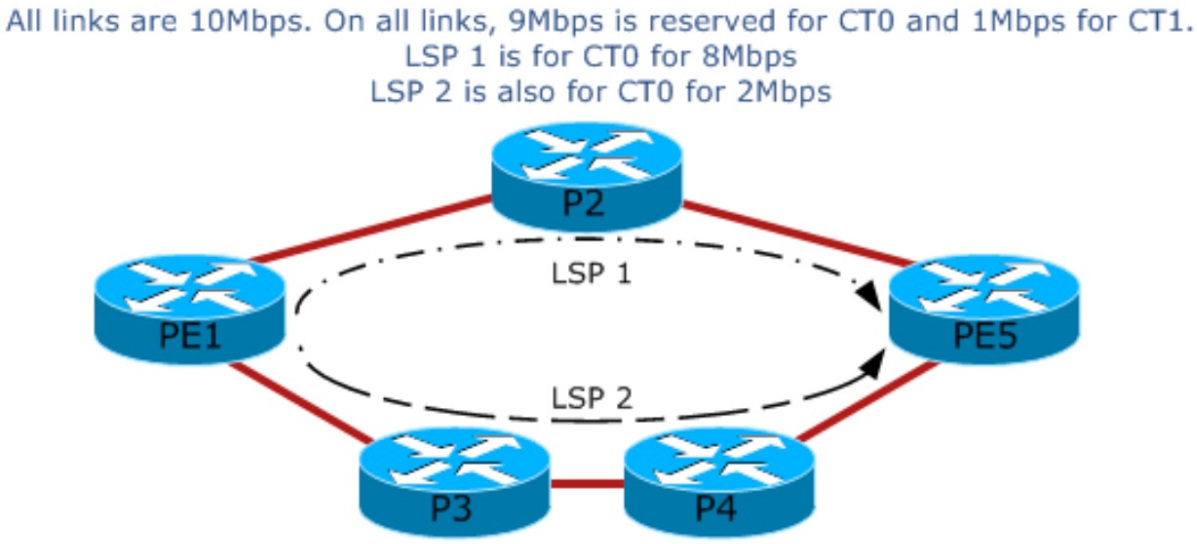
MPLS Traffic Engineering with Diffserv - DS-TE

DS-TE Example - MAM Model

Thus, each CT will always receive their bandwidth without the need for pre-emption

An LSP3 of 1Mbps from CT1 will follow the shortest path, if setup

”The available bandwidth in MAM model is accounted in a similar way, as for aggregate TE, except that it is done on a per-CT basis”



MPLS Traffic Engineering with Diffserv - DS-TE

DS-TE Example - RDM Model

- This model improves bandwidth efficiency over MAM model by allowing CTs to share bandwidth
- In this model, CT7 has strictest QoS requirements and CT0 has best-effort QoS requirements
- BC7 has a fixed percentage of link bandwidth that is reserved for CT7 only
- BC6 accommodates traffic from CT7 and CT6, BC5 accommodates CT7, CT6 and CT5, and so on
- So, BC0 represents the entire link bandwidth and is shared among all CTs

MPLS Traffic Engineering with Diffserv - DS-TE

DS-TE Example - RDM Model

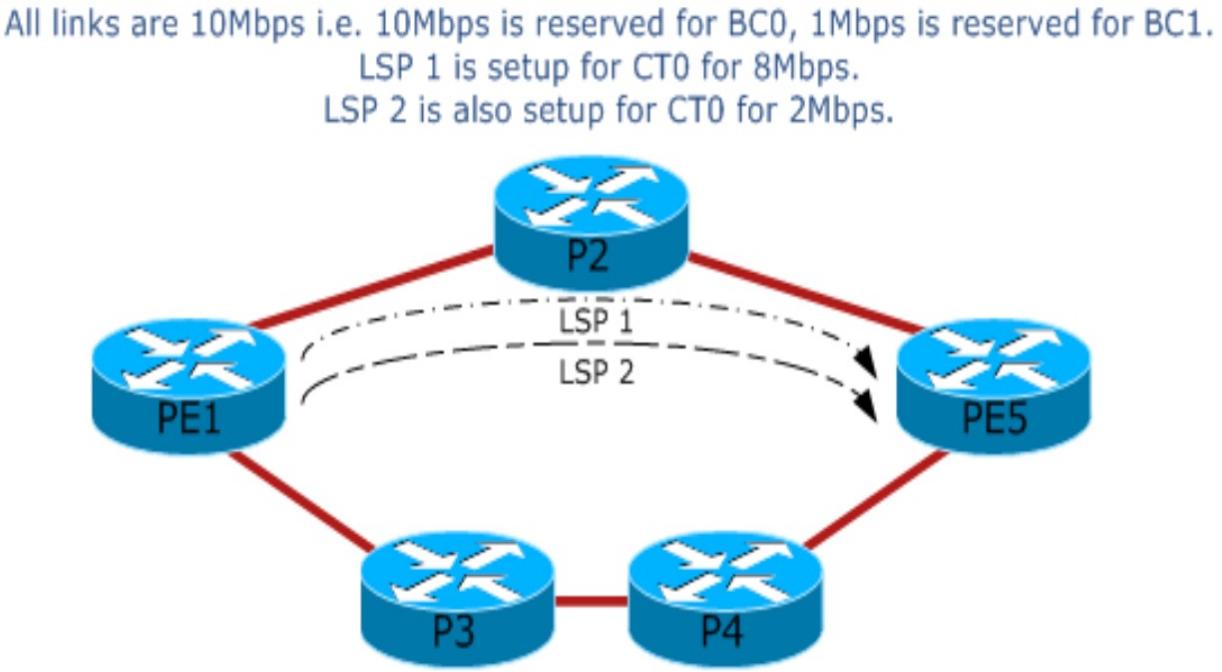
The total bandwidth available on each link is 10Mbps

In other words, BC0 is allocated 10Mbps and BC1 is, suppose, allocated 1Mbps

If LSP1 is setup for 8Mbps from CT0, it follows the shortest path **PE1-P2-PE5**

The remaining 2Mbps can be used for CT0 or CT1

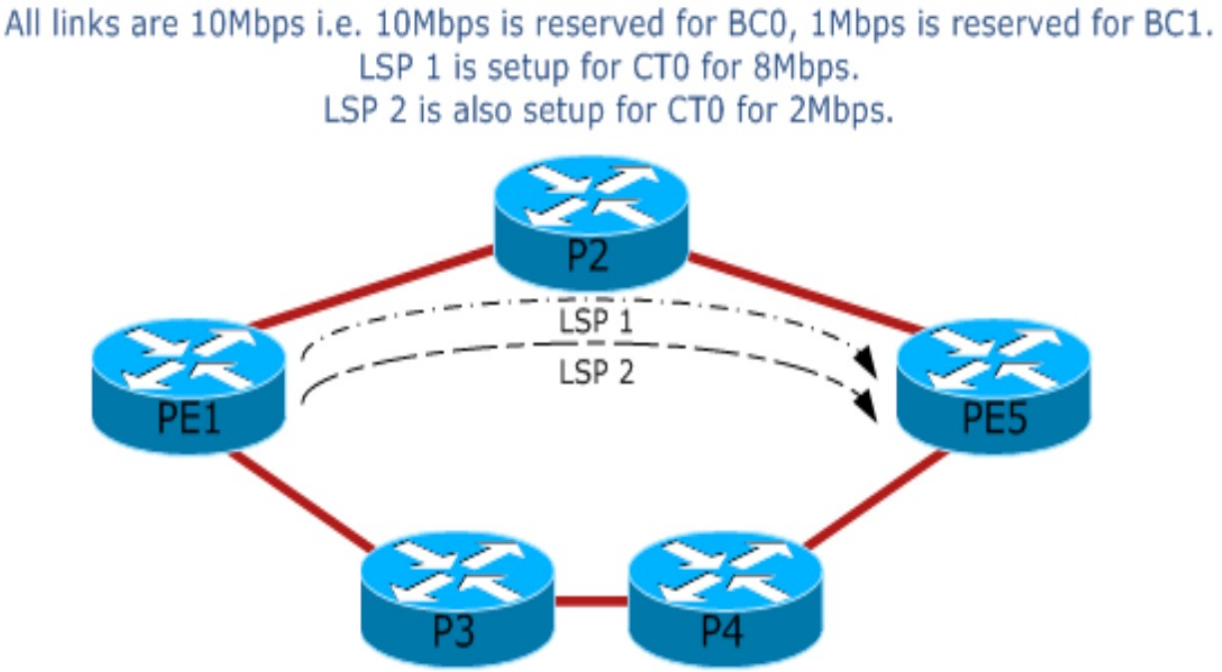
If another LSP2 is to be setup for 2Mbps from CT0, it will follow the shortest path **PE1-P2-PE5** as well



MPLS Traffic Engineering with Diffserv - DS-TE

DS-TE Example - RDM Model

- Now if an LSP3 is to be setup for 1Mbps from CT1, it will pre-empt one of the CT0 LSPs on the shortest path, otherwise bandwidth is not guaranteed
-
- The pre-emption is based on setup priority usually (hold priority can be used too)
- 0 is the best priority, while 7 is the worst priority
- LSP1 will now re-establish through the non-optimal path **PE1-P3-P4-PE5**



MPLS Traffic Engineering with Diffserv - DS-TE

RDM vs MAM Model Comparison

Item	RDM	MAM/Extended-MAM
BC-CT mapping	Maps one BC to one or more CTs.	Maps one BC to one CT, which is easy for bandwidth management.
Bandwidth preemption	Is unable to divide CT bandwidth and requires preemption to provide sufficient bandwidth for CTs.	Divides CT bandwidth and provides sufficient bandwidth for CTs.
Bandwidth use efficiency	Efficiently uses bandwidth.	Wastes bandwidth.

QOS Study Resources

- **Books :**

- http://www.amazon.com/End---End-QoS-Network-Design/dp/1587143690/ref=sr_1_1?ie=UTF8&qid=1436564258&sr=8- 1&keywords=end+to+end+qos+network+design

- **Videos :**

- **Ciscolive Session – BRKCRS -2501**
- **https://www.youtube.com/watch?v=6UJZBeK_JCs**

- **Articles :**

- http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoS-SRND- Book/QoSIntro.html

- http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/qosmrn.pdf

- http://orhanergun.net/2015/06/do-you-really-need-quality-of-service/

- http://d2zmdbbm9feqrf.cloudfront.net/2013/usa/pdf/BRKCRS-2501.pdf

- https://ripe65.ripe.net/presentations/67-2012-09-25-qos.pdf