THE UNIVERSITY OF SYDNEY

Lecture 10

Quality of Service in IP Networks

ELEC 3506/9506
Communication Networks

Dr Wibowo Hardjawana
School of Electrical and Information Engineering

Topics of the day

- Introduction to Quality of Service
- Flow Characteristics
- QoS Metrics
- QoS Principles
- Traffic Management
- Resource Management

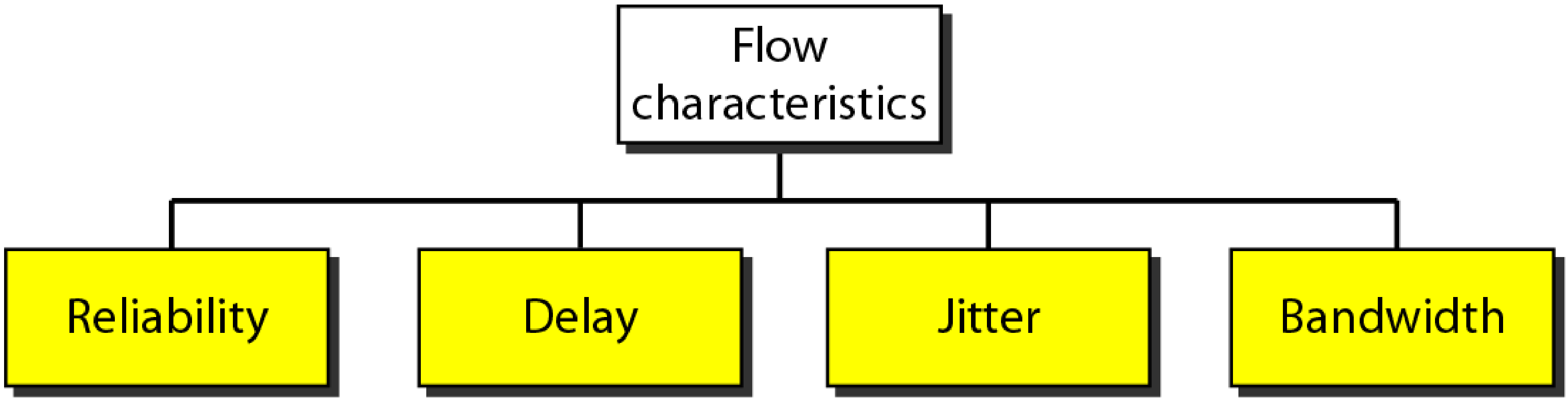
QoS (Quality of Service) in IP Networks

- What is Network QoS?

  - Description or measurement of the overall performance of the network
  - The Internet was originally designed for best-effort service (elastic traffic)
  - A set of techniques and mechanisms that guarantee the performance of the network to deliver predictable service
  - We may loosely define quality of service as something a flow seeks to attain.

Flow Characteristics

- Traditionally, the following characteristics are attributed to a flow.



THE UNIVERSITY OF SYDNEY

Reliability

- Lack of reliability means losing a packet or acknowledgment, which entails retransmission.
- However the sensitivity of application programs to reliability is not the same.

  - Reliability is more important for email, file transfer, web browsing.
  - Reliability is less important for telephony and audio/video conferencing.

Delay

- Source to destination delay is another characteristic.

- This consists of:
  - Sum of delays at routers
  - Propagation delay in the media
  - Set up mechanisms

- Delay tolerance is also application specific
  - Min delay tolerant - telephony, audio/video conferencing, remote log-in
  - High delay tolerant - file transfer and e-mail

Network Delay

- Consists of: Transmission delay, Propagation delay, Processing delay, and Queuing delay
  - **Transmission Delay = (Packet length) / (Transmission rate)**
  - **Propagation Delay = (Distance) / (Propagation speed)**
  - **Processing Delay = Time required to process a packet in a router or a destination host**
  - **Queuing Delay = The time a packet waits in input and output queues in a router**

Jitter

- Variation of delay for packets in the same flow.

- Low jitter
  - Difference is small
- High jitter
  - Difference between delays is large

- Variation of delay is not acceptable for audio and video applications.
  - Except with sufficient application buffering

Bandwidth / Capacity

- Maximum rate at which data can be transferred over a given link.
- Measured in bits per second (bps).
- Bandwidth / Capacity requirements.
  - Application specific.
  - Flow specific.

QoS Metrics

- Throughput and Goodput
- Packet Loss
- Delay
- Jitter

Throughput

- The terms bandwidth and throughput are not same.
- Bandwidth is the **maximum** rate at which data can be moved over a given link.
- Throughput, sometimes called effective bandwidth, is the **actual** speed over a link or circuit.
- Throughput is less than the potential bandwidth.
- For example, we may have a link with bandwidth of 1 Mbps, but the throughput may be 200 kbps if the devices connected to the link may handle only 200kbps.

Goodput

- Goodput is the application level throughput
  - Number of useful/correct bits per unit of time
  - From source address to the certain destination
  - Excluding protocol overhead (E.g. Headers)
  - Excluding retransmitted data packets.

- Goodput < Throughput due to:
  - Protocol overhead
  - Transport layer flow control and congestion avoidance
  - Retransmission of lost or corrupt packets

Packet Loss

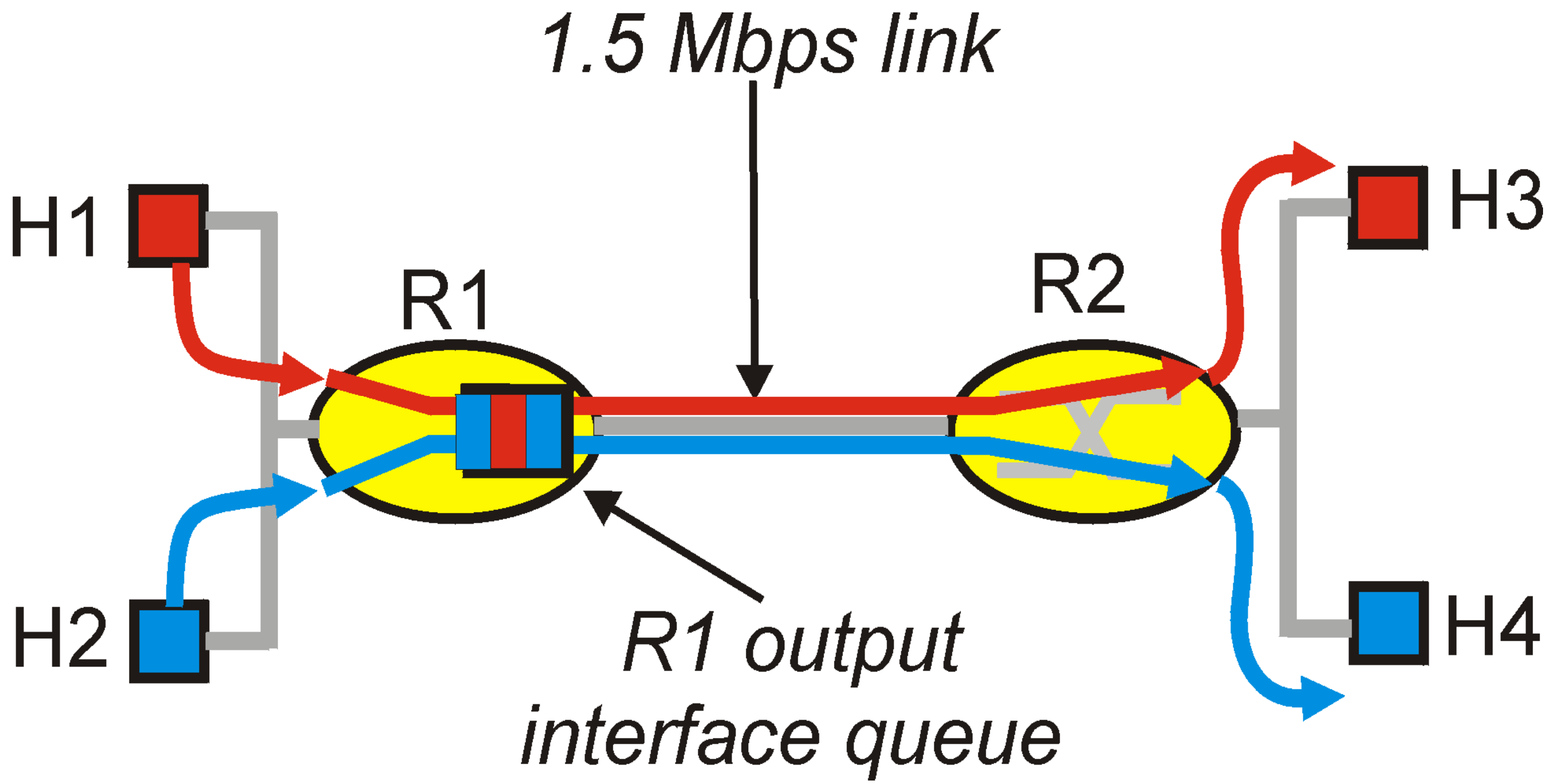
- Some packets fail to reach the destination.
- Packet loss can be caused by a number of factors:
  - Signal degradation
  - Channel congestion
  - Corrupted packets rejected in-transit
  - Faulty networking hardware / routing configurations
- Packet recovering protocols – TCP
- Acceptable packet loss
  - Packet loss does not always indicate a problem

Service Classes

ID	Description	Priority	Duration (s) (min-max)		Data rate (kb/s) (min-max)		BER (min-max)		Delay (ms) (min-max)	
SC1	Large files exchange	8	50MB	500MB	1000	50000	1,00E-06	1,00E-06	200	
SC2	HQ video streaming	6	300	600	2000	40000	1,00E-09	1,00E-09	200	
SC3	LAN access and file service	4	120	300	500	50000	1,00E-06	1,00E-06	100	200
SC4	Interactive ultra high media	1	120	500	1000	50000	1,00E-03	1,00E-06	20	100
SC5	Lightweight browsing	5	300	900	64	512	1,00E-06	1,00E-06	200	
SC6	Data and media telephony	2	60	120	64	512	1,00E-03	1,00E-06	100	200
SC7	Simple telephony and messaging	3	10	120	8	64	1,00E-03	1,00E-06	100	200
SC8	Multimedia messaging	7	5	15	8	64	1,00E-06	1,00E-09	200	

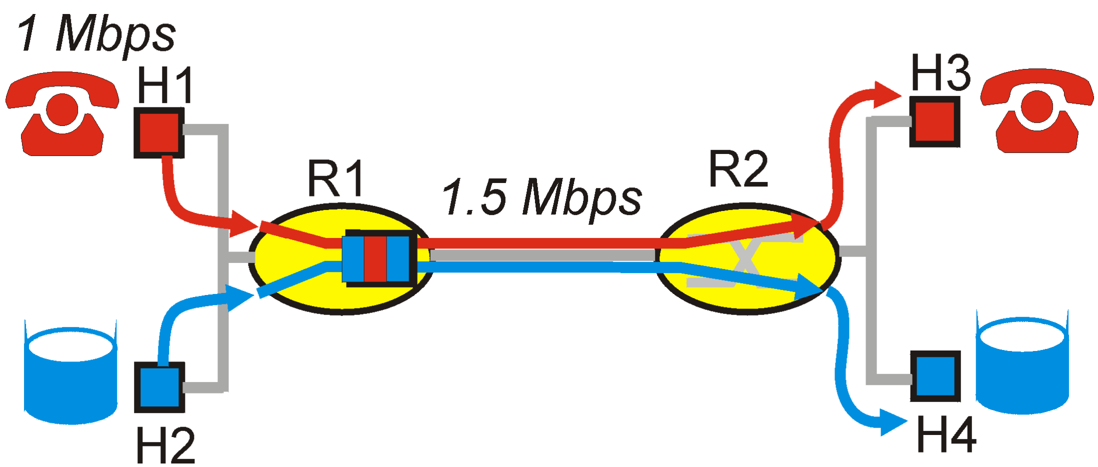
Sharing and Congestion Studies

- Simple model



Principles of QoS Guarantees

- Example: 1Mbps IP phone and FTP share 1.5 Mbps link.
  - Bursts of FTP can congest router, cause audio loss
  - Want to give priority to audio over FTP

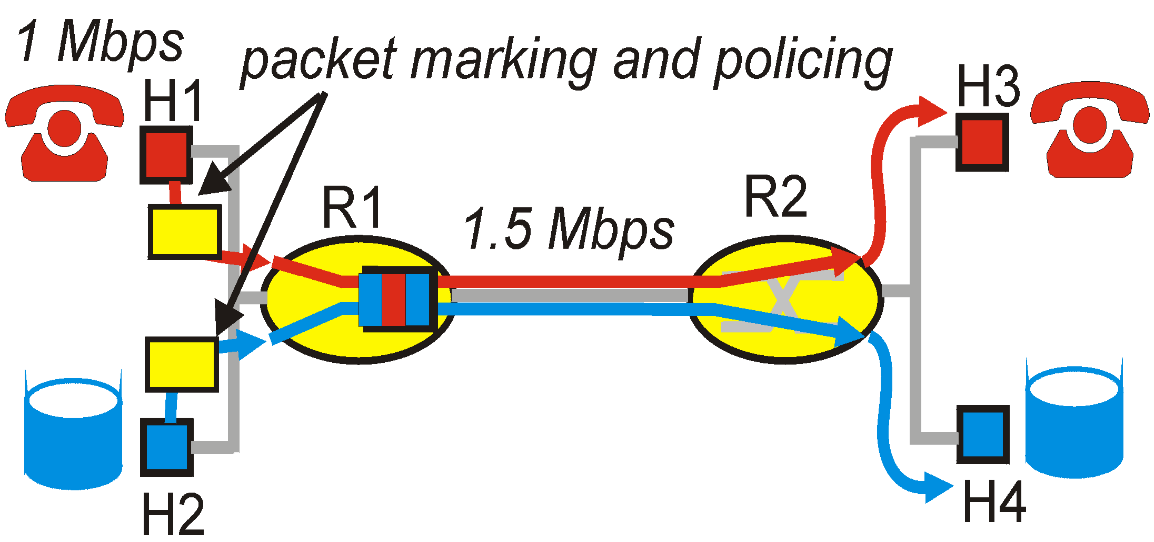


- Principle 1: Packet marking/classification needed for router to distinguish between different classes; and new router policy to treat packets accordingly.

Different Traffic Classification

Principles for QoS Guarantees (more)

- What if applications misbehave (audio sends higher than declared rate)
  - Policing: force source adherence to bandwidth allocations

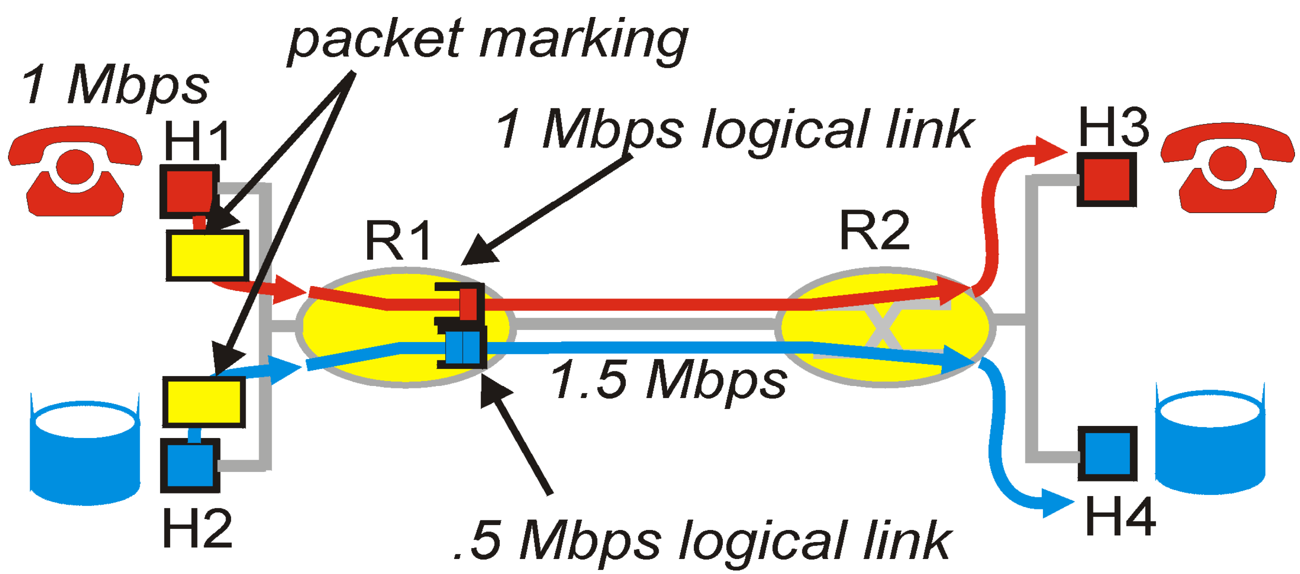


- Principle 2: Provide protection (isolation) for one class from others.

Separate Resources for Different Traffic Classes

Principles for QoS Guarantees (more)

- Allocating *fixed* (non-sharable) bandwidth to flows: *inefficient* use of bandwidth if flows doesn't use its allocation

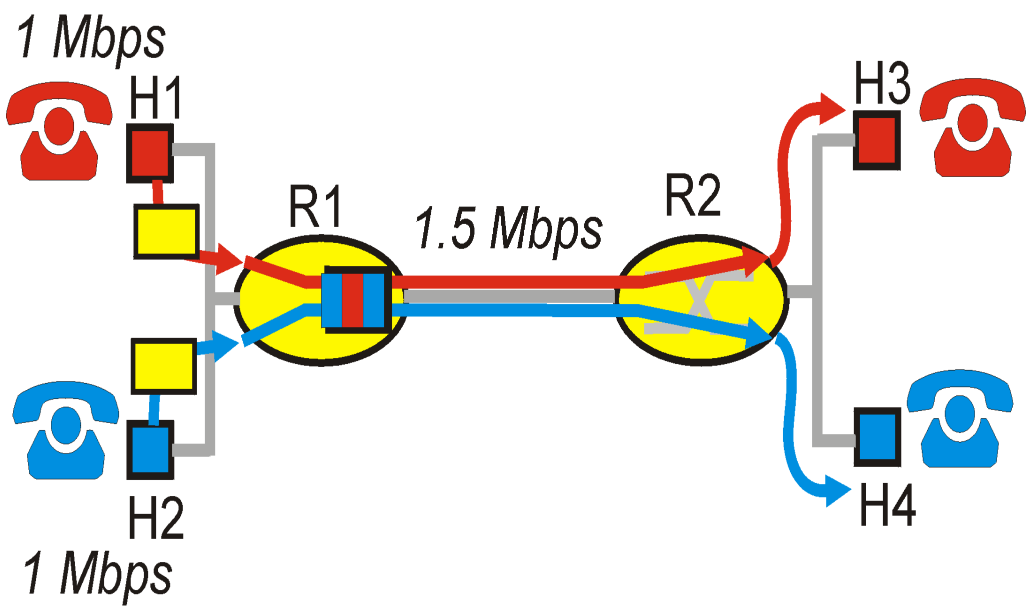


- Principle 3: While providing isolation, it is desirable to use resources as efficiently as possible.

Separate and Flexible Resources for Different Traffic Classes

Principles for QoS Guarantees (more)

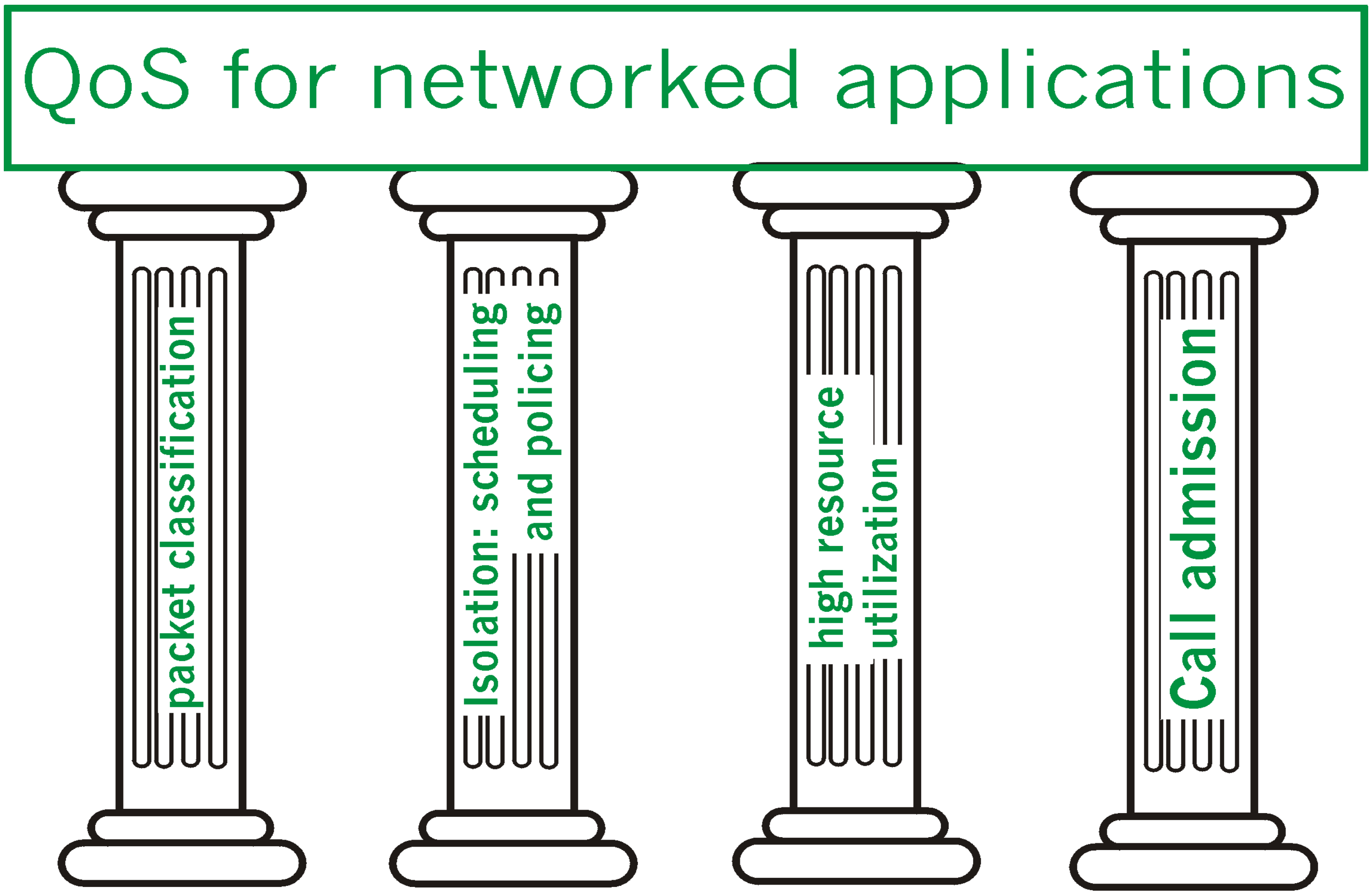
- *Basic fact of life:* can not support traffic demands beyond link capacity.



- Principle 4: Call Admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs.

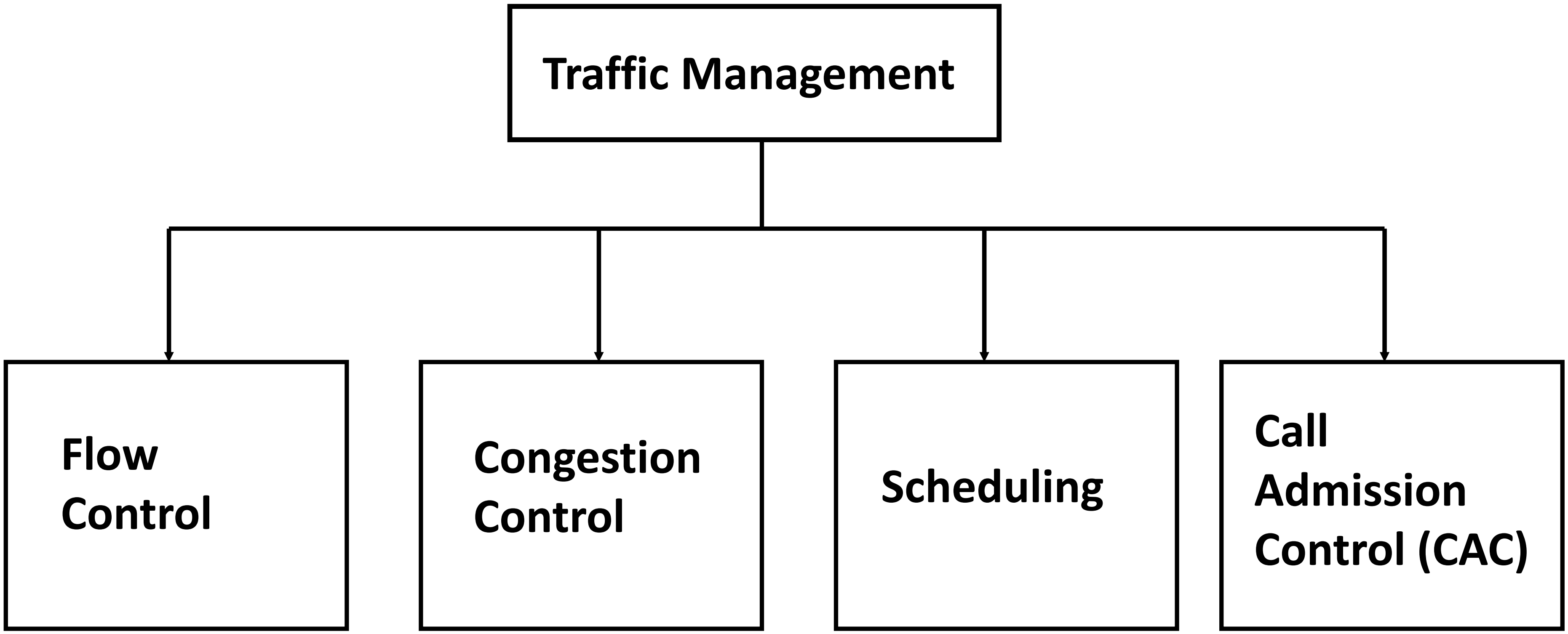
Admission control to ensure sufficient separate and flexible resources for different traffic classes

Summary of QoS Principles



- Let’s next look at mechanisms for achieving this …

Traffic Management



Scheduling Mechanisms

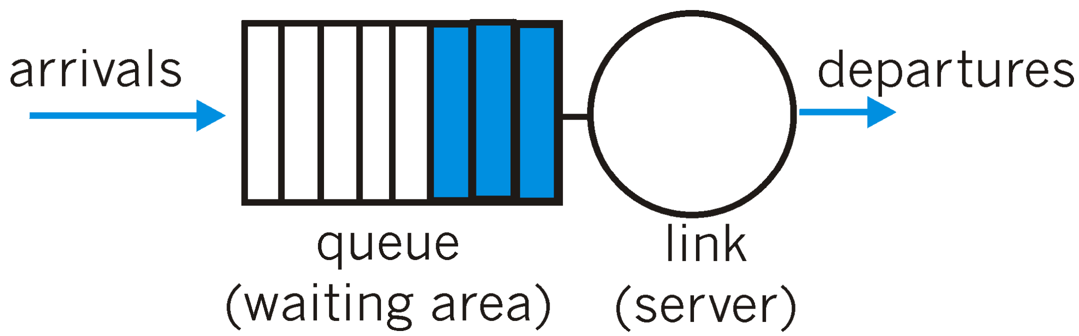
- Scheduling: Choose next packet to send on link

- Examples of Scheduling Mechanism:
  - FIFO- First in First Out
  - Priority Scheduling
  - Round Robin Scheduling
  - WFQ- Weighted Fair Queuing

Scheduling - FIFO

- FIFO (first in first out) scheduling: send in order of arrival to queue.

  - Discard policy: if packet arrives to full queue.
  - Who to discard?
    - Tail drop: Drop arriving packet
    - Priority: Drop/remove on priority basis
    - Random: Drop/remove randomly

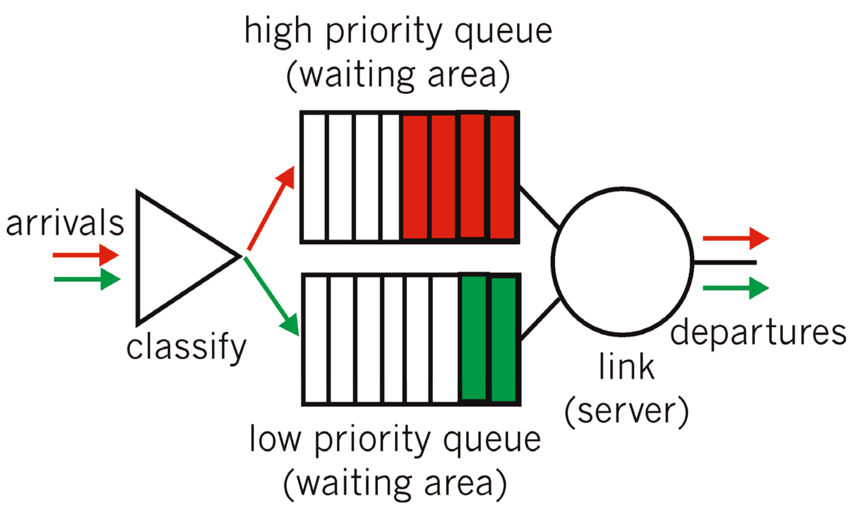


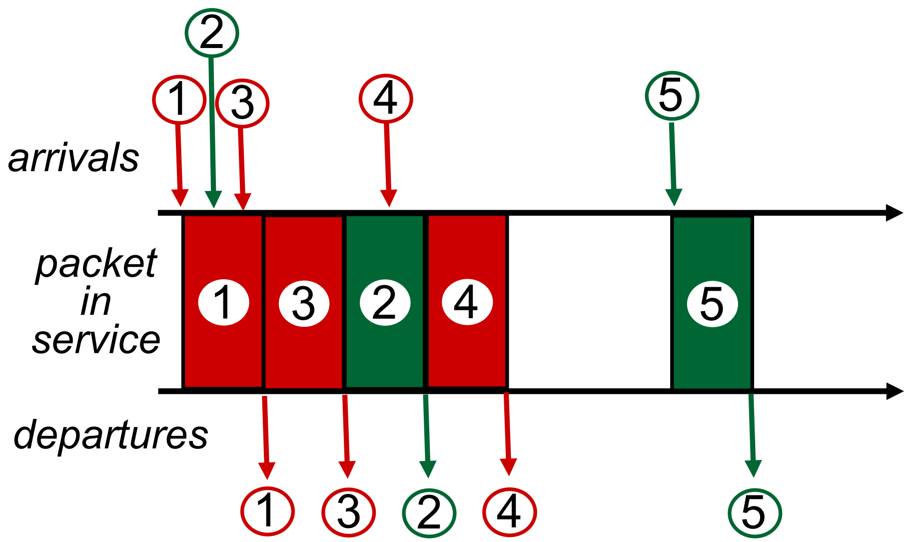
Scheduling - Priority

Priority scheduling*:* Transmit highest priority queued packet

- Multiple classes, with different priorities
  - Class may depend on marking or other header info.

    E.g. IP source/destination, port numbers, etc..





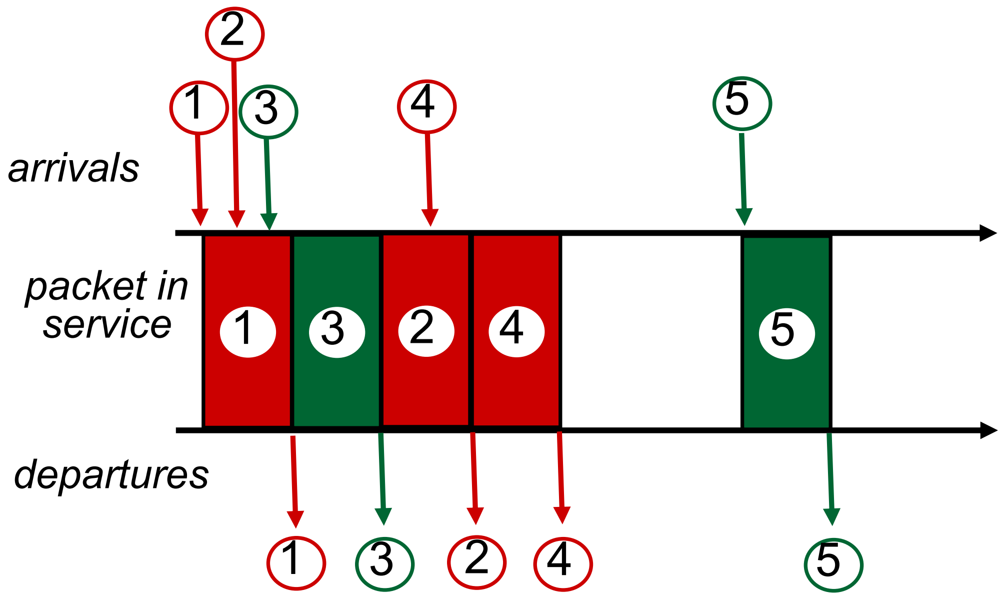
High priority: packet 1,3,4

Low priority: packet 2, 5

Scheduling – Round Robin

Round Robin Scheduling:

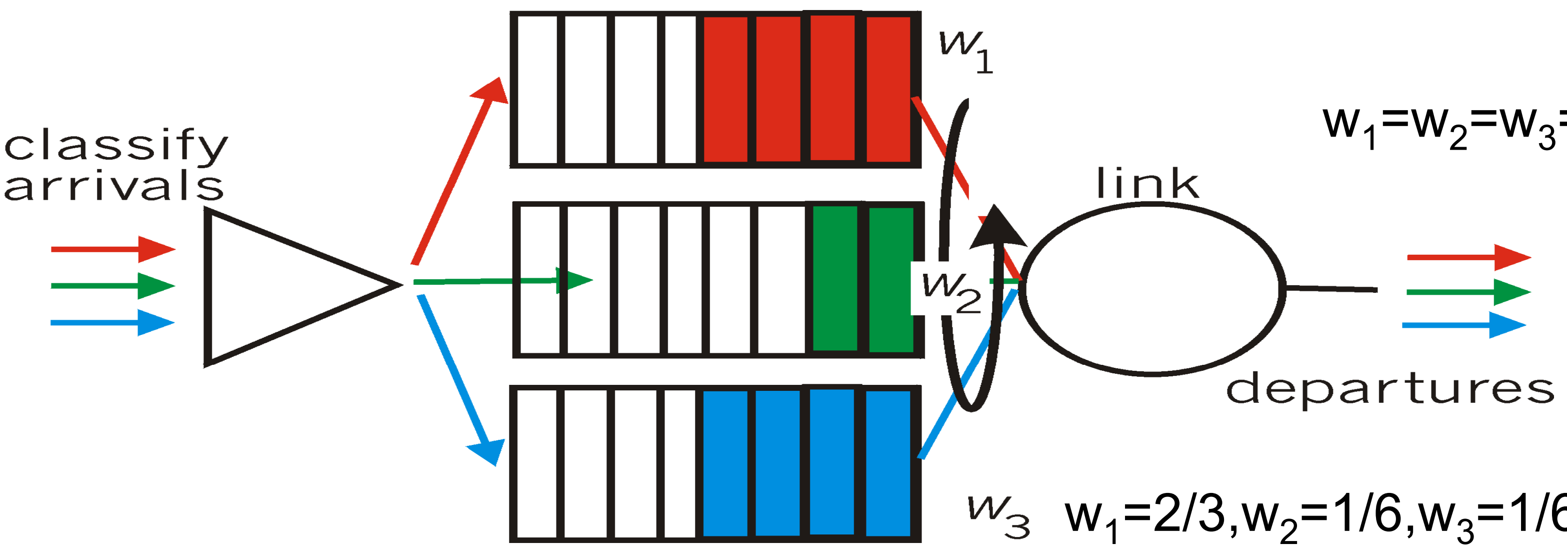
- Packets sorted into classes, but **without** priority class
- Cyclically scan class queues, serving one from each class (if available)
- Example:
  - Class 1: packet 1, 2, 4
  - Class 2: packet 3, 5



Scheduling - WFQ

Weighted Fair Queuing:

- Generalized Round Robin
- each class, *i*, has weight, w_i, and gets weighted amount of service in each cycle: $\frac{w_i}{\Sigma_j w_j}$
- For a link rate of R, each class will then receive a rate of $R \frac{w_i}{\Sigma_j w_j}$



$w_1=w_2=w_3=1/3 \rightarrow$ 123123…RR

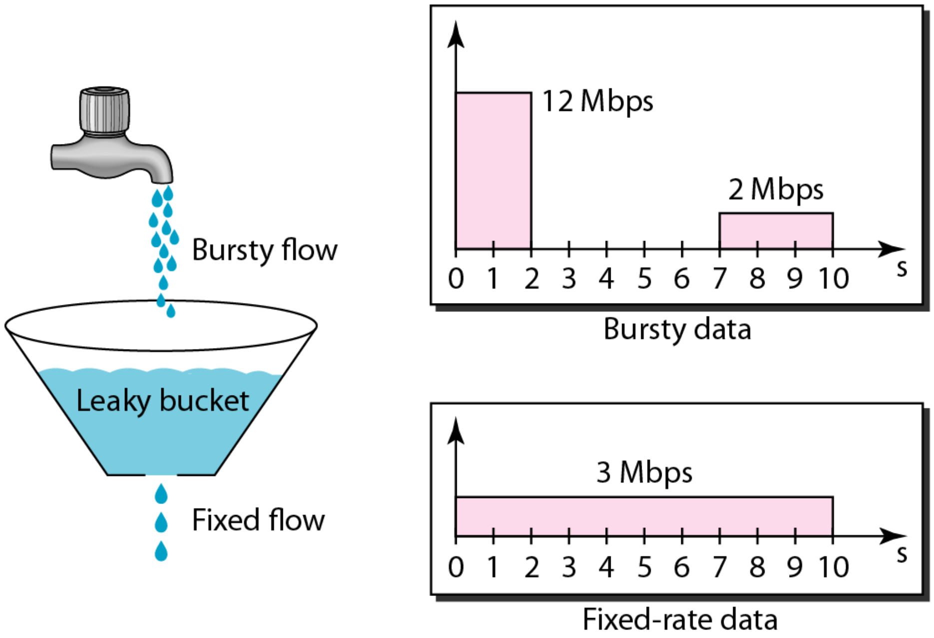
$w_1=2/3, w_2=1/6, w_3=1/6 \rightarrow$ 112311 112311…
or 111123 111123 …

Traffic Shaping

- Controls the amount and the rate of the traffic sent to the network.
- Two techniques can shape traffic:
  - Leaky bucket
  - Token bucket

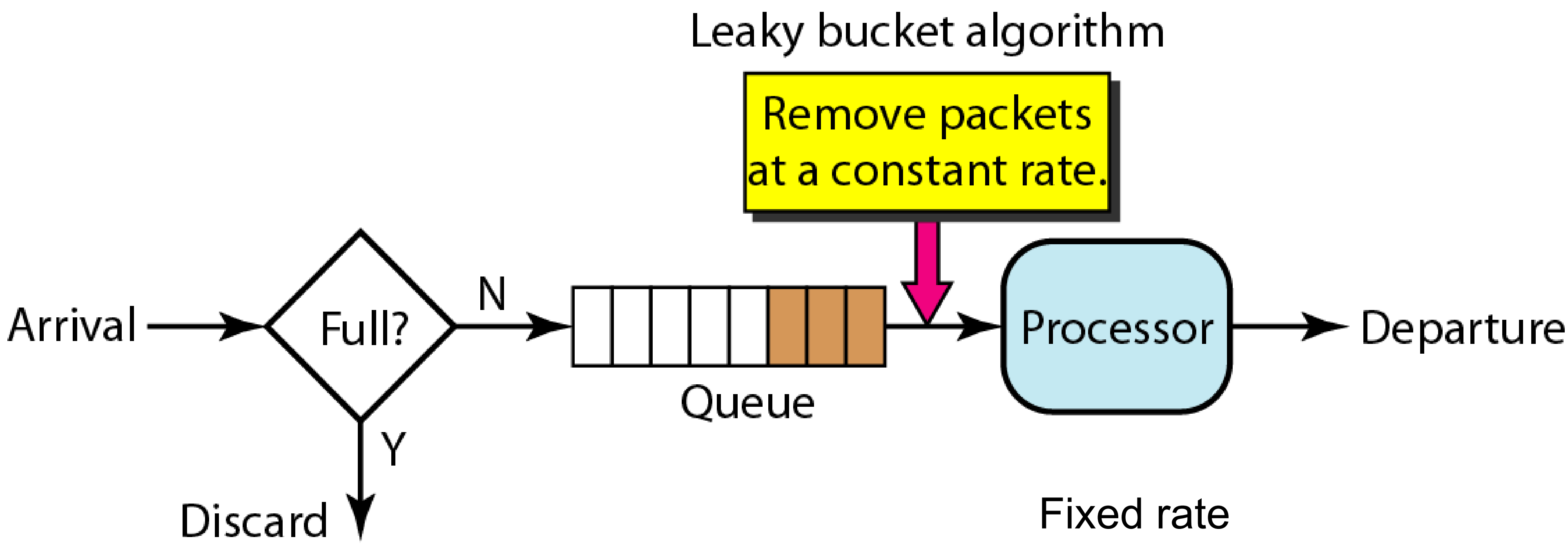
Leaky Bucket

- Shapes bursty traffic into fixed-rate traffic by averaging the data rate
- It may drop packets if the bucket is full
- It also prevents congestion



Leaky Bucket Algorithm

- This is a simple leaky bucket implementation



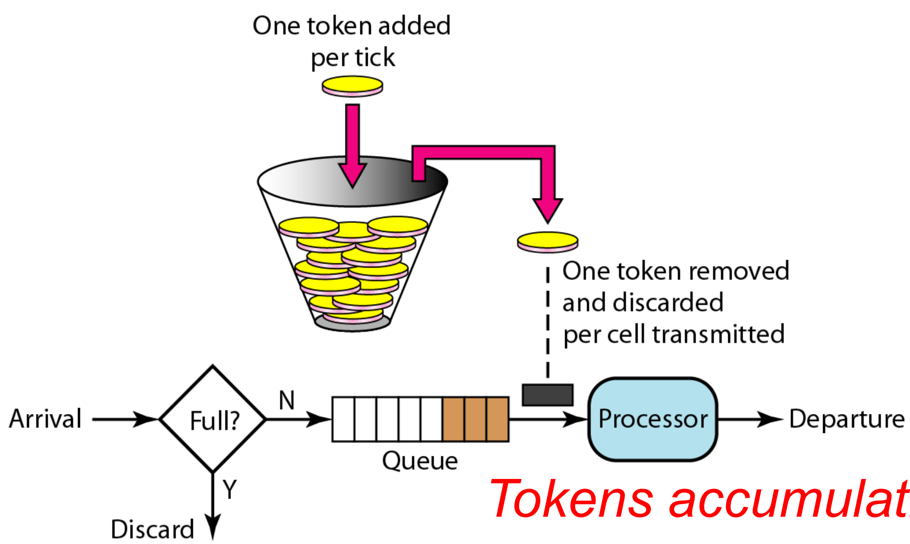
The leaky bucket is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty → buffer/queue status unaware

Token Bucket

- The leaky bucket imposes a hard limit on the data transmission rate.
- The token bucket allows a certain amount of burstiness while imposing a limit on the average data transmission rate.
- The token bucket allows bursty traffic at a regulated maximum rate.
- Allows idle hosts to accumulate credit for the future in the form of tokens → buffer/queue status aware

Token Bucket Algorithm

- Bucket contains tokens
- Tokens are added periodically at desired maximum average rate
- Packets can only be sent when there are tokens in the bucket.
- Tokens are removed when sending packets



Analysis

- Capacity of the bucket: ***c* tokens**
- Tokens rate entering bucket: ***r* tokens/sec**
- The system removes one token for every cell of data sent. The maximum number of cells that can enter the network during any time interval of length *t* is:

Maximum number of packets = $r \times t + c$

- The maximum average rate for the token bucket (departure) is:

Maximum average rate = $(r \times t + c)/t$ **packets per second**

Tokens accumulation when buffer is empty → queue status aware

Resource Management

- Flows need resources
  - Buffer, capacity, CPU time, etc.
  - Ensure availability: call admission
- QoS improved if these resources are guaranteed beforehand
- Resource-reservation models for QoS
  - Integrated Services
  - Differentiated Services

Admission Control

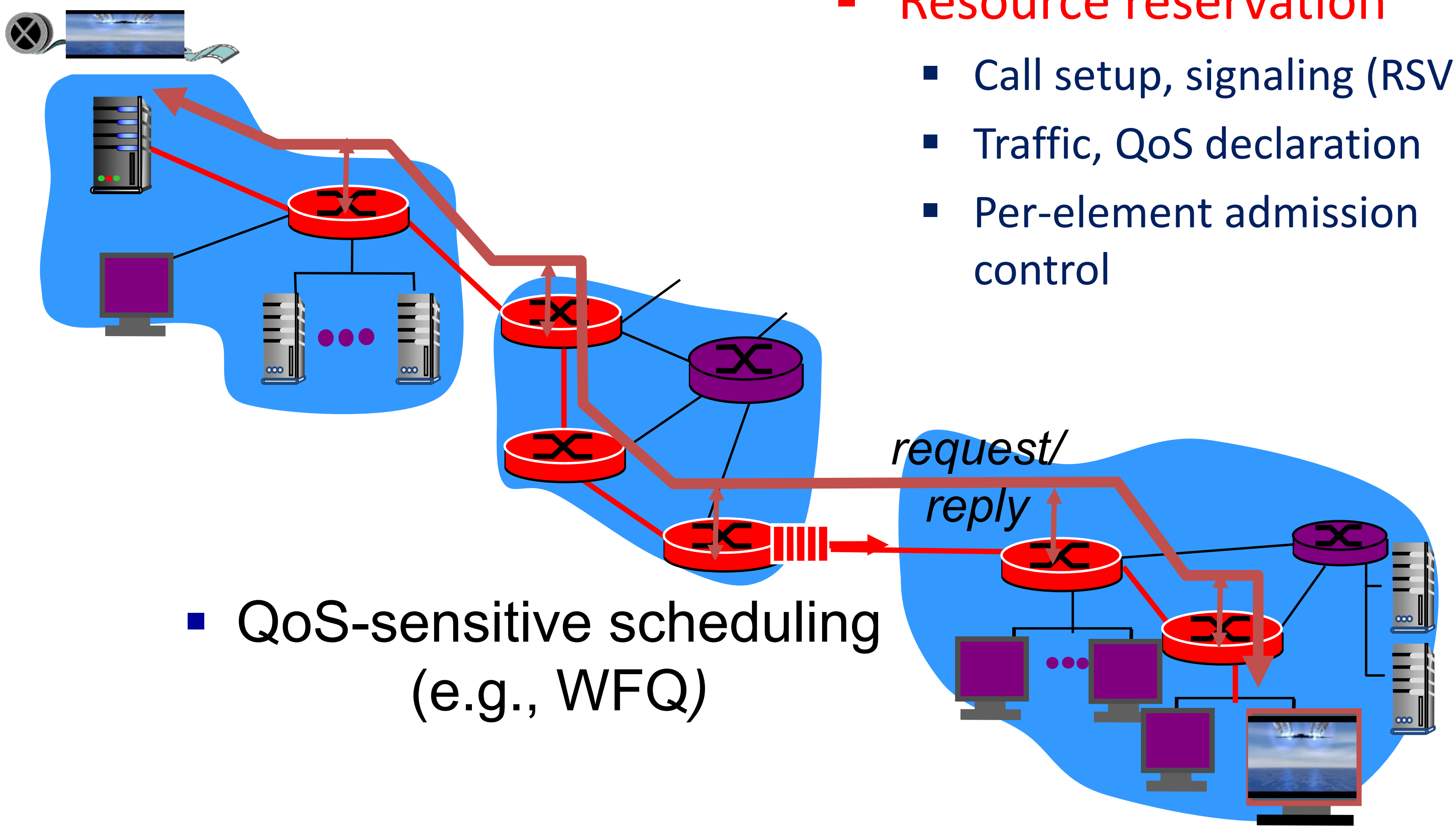
- A mechanism used by a router or switch to accept or reject a flow,
  - based on predefined parameters called flow specifications.

- Before a router accepts a flow for processing, it checks the flow specifications to see
  - If its capacity (in terms of bandwidth, buffer size, CPU speed, etc) can handle the new flow
  - If its previous commitments to other flows can handle the new flow.

Integrated Services

- Traditional Internet only provides best-effort delivery for all users
- Some applications need a minimum bandwidth to function (e.g, real-time video)
- Integrated service (IntServ): architecture for providing QoS guarantees in IP networks for individual application sessions/flows (flow-based).
- Resources are explicitly reserved for a given data flow
  - Routers maintain state info of allocated resources, QoS requests for different flows.
- Admit/deny new call setup requests.

Intserv: QoS Guarantee Scenario



- Resource reservation
  - Call setup, signaling (RSVP)
  - Traffic, QoS declaration
  - Per-element admission control

- QoS-sensitive scheduling (e.g., WFQ)

Call Admission

Arriving session must :

- Declare its QoS requirement
  - R-spec: defines the QoS being requested (buffer, bandwidth, etc.)

- Characterize traffic it will send into network
  - T-spec: defines traffic characteristics of the flow (token bucket algorithm parameters)

- Signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - RSVP (Resource Reservation Protocol)

Intserv QoS: Service Models/Classes

Guaranteed service:

- Designed for real-time traffic that needs a guaranteed minimum end-to-end delay.
- Guarantees that the packets will arrive within a certain delivery time and are not discarded if flow traffic stays within the boundaries of *Tspec*.

Controlled load service:

- Designed for applications that can accept some delays, but are sensitive to an overload network and to the danger of losing packets.

Differentiated Services

- Concerns with *Intserv*
  - Scalability: Router needs to keep information for each flow
  - Service Class Limitations: Only two types,guaranteed and control-load
- Differentiated Services (DiffServ)
  - Main processing moved from the core of the network to the edge of the network (Solves scalability problem)
  - Per-flow service changed to per-class service.

- *IntServ is based on building virtual paths by using the bandwidth reservation technique*
- *DiffServ is based on traffic differentiation/priority without reservation*

Recommended Reading

- J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 8th ed., 2022, Chapter 4