

# Chapter 27

## Network Performance (QoS and DiffServ)

# Topics Covered

- ◆ Introduction
- ◆ Measures of Performance
- ◆ Latency or Delay
- ◆ Throughput, Capacity, and Goodput
- ◆ Understanding Throughput and Delay
- ◆ Jitter
- ◆ The Relationship Between Delay and Throughput
- ◆ Measuring Delay, Throughput, and Jitter
- ◆ Passive Measurement, Small Packets, and NetFlow
- ◆ Quality of Service (QoS)
- ◆ Fine-Grain and Coarse-Grain QoS
- ◆ Implementation of QoS
- ◆ Internet QoS Technologies

# Introduction

- ◆ **This chapter**
  - ◆ **discusses the topic of network performance**
  - ◆ **discusses quantitative measures of networks**
  - ◆ **explains how protocols and packet forwarding technologies can implement mechanisms**
    - ◆ that provide priority for some traffic

# Measures of Performance

- ◆ Term **speed** to describe network performance
  - ◆ we may refer to networks as **low-speed** or **high-speed** networks
- ◆ Such definitions are inadequate
  - ◆ because network technologies change so rapidly
  - ◆ that a network classified as “high speed” can become medium or low speed in as little as three or four years
- ◆ We should use **quantitative** measures
  - ◆ to specify network performance precisely
  - ◆ they make it possible to compare the exact features of two networks and to build mechanisms that provide higher priority for some traffic

# Measures of Performance

Measure	Description
Latency (delay)	The time required to transfer data across a network
Throughput (capacity)	The amount of data that can be transferred per unit time
Jitter (variability)	The changes in delay that occur and the duration of the changes

# Latency or Delay

- ◆ **Latency**
  - ◆ specifies how long it takes for data to travel across a network from one computer to another
  - ◆ it is measured in fractions of seconds
- ◆ **Delays**
  - ◆ across the Internet depend on the underlying infrastructure as well as the location of the specific pair of computers that communicate
- ◆ Users care about the total delay of a network
- ◆ Engineers usually report both the maximum and average delays
- ◆ Also it's custom to divide a delay into several constituent parts

# Components of Delay

Type	Explanation
Propagation Delay	The time required for a signal to travel across a transmission medium
Access Delay	The time needed to obtain access to a transmission medium (e.g., a cable)
Switching Delay	The time required to forward a packet
Queuing Delay	The time a packet spends in the memory of a switch or router waiting to be selected for transmission
Server Delay	The time required for a server to respond to a request and send a response

# Propagation Delay

- ◆ A signal requires a small amount of time to travel across a transmission medium
  - ◆ propagation delays are proportional to the distance spanned
- ◆ Such delays seem irrelevant to a human
  - ◆ But a modern computer can execute over one hundred thousand instructions in a millisecond
- ◆ A millisecond delay is significant
  - ◆ when a set of computers need to coordinate work
- ◆ Consider the following:
  - ◆ A network that uses a GEO satellite has much higher delay
  - ◆ Even at the speed of light
    - ◆ it takes hundreds of *ms* for a bit to travel to the satellite and back to earth
    - ◆ For LEO satellites the round trip delay (four hops) will still be around 20 ms

# Access Delay

- ◆ Many networks use **shared media**
- ◆ Set of computers that share a medium must **contend for access**
  - ◆ Such delays are known as **access delays**
- ◆ Access delays depend on
  - ◆ the number of stations that contend for access and
  - ◆ the amount of traffic each station sends

# Switching Delay

- ◆ The computation often involves table lookup
  - ◆ which means memory access
- ◆ Additional time may be needed to send the packet over an internal communication mechanism
- ◆ The time required to compute a next hop and begin transmission is known as a **switching delay**

# Queuing Delay

- ◆ The store-and-forward paradigm used in packet switching means that a device such as a router
  - ◆ collects the bits of a packet
  - ◆ places them in memory
  - ◆ chooses a next-hop
  - ◆ and then waits until the packet can be sent
- ◆ Such delays are known as **queuing delays**
- ◆ In the simplest case
  - ◆ a packet is placed in a FIFO output queue
  - ◆ the packet only needs to wait until packets that arrived earlier are sent
- ◆ In more complex systems
  - ◆ a selection algorithm that gives priority to some packets

# Server Delay

- ◆ **The time required for a server to process the following constitute a significant part of overall delay**
  - ◆ to examine a request and
  - ◆ to compute and send a response
- ◆ **Servers queue incoming requests**
  - ◆ means that server delay is variable and depends on the current load
- ◆ **In many cases**
  - ◆ a user's perception of Internet delay arises from server delay rather than network delays

# Throughput and Capacity

- ◆ **Capacity**
  - ◆ often expressed as the **maximum throughput**
- ◆ **Throughput**
  - ◆ a measure of the rate at which data can be sent through the network
  - ◆ specified in bits per second (bps)
- ◆ Throughput can be measured in several ways
- ◆ We should specify exactly what has been **measured**
- ◆ There are several possibilities:
  - ◆ Capacity of a single channel
  - ◆ Aggregate capacity of all channels
  - ◆ Theoretical capacity of the underlying hardware
  - ◆ Effective data rate achieved by an application

# Throughput, Capacity, and Goodput

- ◆ **Hardware capacity**
  - ◆ often cited as an approximation of the potential throughput
- ◆ The capacity gives an upper bound on performance
  - ◆ it is impossible for a user to send data faster than the rate at which the hardware can transfer bits
- ◆ Users typically assess the **effective data rate**
  - ◆ that an application achieves by measuring the amount of data transferred per unit time
  - ◆ the term **goodput** is sometimes used to describe this measure
- ◆ The goodput rate is less than the capacity of the hardware
  - ◆ because protocols impose overhead (see next page)

# Effective Throughput

- ◆ **Some network capacity is not available to user data because protocols:**
  - ◆ **Send packet headers, trailers, and control information**
  - ◆ **Impose a limit on the window size (receive buffer)**
  - ◆ **Use protocols to resolve names and addresses**
  - ◆ **Use a handshake to initiate and terminate communication**
  - ◆ **Reduce the transmission rate when congestion is detected**
  - ◆ **Retransmit lost packets**
- ◆ **The disadvantage of using goodput as a measure**
  - ◆ **the amount of overhead depends on the protocol stack being used**
- ◆ **Goodput depends on the application protocol**
  - ◆ **In addition to Transport, Internet, and Layer 2 protocols**

# Understanding Throughput and Delay

- ◆ The terminology that professionals use to describe network throughput or network capacity can be confusing
  - ◆ often they use the terms **bandwidth** and **speed** as synonyms for **throughput**
- ◆ Throughput is a measure of capacity
  - ◆ not speed
- ◆ As an analogy
  - ◆ Imagine a network to be a road between two locations and packets traveling across the network to be cars traveling down the road
    - ◆ throughput determines how many cars can enter the road each second
    - ◆ and the propagation delay determines how long it takes a single car to travel the road from one town to another

# Understanding Throughput and Delay

- ◆ **Networking professionals have an interesting aphorism:**
  - ◆ You can always buy more throughput
  - ◆ But you cannot buy lower delay
- ◆ **The analogy to a road helps explain:**
  - ◆ adding more lanes to a road will increase the number of cars that can enter the road per unit of time
    - ◆ but will not decrease the total time required to traverse the road
- ◆ **Networks follow the same pattern:**
  - ◆ adding more parallel transmission paths
    - ◆ will increase the throughput of the network
    - ◆ but the propagation delay will not decrease

# Jitter

- ◆ Another measure of networks is used for the transmission of real-time voice and video
  - ◆ It is known as a network's **jitter**
  - ◆ It assesses the **variance in delay**
- ◆ Two networks can have the same average delay
  - ◆ but different values of jitter
- ◆ If all packets that traverse a given network have exactly the same delay, **D**
  - ◆ the network has no jitter
- ◆ If packets alternate between a delay of  **$D+\delta$**  and  **$D-\delta$** 
  - ◆ the network has the same average delay, but has a nonzero jitter

# Jitter

- ◆ Why is jitter important?
  - ◆ consider sending voice over a network
- ◆ If the network has **zero jitter** (i.e., each packet takes exactly the same time to transit the network)
  - ◆ the audio output will exactly match the original input
- ◆ Otherwise, if the network has **non-zero jitter**
  - ◆ the output will be flawed
- ◆ There are two general approaches to handling jitter:
  - ◆ Design an isochronous network with no jitter
  - ◆ Use a protocol that compensates for jitter

# Jitter

- ◆ A traditional telephone system uses the first approach:
  - ◆ The phone system implements an **isochronous network**
- ◆ Telephone system guarantees the delay along all paths is the same
  - ◆ Thus, if digitized data from a phone call is transmitted over two paths
    - ◆ the hardware is configured so that both paths have exactly the same delay
- ◆ Transmission of voice or video over the Internet takes the second approach
- ◆ In Internet, the underlying network may have substantial jitter
  - ◆ voice and video applications rely on real-time protocols to compensate for jitter
- ◆ Using real-time protocols is much less expensive than building an isochronous network

# The Relationship Between Delay and Throughput

- ◆ In theory
  - ◆ the delay and throughput of a network are independent
- ◆ In practice
  - ◆ however, they are often related
- ◆ Think of the road analogy discussed before
- ◆ If a router has a queue of packets waiting when a new packet arrives
  - ◆ the new packet will be placed on the tail of the queue
  - ◆ and will need to wait while the router forwards the previous packets
- ◆ If congestion occurs
  - ◆ the packets will experience longer delays than data entering an idle network

# Utilization As an Estimate of Delay

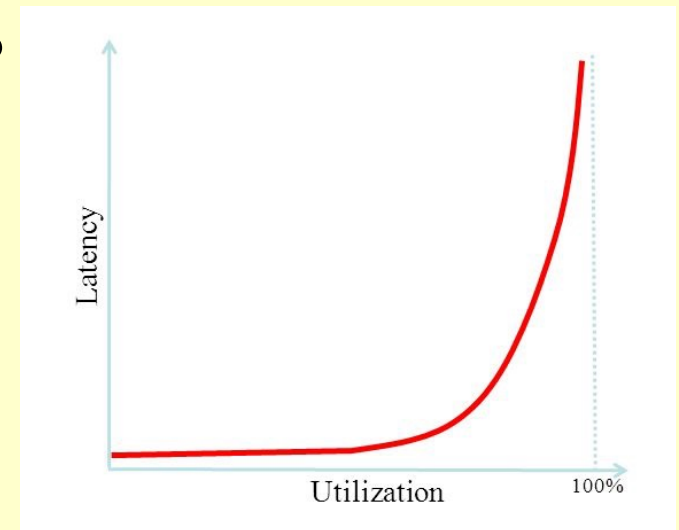
- ◆ In many cases, the expected delay can be estimated from the current percentage of the network capacity being used
  - ◆ If  $D_0$  denotes the delay when a network is idle, and  $U$  is a value between 0 and 1 that denotes the current utilization the effective delay,  $D$ , is given by a simple formula:

$$D = \frac{D_0}{(1 - U)}$$

- ◆ Formula only provides an estimate of effective delay
- ◆ But we can conclude:
  - ◆ Throughput and delay are not completely independent
  - ◆ As traffic in a computer network increases, delays increase
    - ◆ a network that operates at close to 100% of its throughput capacity experiences severe delays

# Utilization As an Estimate of Delay

- ◆ High **utilization** can produce disastrous delay
- ◆ Most managers work to keep utilization low
- ◆ They measure the traffic on each network constantly
  - ◆ When the average (or peak) utilization begins to climb above a pre-set threshold
    - ◆ the manager increases the capacity of the network
- ◆ How high should the utilization threshold be?
  - ◆ There is no simple answer
    - ◆ many managers choose a conservative value
- ◆ For example
  - ◆ major ISP's that runs a large backbone network may keep utilization on all its digital circuits under 50%



# Delay-Throughput Product

- ◆ Once a network's delay and throughput are known
  - ◆ it is possible to compute the **delay-throughput product**
  - ◆ the delay-throughput product is often called **bandwidth-delay product (DBP)**
- ◆ You can again think of it as road analogy
- ◆ In terms of networks, the number of bits traveling through a network at any time is given by:

$$\text{Bits present in a network} = D \times T$$

where  $D$  is the delay measured in seconds

$T$  is the throughput measured in bits per second

- ◆ The product of delay and throughput measures the volume of data that can be present on the network
- ◆ It is very important in a Window based protocol such as TCP to know how big receiver buffer you need to allocate to efficiently use the network capacity

# Measuring Throughput, and Jitter

## ◆ To assess throughput

- ◆ a sender transfers a large volume of data

- ◆ a receiver

- ◆ records the time from the start of data arriving until all data has arrived
- ◆ calculates the throughput as the amount of data sent per unit of time

## ◆ The technique for measuring jitter (known as a packet train)

- ◆ A sender

- ◆ emits a series of packets with a small fixed delay between packets
  - > typically, packets in the train are sent back-to-back

- ◆ A receiver

- ◆ records the time at which each packet arrives
- ◆ uses the sequence of times to compute the differences in delay

# Measuring Delay

- ◆ **Measurement of the delay on a path from host A to host B**  
(requires that the two hosts have synchronized clocks)
  - ◆ The clocks must be extremely accurate
- ◆ **Instead of using synchronized clocks**
  - ◆ many network measurement tools choose an easier approach:
    - ◆ measure the **round-trip time** and divide by two
    - ◆ for example, **ping** can be used
- ◆ **Measuring network performance can be difficult**
  - ◆ **Because of the following reasons:**
    - ◆ Routes can be asymmetric
    - ◆ Conditions change rapidly
    - ◆ Measurement can affect performance
    - ◆ Traffic is bursty

***Details of these can be found in the text book***

# Passive Measurement, Small Packets, and NetFlow

- ◆ We should distinguish between two forms of measurement:
- ◆ Active
  - ◆ by injecting traffic into a network
  - ◆ the measurement traffic affect the performance of the network
- ◆ Passive
  - ◆ monitors a network and counts packets
  - ◆ but does not inject additional traffic
- ◆ A switch/router capacity is measured in packets per second
- ◆ Also we should distinguish between the **data rate** and the **packet rate**
- ◆ One of the most widely used passive measurement techniques was originally created by **Cisco Systems**
  - ◆ and it is now an IETF standard: **NetFlow**

# Quality of Service (QoS)

- ◆ **The counterpart of network measurement is network provisioning:**
  - ◆ designing a network to provide a specific level of service
- ◆ This topic is known as **Quality of Service (QoS)**
- ◆ **What is QoS?**
  - ◆ consider the contract between a service provider and a customer
- ◆ **The simplest contracts define a service**
  - ◆ by specifying the capacity that the provider guarantees
- ◆ **More complex contracts define **tiered services****
  - ◆ where the **level of service** received depends on the amount paid
- ◆ **Large corporate customers often demand more stringent service guarantees**
  - ◆ **The financial industry typically creates service contracts**
    - ◆ with bounds on the delay between specific locations

# Fine-Grain and Coarse-Grain QoS

- ◆ What technologies does a provider use to enforce QoS?
- ◆ Below are the two general approaches that have been proposed for service specification

Approach	Description
Fine-Grain	A provider allows a customer to state specific QoS requirements for a given instance of communication; a customer makes a request each time a flow is created (e.g., for each TCP connection)
Coarse-Grain	A provider specifies a few broad classes of service that are each suitable for one type of traffic; a customer must fit all traffic into the classes

- ◆ As the figure indicates, the approaches differ in their granularity and whether a provider or a customer selects parameters

# Fine-Grain QoS and Flows

- ◆ **The designers assumed a connection-oriented data network modelled after the telephone system:**
  - ◆ when a customer needed to communicate with a remote site, the customers would create a connection
- ◆ **Designers assumed a customer would issue QoS requirements for each connection**
  - ◆ and a provider would compute a charge according to the distance spanned and QoS used
- ◆ **The phone companies incorporated many QoS features in the design of **Asynchronous Transmission Mode (ATM)****
  - ◆ **ATM did not survive and providers do not generally charge for each connection**
    - ◆ But some of the terminology that ATM created for fine-grain QoS still persists with minor modifications

# Fine-Grain QoS and Flows

- ◆ Instead of specifying the QoS on a connection
  - ◆ the term **flow** is used
  - ◆ A flow generally refers to transport-layer communication
- ◆ four main categories of service
  - ◆ that were present in ATM, and explains how they relate to flows

Abbreviation	Expansion	Meaning
CBR	Constant Bit Rate	Data enters the flow at a fixed rate, such as data from a digitized voice call entering at exactly 64 Kbps
VBR	Variable Bit Rate	Data enters the flow at a variable rate within specified statistical bounds
ABR	Available Bit Rate	The flow agrees to use whatever data rate is available at a given time
UBR	Unspecified Bit Rate	No bit rate is specified for the flow; the application is satisfied with best-effort service

# Fine-Grain QoS and Flows

- ◆ VBR asks users to specify:
  - ◆ Sustained Bit Rate (SBR)
  - ◆ Peak Bit Rate (PBR)
  - ◆ Sustained Burst Size (SBS)
  - ◆ Peak Burst Size (PBS)
- ◆ ABR service implies sharing:
  - ◆ a customer is willing to pay for any amount of service that is available
- ◆ When Internet QoS was first considered
  - ◆ telephone companies argued that fine-grain services would be needed before the quality of voice telephone calls over a packet network would be acceptable
    - ◆ consequently, in addition to the work on ATM, the research community began to explore fine-grain QoS on the Internet
- ◆ The research was known as **Integrated Services (IntServ)**

# Coarse-Grain QoS and Classes of Service

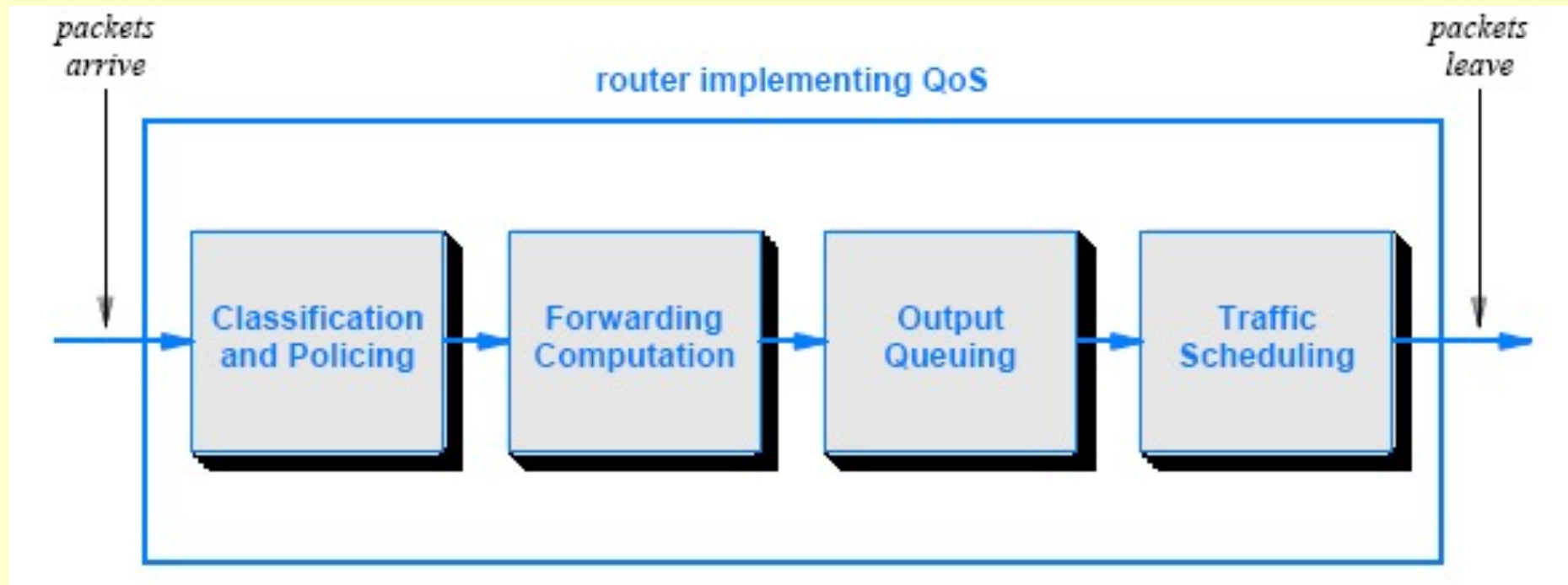
- ◆ In coarse-grain approach, traffic is divided into classes
  - ◆ and QoS parameters are assigned to the **class** rather than individual **flows**
- ◆ What is the motivation for the coarse-grain approach?
  - ◆ consider an implementation of QoS on a **core router**
- ◆ A core router carries traffic among major ISPs
  - ◆ It can handle millions of simultaneous flows
  - ◆ QoS requires many additional resources
- ◆ The router must maintain state for millions of flows
  - ◆ and must perform a complex computation for each packet
  - ◆ memory accesses slow down processing
  - ◆ it allocates resources when a flow begins
  - ◆ it deallocates them when a flow ends

# Coarse-Grain QoS and Classes of Service

- ◆ The research community and the IETF concluded that
  - ◆ a fine-grain approach was generally both impractical and unnecessary
- ◆ On one hand
  - ◆ an average user will not have sufficient understanding of QoS to choose parameters
    - ◆ What throughput requirements would one specify for a connection to a typical web site?
- ◆ On the other hand
  - ◆ core routers have insufficient processing power to implement per-flow QoS
- ◆ Thus, most work on QoS concentrates on defining a few broad classes of service
  - ◆ rather than trying to provide end-to-end QoS for each individual flow

# Implementation of QoS

- ◆ The steps a switch or router uses to implement QoS

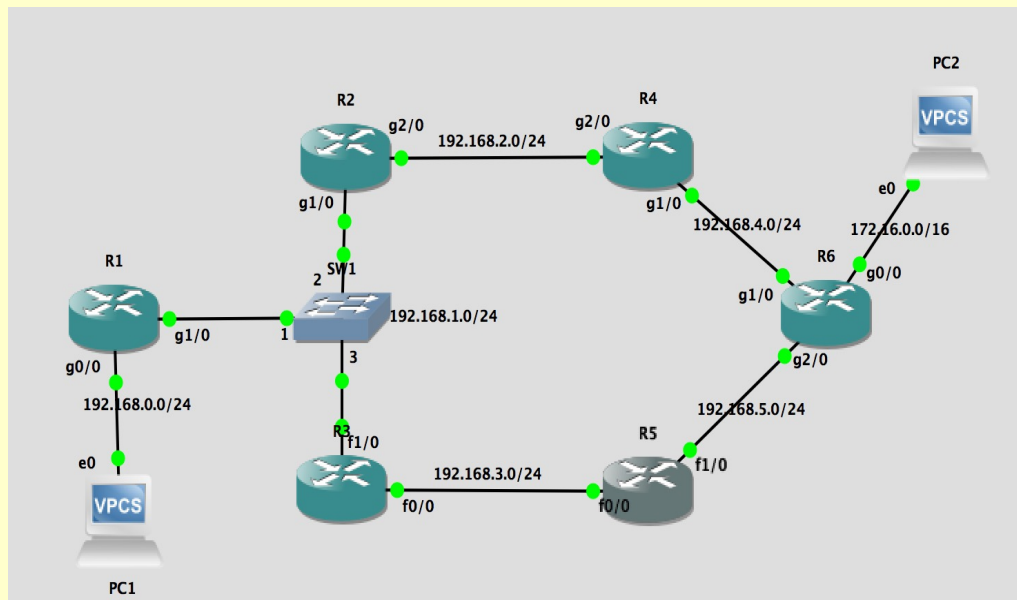


# QoS - Classification and Policing

- ◆ Router **classifies** the packet by assigning to it a flow identifier
- ◆ For a fine-grain system
  - ◆ the identifier specifies an individual connection (flow)
    - ◆ for a coarse-grain system, the identifier specifies a traffic class
- ◆ Once an identifier has been assigned
  - ◆ the router performs **policing**
    - ◆ which means that the router verifies that the packet does not violate parameters for the flow
  - ◆ If a customer sends faster than the max rate (customer is paying)
    - ◆ a policer begins to discard packets
- ◆ One technique used for policing is **Random Early Discard (RED)**
  - ◆ Packets on a given flow are dropped probabilistically
  - ◆ A queue is established for the flow
  - ◆ And the current size of the queue is used to determine the probability of drop

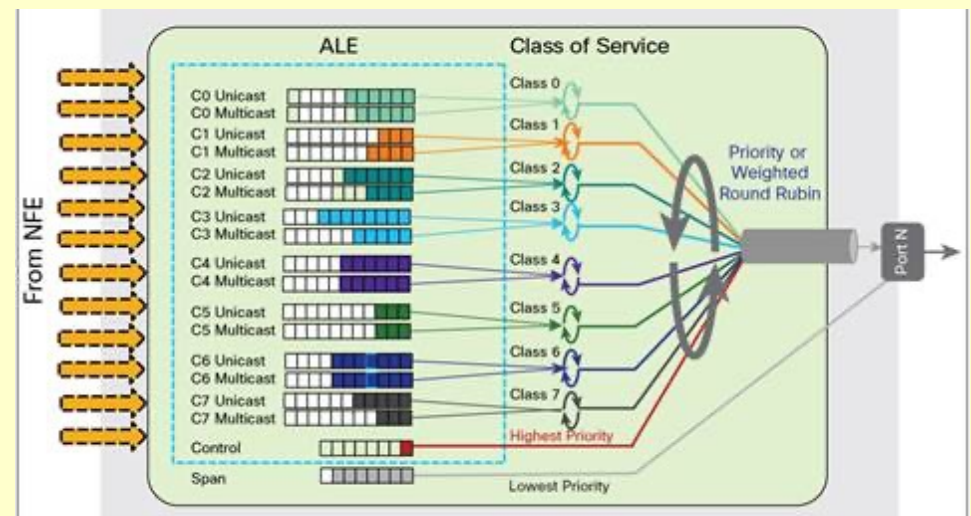
# QoS - Forwarding

Can select among multiple paths  
(router may have many output queues)



# QoS - Output Queuing

- ◆ Most implementations of QoS create a set of queues
  - ◆ for each output port
- ◆ Once the forwarding computation selects an output port for the packet
  - ◆ the output queuing mechanism uses the flow identifier to place the packet in one of the queues associated with the port
- ◆ A coarse-grain system typically uses one queue per class
  - ◆ Thus, if a manager establishes eight QoS classes
    - ◆ each output port will have eight queues
- ◆ A fine-grain system often has one queue per connection
  - ◆ with queues arranged in a hierarchy



# QoS - Traffic Scheduling

- ◆ **A traffic scheduler implements the QoS policies**
  - ◆ **by selecting a packet to send whenever a port is idle**
    - ◆ For example, a manager might specify that three customers each receive 25% of the capacity and all other customers share the remaining capacity
- ◆ **To implement such a policy**
  - ◆ a traffic scheduler might use four queues and a **round-robin** approach to select packets
- ◆ **More sophisticated packet selection algorithms can be used**
  - ◆ to implement complex proportional sharing
- ◆ **Complexity arises because a traffic scheduler must maintain long-term policies**
  - ◆ even though packets arrive in bursts

# QoS - Traffic Scheduling

- ◆ Many traffic scheduling algorithms have been proposed
  - ◆ It is not possible to create a practical algorithm that achieves perfection; each is a compromise between fairness and computational overhead.

Algorithm	Description
Leaky Bucket	Allows a queue to send packets at a fixed rate by incrementing a packet counter periodically and using the counter to control transmission
Token Bucket	Allows a queue to send data at a fixed rate by incrementing a byte counter periodically and using the counter to control transmission
Weighted Round Robin	Selects packets from a set of queues according to a set of weights that divide the capacity into fixed percentages, assuming a uniform packet size
Deficit Round Robin	A variant of the round-robin approach that accounts for bytes sent rather than packets transferred, and allows a temporary deficit caused by a large packet

# Internet QoS Technologies

- ◆ **The IETF has designed a series of technologies and protocols related to QoS**
- ◆ **Three significant efforts are:**
  - ◆ **RSVP and COPS (IntServ)**
  - ◆ **DiffServ**
  - ◆ **MPLS**

# Resource reSeRVation Protocol (RSVP)

An application sends a request that specifies the desired QoS

Each router along the path from the source to the destination

- reserves the requested resources and
- passes the request to the next router

Eventually, the destination host must agree to the request

When every hop along the path has agreed to **honour the request**

a flow identifier is generated and returned

Traffic can then be sent along the reserved path

## Common Open Policy Services (COPS)

- ◆ A companion protocol for RSVP used to specify and enforce policies
- ◆ A router that implements policing
  - ◆ uses COPS to communicate with a policy server and
  - ◆ obtain information about the flow parameters

# DiffServ

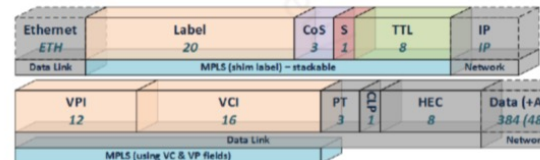
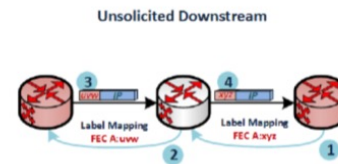
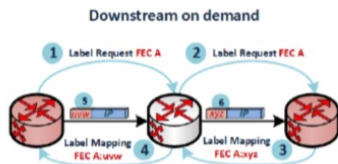
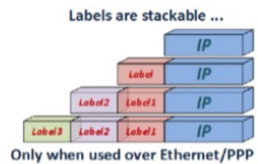
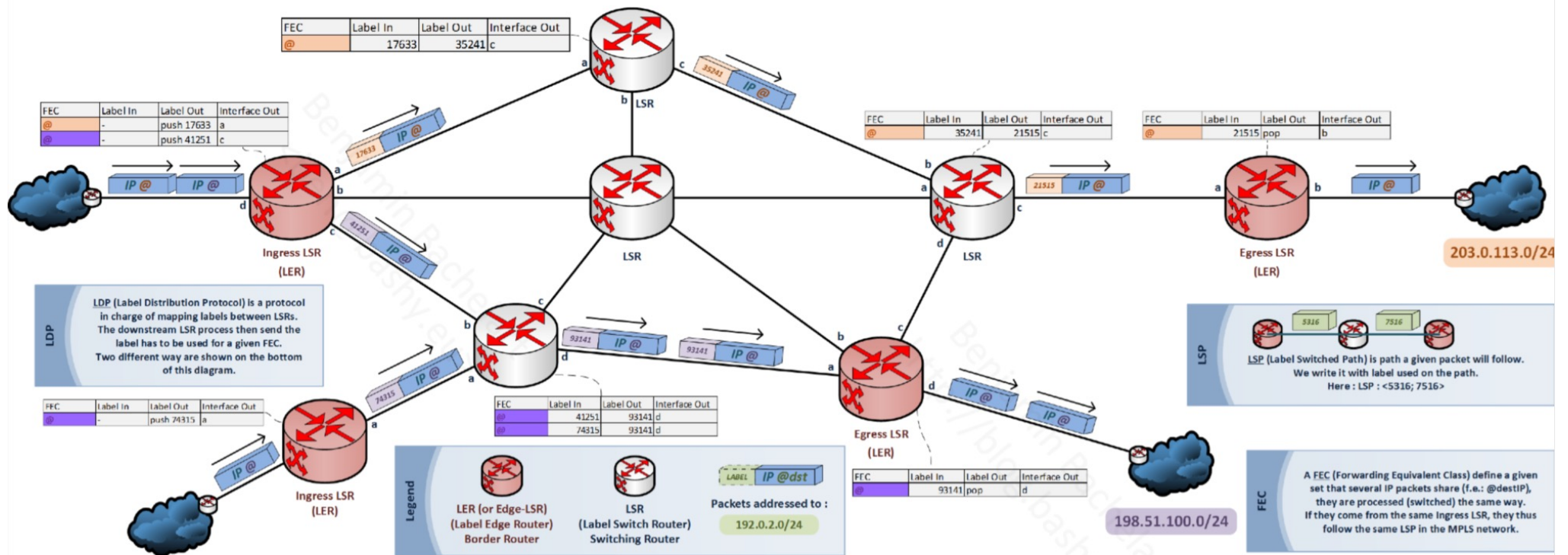
- ◆ **IETF**
  - ◆ **abandoned IntServ and fine-grain QoS**
  - ◆ **then created Differentiated Services (DiffServ)**
    - ◆ to define a coarse-grain QoS mechanism
- ◆ **The DiffServ effort produced some answers to the following questions:**
  - ◆ How can classes be specified?
  - ◆ How can the TYPE OF SERVICE field in an IPv4 or IPv6 header be used to specify the class of a datagram?
- ◆ **Although various ISPs have experimented with DiffServ**
  - ◆ the technology does not enjoy widespread acceptance

# MultiProtocol Label Switching (MPLS)

- ◆ **A connection-oriented mechanism built below IP (IP is a layer 3 protocol, MPLS is considered to be 2.5 layer protocol)**
- ◆ **Forwarding paths through a set of MPLS-capable routers are configured**
  - ◆ **At one end of a path**
    - ◆ each datagram is encapsulated in an MPLS header
    - ◆ and injected into the MPLS path
  - ◆ **At the other end**
    - ◆ each datagram is extracted, the MPLS header is removed
    - ◆ and the datagram is forwarded to its destination
  - ◆ **In many cases, a traffic scheduling policy is assigned to an MPLS path**
    - ◆ means that when a datagram is inserted in a particular path, QoS parameters are set for the datagram
- ◆ **Thus, an ISP might set up an MPLS path for voice data**
  - ◆ that is separate from the MPLS path used for other data

# MPLS example

## MPLS



# Summary

- ◆ Quantitative measures of networks include delay, throughput, goodput, and jitter
- ◆ Delay increases as utilization increases
- ◆ One can purchase more throughput, but not easily less delay
- ◆ Quality of Service (QoS) technologies provide guarantees on performance
- ◆ The industry has moved away from fine-grain QoS (per-flow as in ATM and IntServ) to coarse-grain QoS (DiffServ)
- ◆ Multi-Protocol Label Switching (MPLS) is used by tier-1 ISPs to provide circuit-oriented networking