CPSC 471: Computer Communications

Resource Allocation and Congestion Control

Figures from Computer Networks: A Systems Approach, version 6.02dev (Larry L. Peterson and Bruce S. Davie)

Introduction

- Packets contend at a router for use of a link
  - Each packet placed in a queue
- Congestion control and resource allocation are related
  - Congestion can be avoided if network allocates resources effectively
  - Easier to recover from congestion once it happens

Resource Allocation Definition

- How network elements try to meet demands for network resources
  - E.g., link bandwidth and router/switch buffer space
- Not isolated to a single level of a protocol hierarchy

Congestion Control Definition

- Efforts made by network nodes to prevent or respond to overload conditions
  - Usually involve “fairness”
- Flow control
  - Keep a fast sender from overrunning a slow receiver
- Congestion control
  - Keep senders from sending too much data into the network because of a lack of resources at some point

Resource Allocation Models

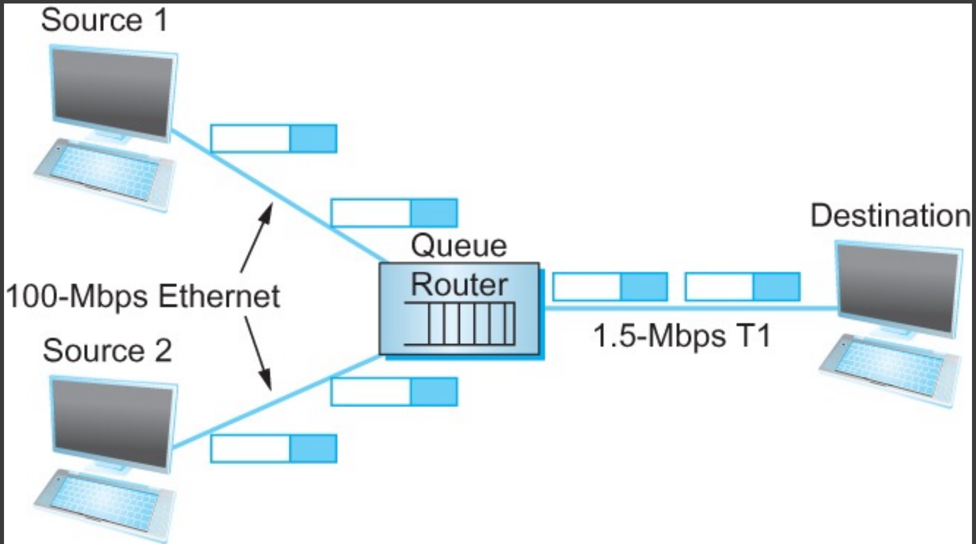


Figure 152

- Packet-switched network/internetwork
- Connection-oriented networks allocate network resources during connection setup

Flows

- Sequence of packets sent between a source/destination pair
  - Follows the same route through the network
  - Datagrams switched independently not completely independent

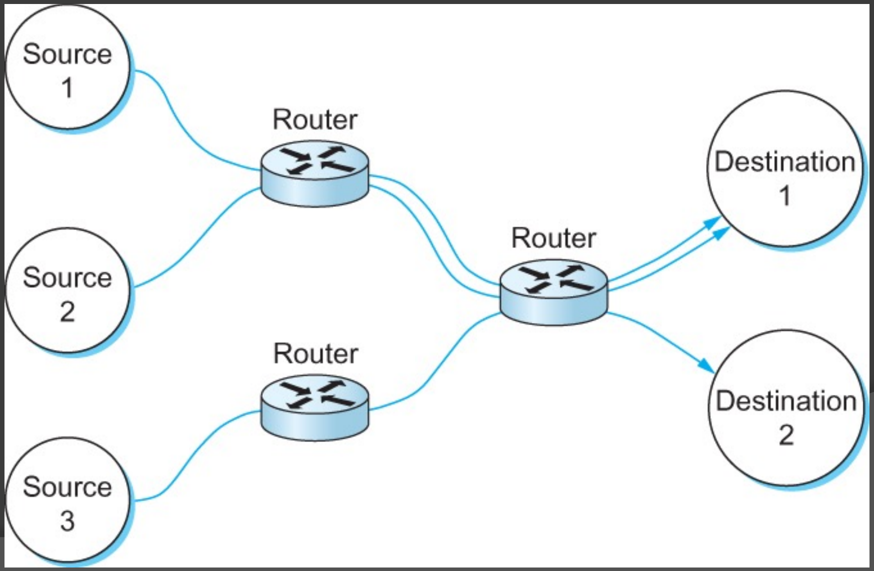


Figure 153

Flows continued

- May wish to maintain state info for each flow
- Flow may be
  - Implicitly defined
  - Explicitly established
- Service Model
  - Best-effort
  - Qualities of Service (QoS)

Resource Allocation Mechanisms

- Router-centric approach
  - Address resource allocation from inside the network
- Host-centric approach
  - Address resource allocation from edges of the network
- Not mutually exclusive

Resource Allocation Mechanisms continued

- Reservation-based approach
  - End host asks the network for a certain amount of capacity to be allocated for a flow
- Feedback-based approach
  - End hosts send data then adjust sending rate according to feedback they receive
  - Feedback can be
    - Explicit
    - Implicit

Resource Allocation Mechanisms continued

- Window-based approach
  - Window advertisement can be used to reserve buffer space
- Rate-based approach
  - How many bits/second receiver or network can absorb

Summary of Resource Allocation Mechanisms

- Two general strategies
  - Best-effort service model
    - Feedback
    - Host-centric
    - Window-based
    - General strategy adopted by the Internet
  - QoS-based service model
    - Reservation
    - Likely router-centric
    - Rate-based

Evaluation Criteria

- Network effectively and fairly allocates its resources
- Effective resource allocation
  - Power = Throughput / Delay

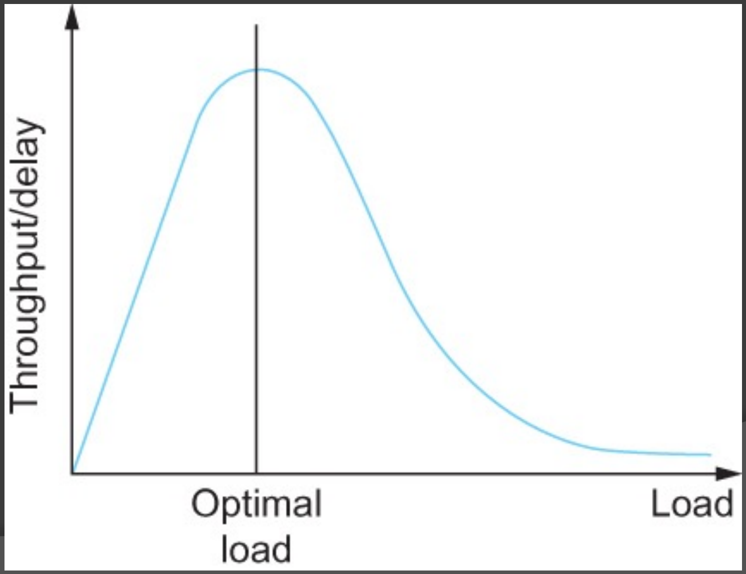


Figure 154

Evaluation Criteria continued

- Fair resource allocation
  - Equal does not necessarily mean fair

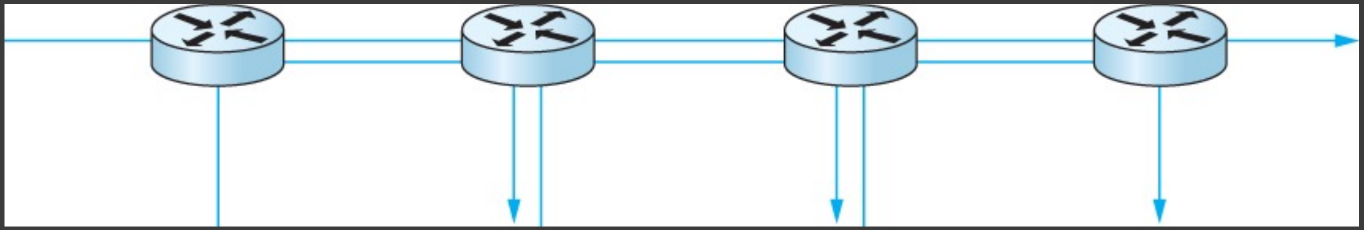


Figure 155

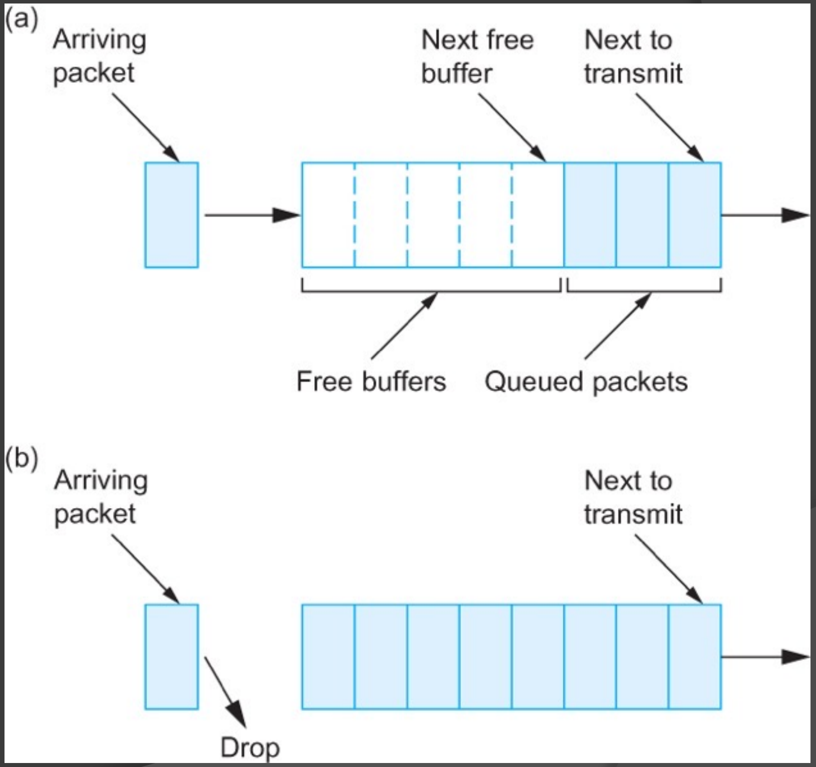
Queuing Disciplines

- Governs how packets are buffered while waiting to be transmitted
- Scheduling Discipline
- Drop Policy

FIFO

- First-come, first-served
- Droptail queue
- FIFO with tail drop
  - Simplest of all queuing algorithms
- Alternative:
  - Priority Queue

Figure 156



Fair Queuing

- FIFO does not separate packets by flow
- FQ maintains a separate queue for each flow handled by the router

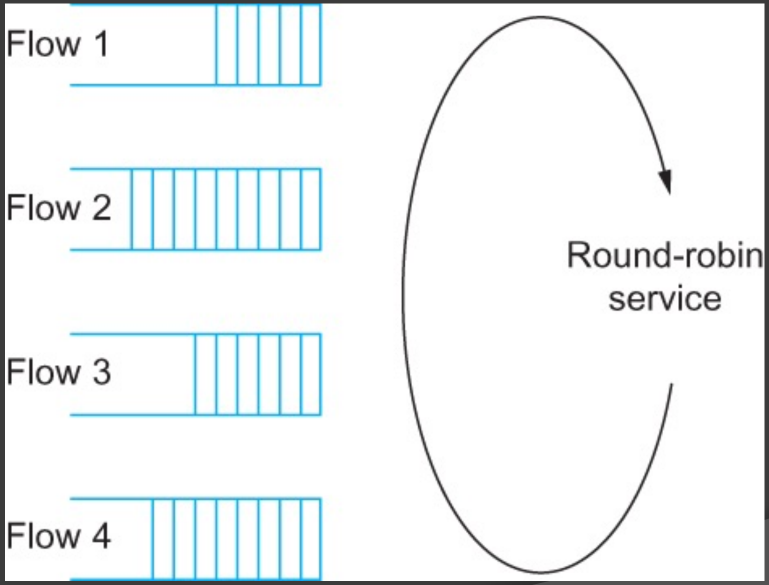


Figure 157

Fair Queuing continued

- Packets serviced at the router may not be the same length
- Uses finishing time to sequence the packets for transmission

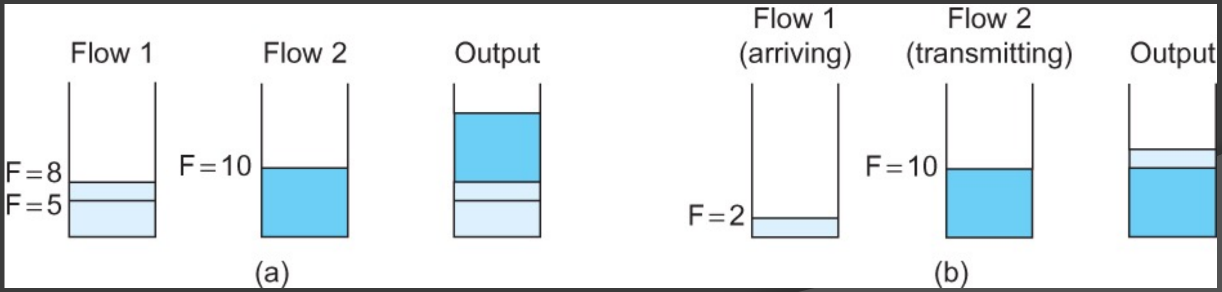


Figure 158

Fair Queuing continued

- Work conserving
  - Link is never left idle as long as there is at least one packet in the queue
- Provides a guaranteed minimum share of bandwidth to each flow
- Weighted Fair Queuing (WFQ)
  - Weight specifies how many bits to transmit each time the router services queue

TCP Congestion Control

- Each TCP source
  - Determines how many packets can safely be in transit
  - Uses the arrival of an ACK to signal a packet has left the network
- Additive Increase/Multiplicative Decrease
- Slow Start
- Fast Retransmit and Fast Recovery

Additive Increase/ Multiplicative Decrease (AIMD)

- TCP maintains a *CongestionWindow*
  - Limits how much data the source can have in transit
- Source limited by *CongestionWindow* or *AdvertisedWindow*
- How to set *CongestionWindow*?

Multiplicative Decrease

- TCP interprets timeouts as congestion
- Halves *CongestionWindow* when timeout occurs
  - *CongestionWindow* ≥ MSS

Additive Increase

- Must take advantage of new capacity in the network
  - Must increase CongestionWindow

- Increment CongestionWindow when ACKs received from last packets
- Increment = MSS x (MSS/CongestionWindow)
- CongestionWindow += Increment

Additive Increase

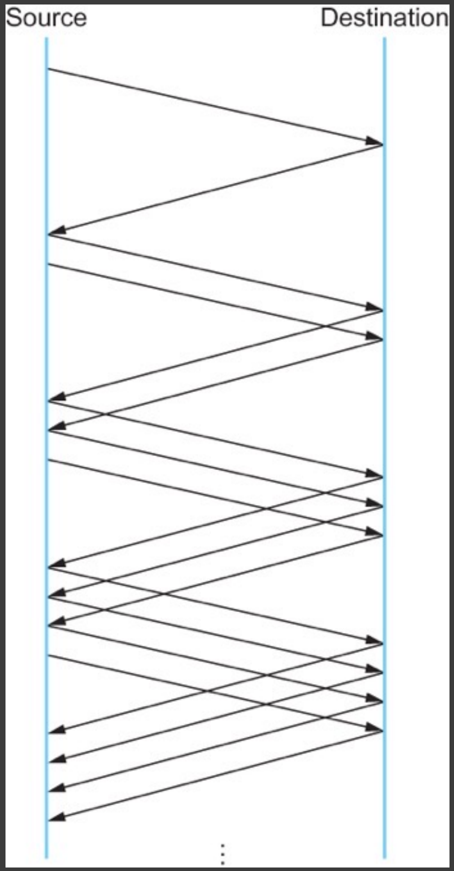


Figure 159

AIMD

- Source reduces congestion window at a faster rate than increasing it
- AIMD is necessary condition for stable congestion-control

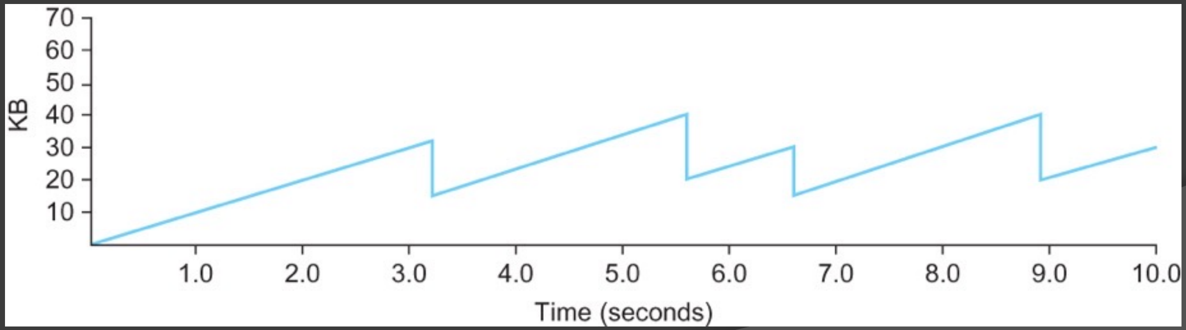


Figure 160

Slow Start

- Rapidly increases congestion window from a cold start
  - Doubles the number of packets in transit every RTT
- Occurs at
  - Beginning of a connection
  - When the connection goes dead waiting for a timeout to occur

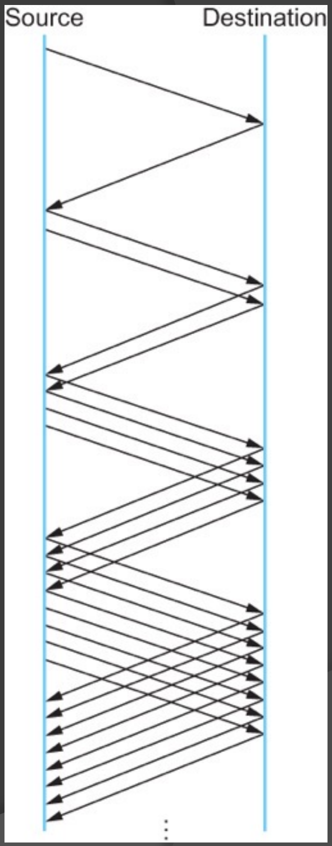


Figure 161

Slow Start

- At beginning of connection
  - Doubles congestion window each RTT until loss
    - Timeout causes multiplicative decrease
- When connection goes dead waiting for timeout
  - Restarts the flow of data
  - Already has a value for congestion window
    - Use slow start to reach CongestionThreshold

Slow Start Behavior

- Blue line: Congestion Window
- Hash marks: A packet is transmitted
- Vertical bars: Time when retransmitted packet was 1st transmitted
- Bullets: Timeouts

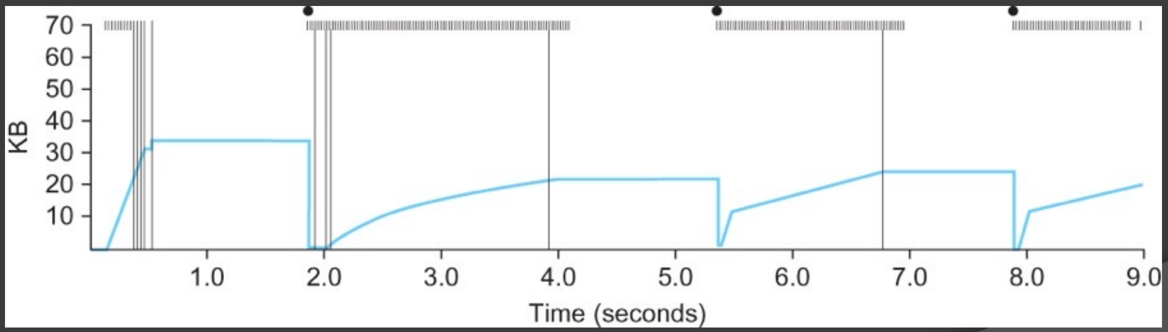


Figure 162

Slow Start

- Many packets lost during initial slow start period
- TCP attempts to learn how much network bandwidth is available
- Linear increase vs. Exponential increase

Fast Retransmit

- An enhancement to original proposal for TCP congestion control
- Triggers quicker retransmission of a dropped packet
  - Sooner than waiting for a timeout

Duplicate ACKs

- If packet arrives out-of-order
  - Receiver sends same ACK as last time
  - What does this tell the sender?
- TCP waits for three duplicate ACKs before retransmitting the packet

Fast Retransmit Behavior

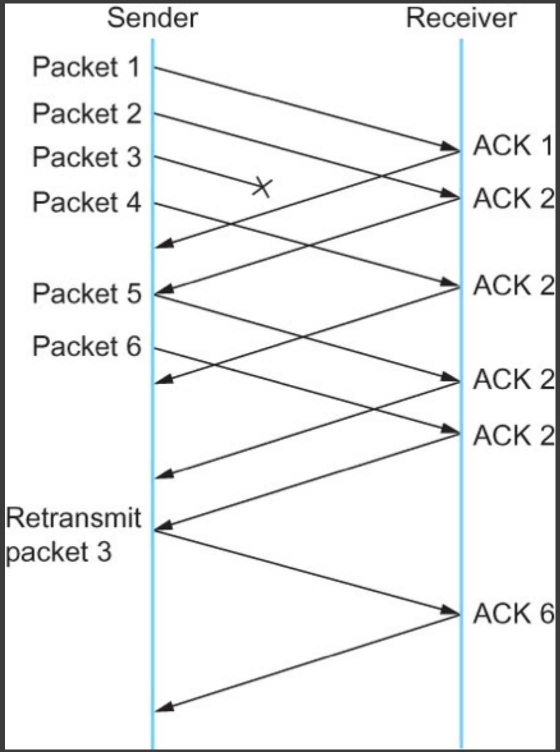


Figure 163

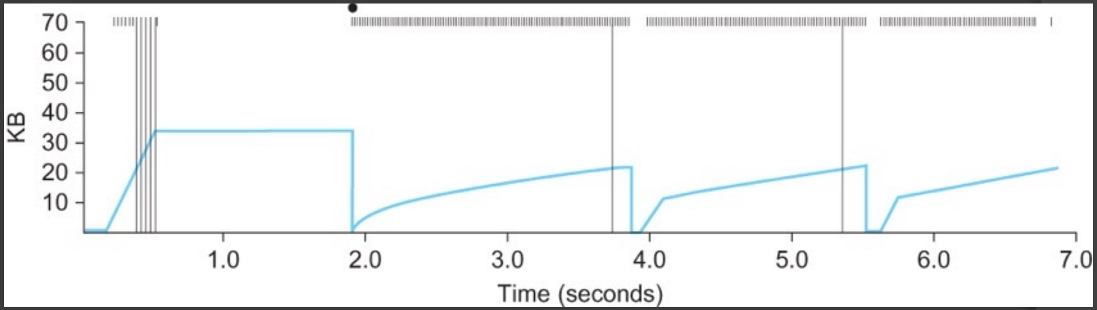


Figure 164

Fast Recovery

- When fast retransmit signals congestion
  - Use ACKs that are still in transit to clock sending packets
  - Don’t set congestion window to one packet and run slow start

Congestion-Avoidance Mechanisms

- TCP increases the load until losses occur and are detected
  - Controls congestion instead of avoiding it

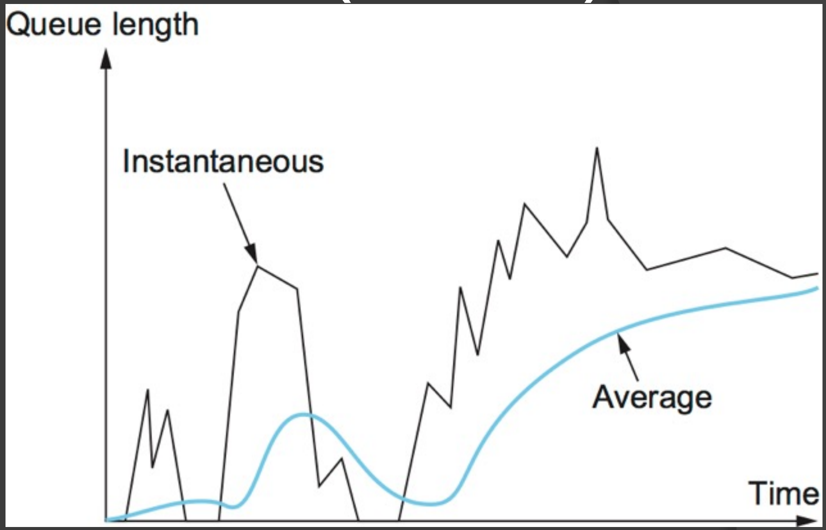
DECbit

- Each router monitors load and sets a congestion bit in packet
- Host looks at how many of last window's packets had bit set
  - < 50%
    - Increase congestion window by one packet
  - ≥ 50%
    - Decrease congestion window by 12.5%

Random Early Detection (RED)

- Router drops a source’s packet early
  - Implicit notification
- Drops a packet based on drop probability
  - Based on average queue length

Figure 167



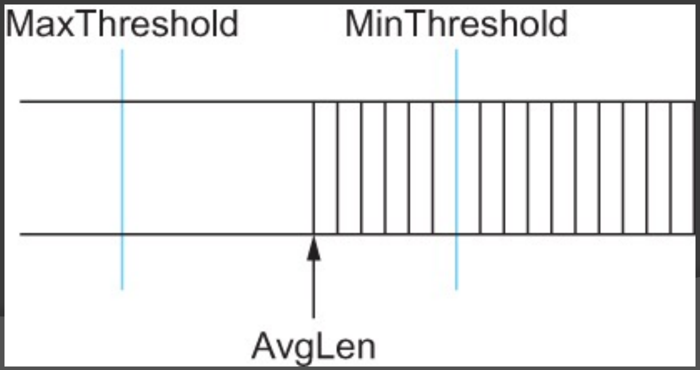


Figure 168

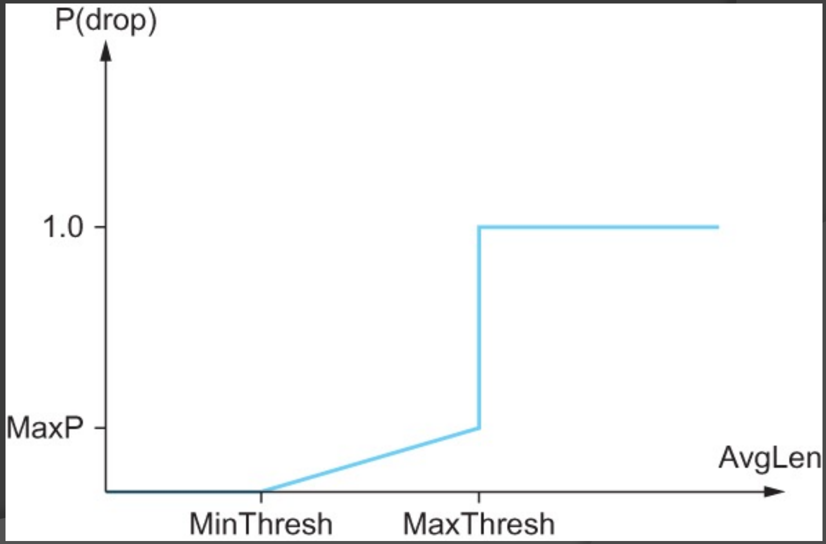


Figure 169

Source-Based Congestion Avoidance

- End hosts detect congestion instead of routers
- Check if measurable increase in RTT
- Could look at RTT and window size
  - (CurrentWindow – OldWindow) x (CurrentRTT – OldRTT)
- Look for flattening of sending rate
  - Compares throughput
    - Throughput = (# bytes in transit)/RTT

TCP-Vegas

- Compares measured throughput rate to an expected throughput rate

- TCP-Tahoe
  - Original implementation of TCP congestion-control mechanism
- TCP-Reno
  - Adds fast recovery, header prediction, and delayed ACKs

Monitoring Throughput

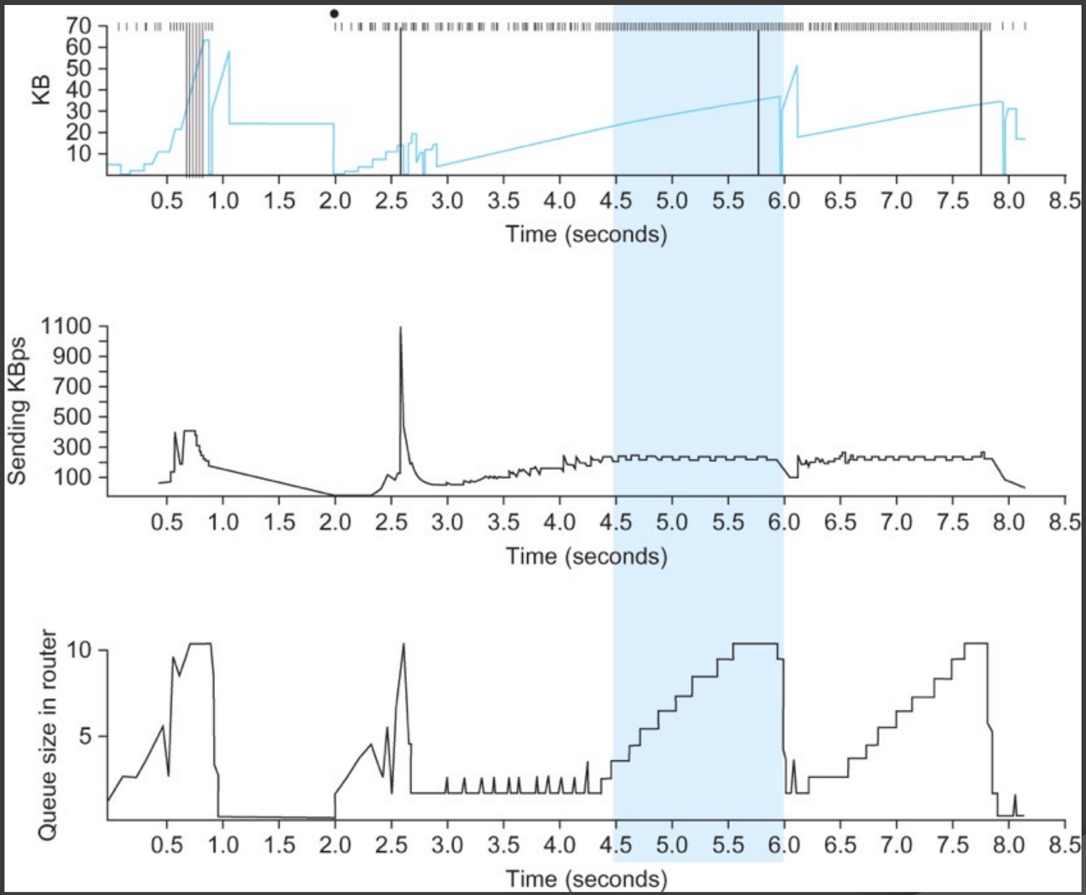


Figure 170

TCP-Vegas

- Tries to match available bandwidth of network
- Mechanism based on
  - Estimated amount of extra data in the network
  - Dropped packets
- Defines an expected rate
  - ExpectedRate = CongestionWindow/BaseRTT

TCP-Vegas

- Calculates ActualRate once per RTT
- Compares ActualRate to ExpectedRate and adjusts congestion window
- Defines two thresholds
  - Goal is to keep sending rate between alpha and beta

- Linear early decrease before congestion occurs
- Multiplicative decrease used for timeouts

TCP-Vegas Behavior

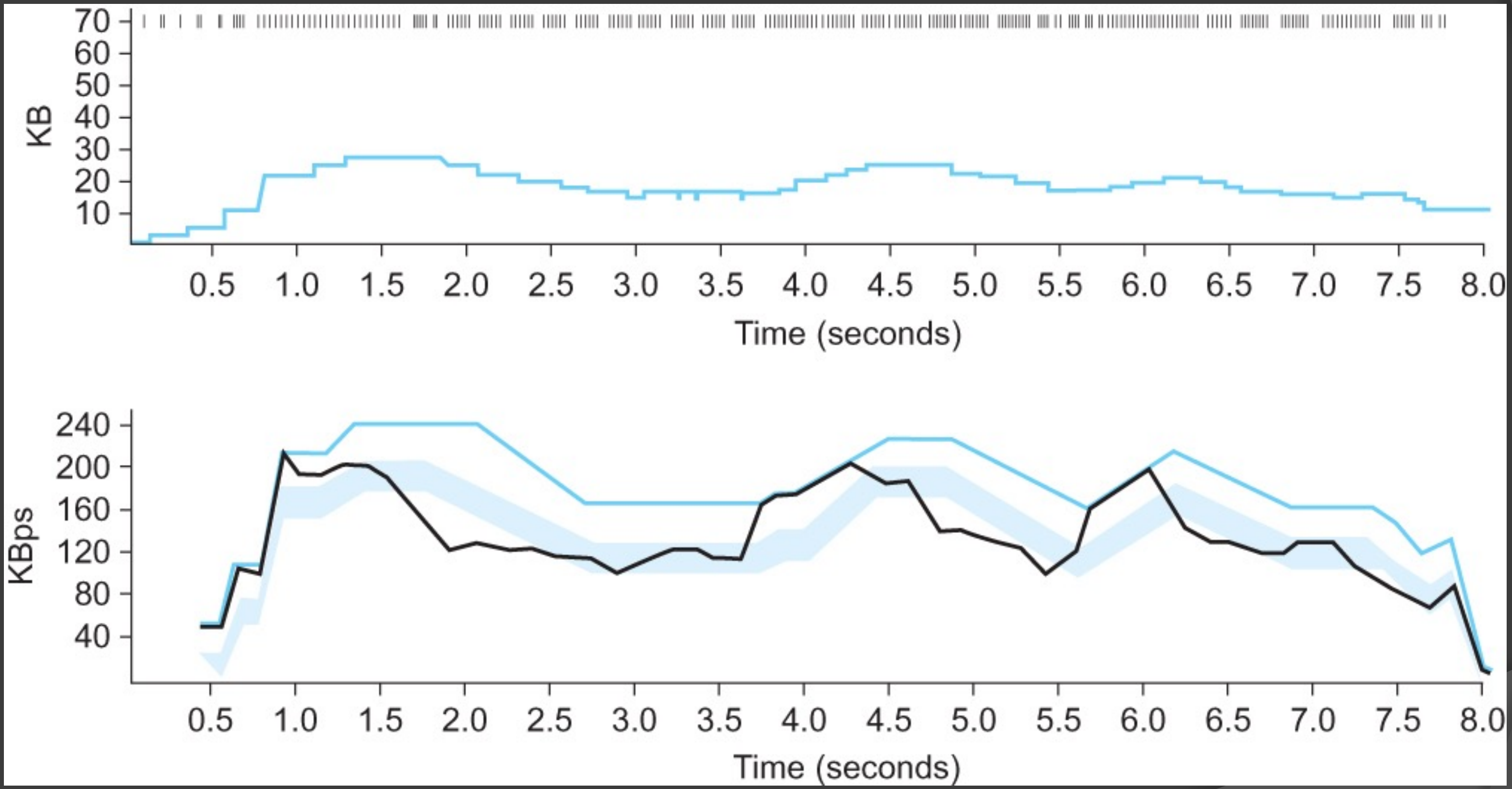


Figure 171

Real-Time Applications

- Multimedia applications
  - Have higher bandwidth needs
  - Timeliness of delivery can be very important
- Real-time applications
  - Need assurance from network that data is likely to arrive on time
- Non-real-time applications
  - Data arrives correctly, but cannot provide timeliness
  - Also called "elastic"

Quality of Service (QoS)

- Implied that network will treat some packets differently than others
  - Not done in the best-effort model
- This is QoS
  - Network provides different levels of service

Voice Example

- Each sample/packet has a playback time
  - Point of time it is needed by the receiver
- If data arrives after the playback time, it is useless
- Limits to how far we can delay playback of data

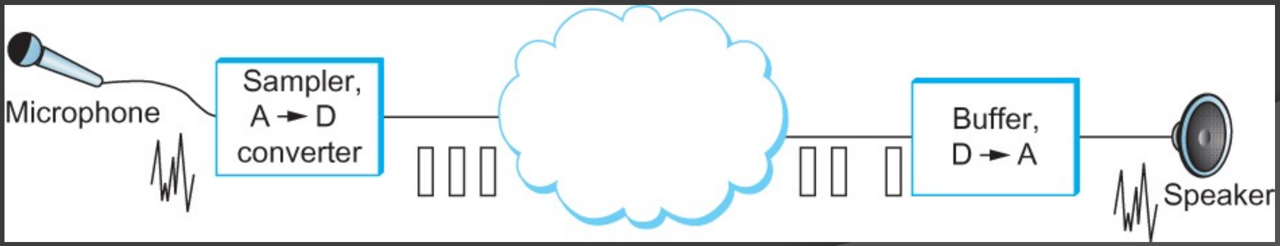


Figure 172

Playback Buffer

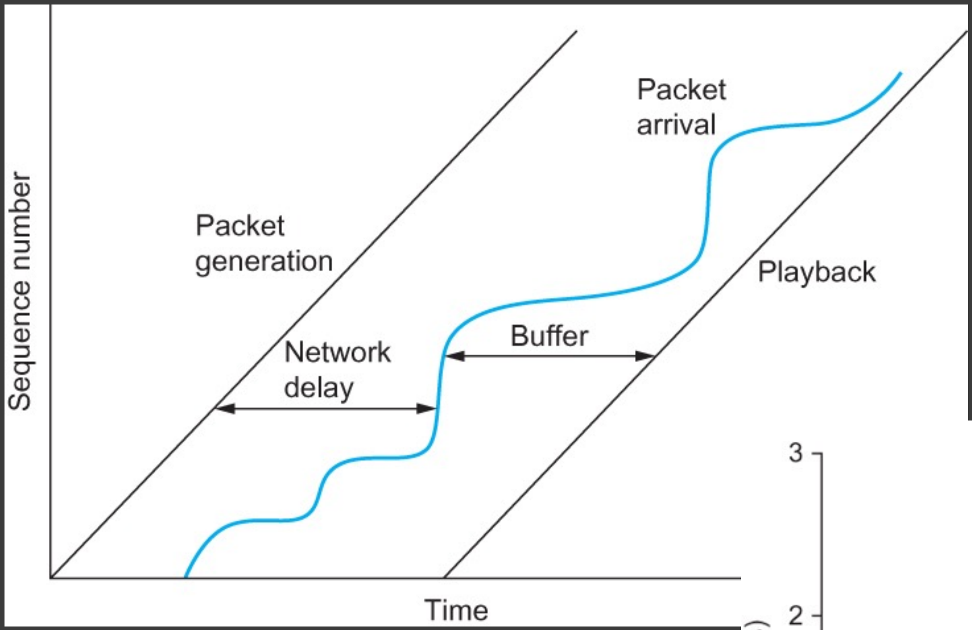


Figure 173

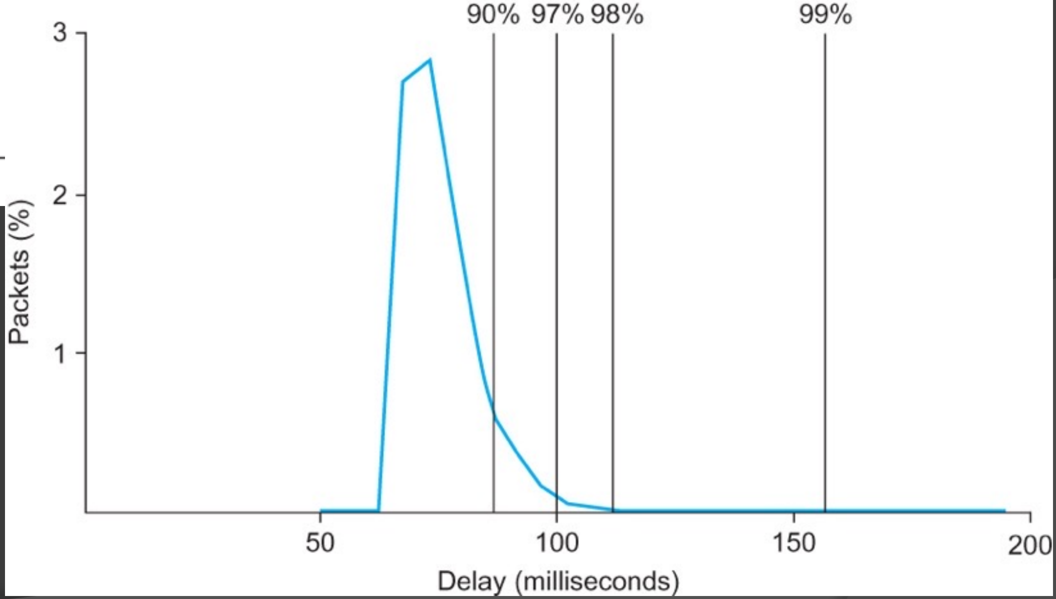


Figure 174

Taxonomy of Real-Time Apps

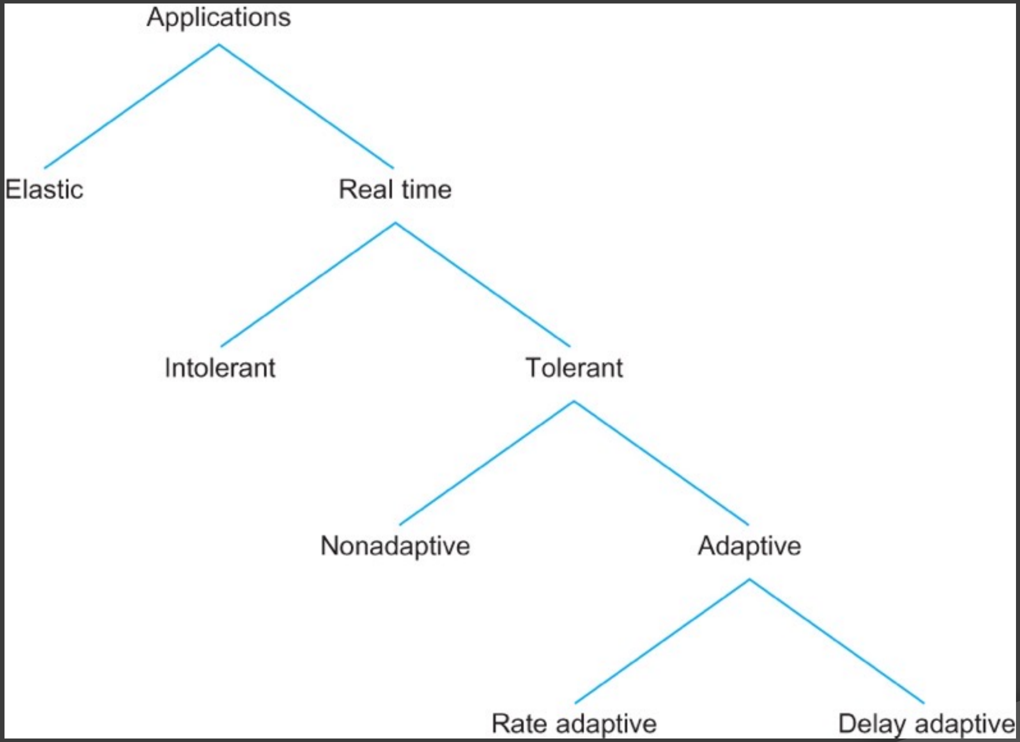


Figure 175

Taxonomy of Real-Time Apps

- Tolerance of loss
  - Tolerant or intolerant
- Adaptability
  - Delay-adaptive applications
    - Can adjust their playback point
  - Rate-adaptive applications
    - Trade off bit rate vs. quality

Approaches to QoS Support

- Fine-grained approaches
  - Provide QoS to individual applications or flows
  - Find “Integrated Services”
- Course-grained approaches
  - Provide QoS to large classes of data or traffic
  - Find “Differentiated Services”
    - Most widely-deployed QoS mechanism

Integrated Services

- Designed service classes to meet application needs
  - Guaranteed Service
    - Guarantee the maximum delay a packet will see
  - Controlled Load Service
    - Emulate a lightly-loaded network
- RSVP (Resource Reservation Protocol) can be used to make reservations using these service classes

Integrated Services Mechanisms

- Flowspec
  - Tells the network the type of service required
- Admission Control
  - Network decides if it can provide the service
  - When to say yes/no
- Policing
- Resource Reservation
  - How requests for service, flowspecs, and admission control decisions are exchanged
- Packet Scheduling
  - Managing how packets are queued and scheduled to meet flow requirements

Flowspec

- RSpec: Describes the service requested
  - Specify a target delay or bound
- TSpec: Describes flow’s traffic characteristics
  - Give network info about bandwidth used by flow
  - Bandwidth varies constantly for most apps
  - Knowing the average bandwidth is not enough

Token Bucket Filter

- Described by two parameters
  - A token rate (*r*)
  - A bucket depth (*B*)
- A token is needed to send a byte
- Tokens accumulated at a rate *r* per sec
- No more than *B* tokens can be accumulated

Token Bucket Filter

- Can send a burst of *B* bytes
  - But cannot send more than *r* bytes per sec over an interval
- Info is important for admission control algorithm

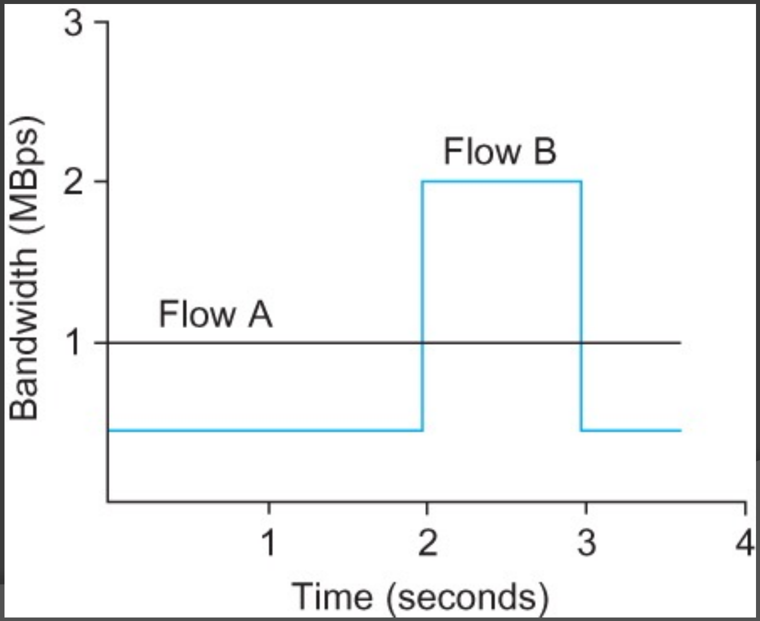


Figure 176

Integrated Services Reservation Protocol

- Some sort of setup protocol is necessary to establish state
  - Connectionless networks do not have this
- Receiver needs to know sender's TSpec
- Needs to know what path the flow will follow
  - Resource reservations must occur for each router on the path

Integrated Services Reservation Protocol

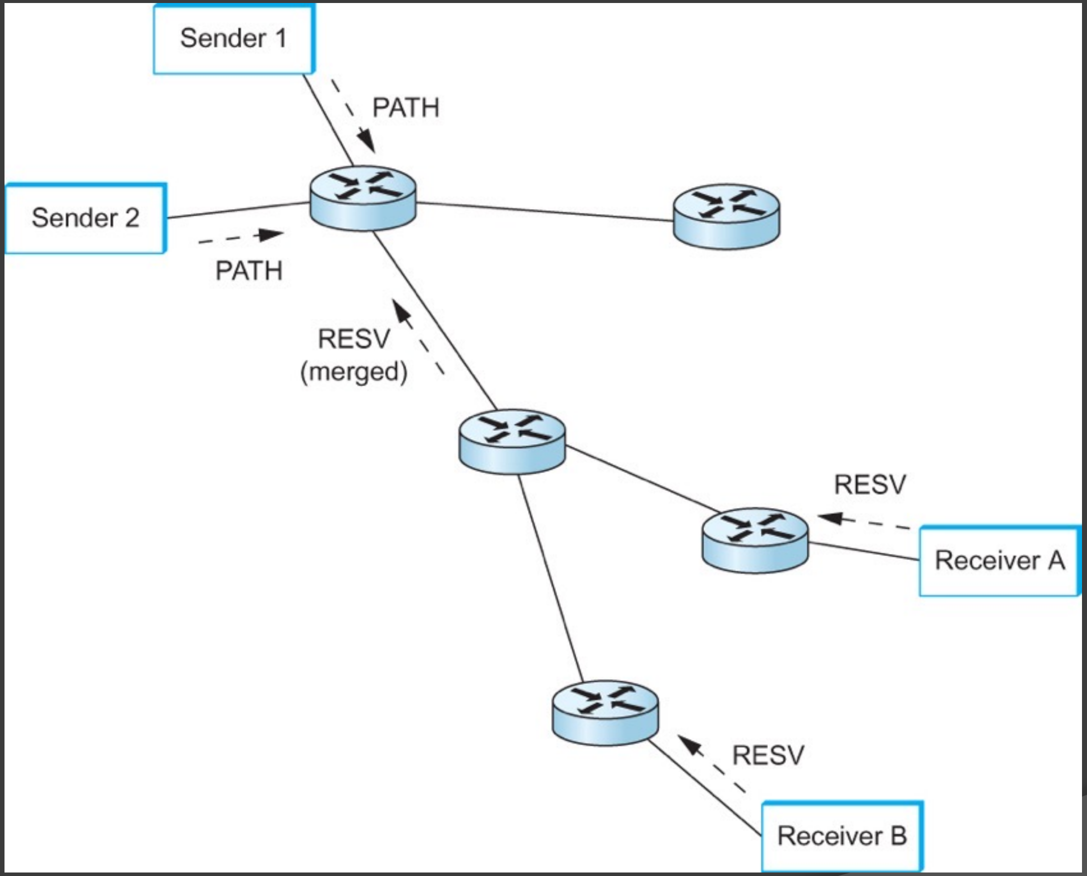


Figure 177

Integrated Services

- Packet Classifying
  - Associate each packet with the appropriate reservation
- Packet Scheduling
  - Manage the packets in the queues so they receive the requested service
- Suffers from scalability issues
  - Need a lot of state at routers for each flow

Differentiated Services

- Allocates resources to a small number of classes of traffic
- Example:
  - Normal service and premium service
  - Use a bit in the packet header
  - Who sets the premium bit and when?
  - What does a router do differently for premium service?

Per-Hop Behaviors (PHBs)

- Indicates the behavior of individual routers
- Expedited Forwarding (EF) PHB
  - Forward marked packets with minimal delay and loss
- Assured Forwarding (AF) PHB
  - Came from “RED with In and Out” (RIO) or “Weighted RED”
    - Enhancements to basic RED algorithm

RIO

- Packets are "in" or "out" of profile for assured service
  - E.g., profile may be a rate, packets "out" of profile exceed the rate

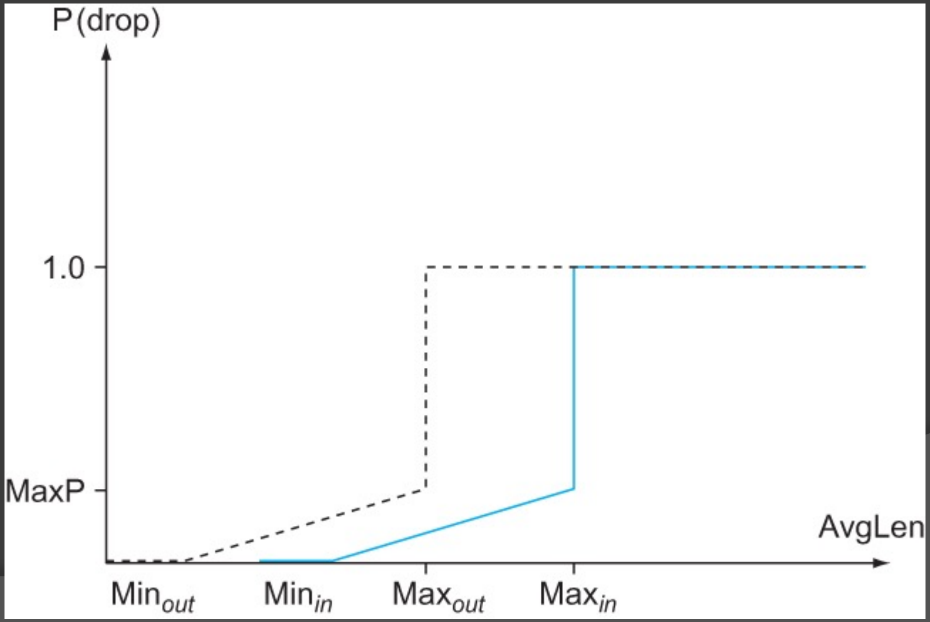


Figure 178