

Network Support for Multimedia

Daniel Zappala

CS 460 Computer Networking
Brigham Young University

Network Support for Multimedia

- *make the best of best effort*
 - use application-level techniques
 - use CDNs for stored content
 - provision the network
 - *simple and broadly supported*
- *differentiated services*
 - treat different classes of traffic differently, but no circuit-like guarantees
 - use packet marking, policing, scheduling
 - *incremental, some deployment*
- *integrated services*
 - per-flow virtual circuits to guarantee bandwidth, delay
 - use packet marking, policing, scheduling, call admission and signalling
 - *needs widespread adoption, little deployment*

Making the Best of Best Effort

- provide bandwidth to eliminate queueing delay and loss
 - **bandwidth provisioning**: given a topology, determine the capacity required on all links to meet performance requirements
 - **network dimensioning**: design a topology (place routers, create links, assign bandwidth to links) to meet performance requirements
- provisioning process
 - model traffic between network end points (e.g. arrival rate of users, arrival rate of packets for a given application)
 - define performance requirements
 - develop a model to predict performance given a workload
 - find a minimal cost bandwidth allocation that will meet all user requirements
- **who will pay the ISPs to install sufficient bandwidth?**
- **will ISPs cooperate to ensure the entire path is properly provisioned?**

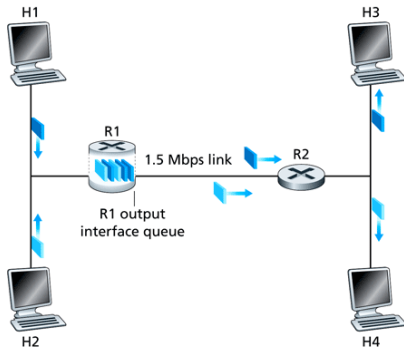
Multiple Classes of Service

Providing Multiple Classes of Service

- simple solution to provide different levels of service to different types of applications
 - analogous to first-class airline service
 - provided for aggregates of traffic, not individual flows
- dates back to 1979, when Type of Service (ToS) bits were provided in the IPv4 header

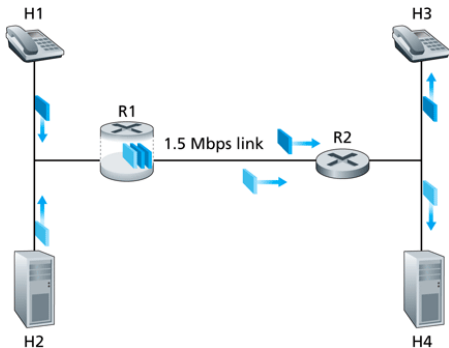
Motivating Scenarios

- we will use the figure below to explore general principles
 - H1 is talking to H3, H2 is talking to H4
 - classic example used for TCP and QoS



Packet Marking

- example: 1 Mbps VoIP, competing with FTP application
 - bursts of FTP packets can cause audio loss
 - want to give priority to audio over FTP
- *packet marking*: allows a router to distinguish among packets belonging to different classes of traffic
 - can be done by application or by router

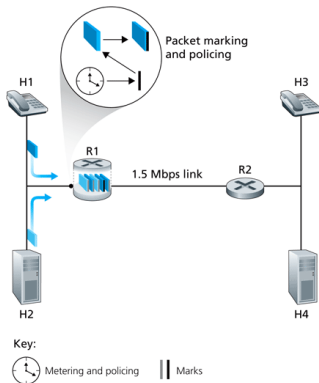


Traffic Isolation

- problems
 - what if audio application sends as many high-priority packets as it wants?
 - what if many audio applications share the same class but one is not well behaved?
- *traffic isolation* : protect classes and flows from each other so that they are not adversely affected when someone misbehaves
- two approaches: *policing* and *resource reservation*

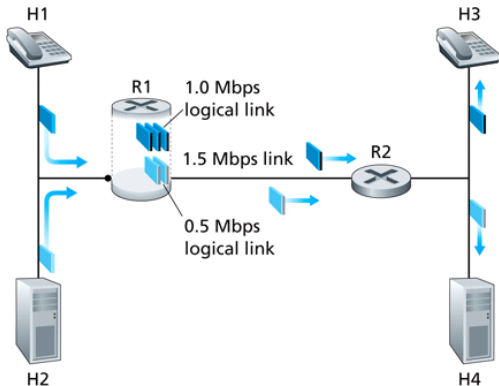
Policing

- **policing**: enforce a rate on a network flow, usually at network edge
- consequences if you exceed your rate: marking or dropping extra packets



Resource Reservation

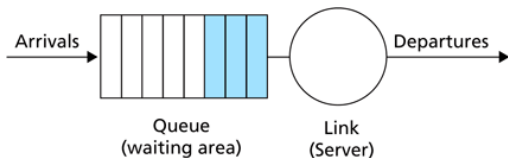
- **resource reservation**: allocate resources (e.g. bandwidth) to each flow
- ideally, if audio application goes silent, data application can utilize unused bandwidth



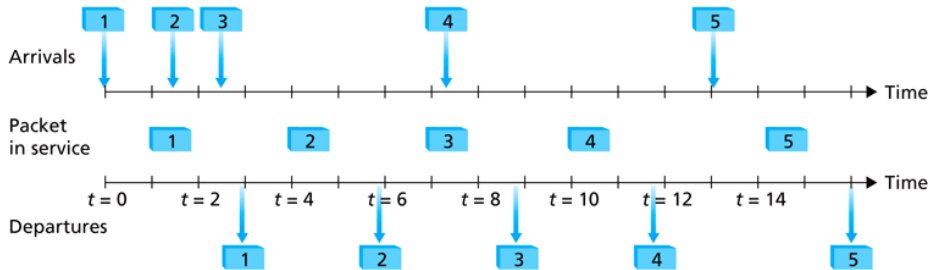
Scheduling

FIFO: First In First Out

- transmit in order of arrival to queue
- **discard policy**: if packet arrives and queue is full, which packet should the scheduler discard?
 - drop-tail: drop the arriving packet
 - priority: remove a packet based on priority
 - random: remove a packet randomly

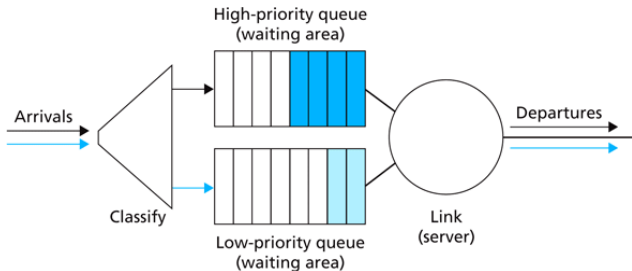


FIFO in Operation

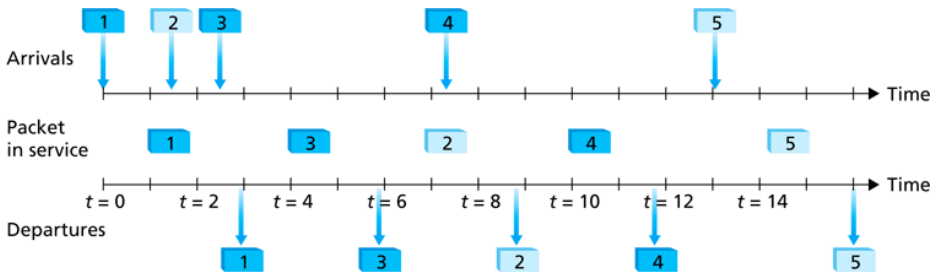


Priority Queuing

- transmit the highest priority queued packet
- may have multiple classes with different priorities, based on marking or IP addresses, ports

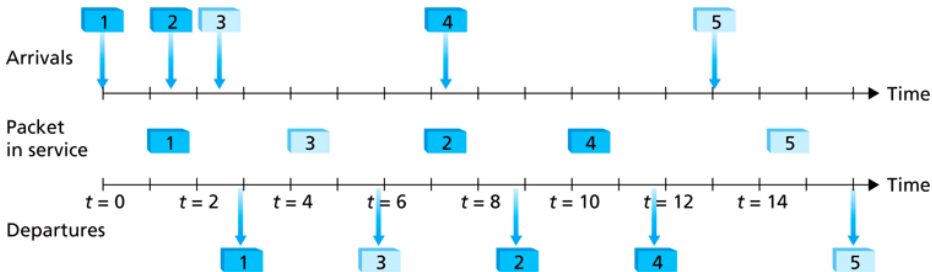


Priority Queue in Operation



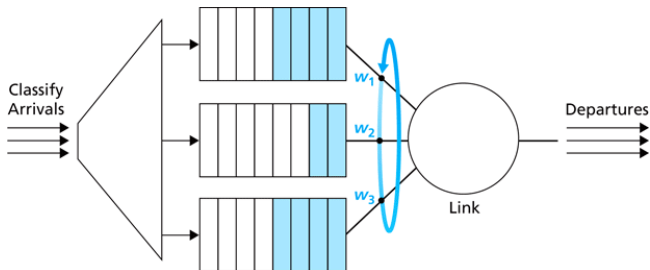
Round Robin Scheduling

- round robin among multiple classes of traffic, either through marking or IP header



Weighted Fair Queuing

- generalized round robin among classes or flows of traffic
- each flow is weighted with w_i and is guaranteed to receive throughput of $R * w_i / \sum w_j$
- takes into account packet sizes



Work-Conserving

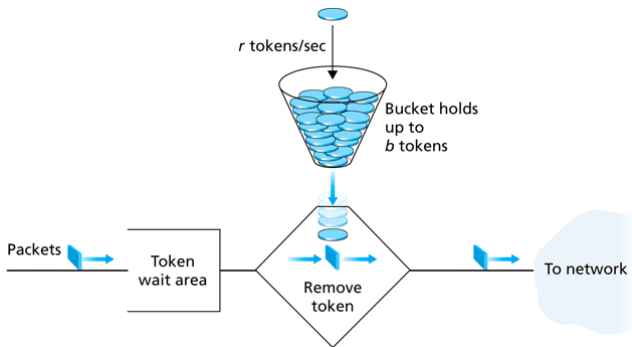
- **work-conserving**: indicates that a scheduling algorithm will never allow the link to become idle if there are packets to be sent
 - FIFO
 - priority queuing
 - WFQ
- **non-work-conserving**: link may be left idle even when there are packets ready
 - round robin : if each class allocated a fixed number of slots (similar to TDMA)

Policing

Policing Parameters

- goal: limit traffic to a set of declared parameters
- common parameters:
 - **average rate**: number of bits per second, over some time interval
 - 100 bits/second = 6000 bits/minute
 - what interval do you want to average over?
 - **peak rate**: highest rate allowed, over some time interval
 - typically over a shorter interval
 - example: 256 kbps average per minute, 1 Mbps peak per second
 - **burst size**: number of packets sent consecutively

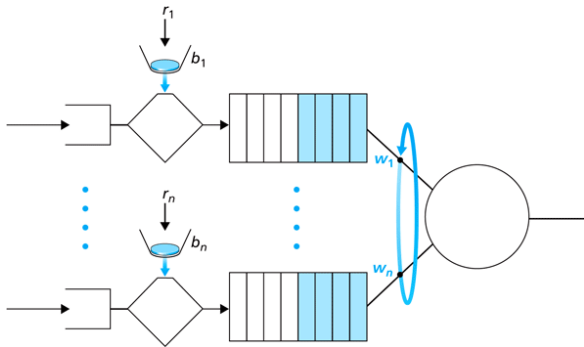
Leaky (Token) Bucket



- tokens generated at rate of r tokens/s
- bucket holds at most b tokens, excess is lost
- maximum burst size: b
- average rate over interval t : $rt + b$

Guaranteed Delay

- leaky bucket + weighted fair queuing = provable maximum delay in a queue
- $d_{max} = \frac{b_i}{R * w_i / \sum w_j}$



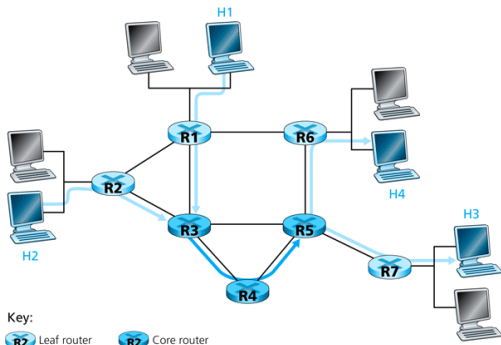
Differentiated Services

Differentiated Services

- scalability
 - need to handle hundreds of thousands of simultaneous flows
 - keep the network core simple, with complexity at the edges
- flexibility
 - may develop new services class or stop using old classes
 - provide functional components, rather than specific classes
 - generic ability to map flows into classes

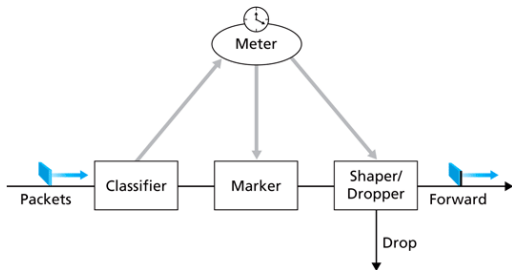
Architecture

- **edge routers:** packet classification and traffic conditioning (policing)
- **core routers:** schedule packets based on traffic class (per-hop behavior), no discrimination between flows in the same class



Packet Classification and Traffic Conditioning

- occurs at edge routers
- classifier marks traffic class
- marker marks packets in or out of profile based on a pre-negotiated rate (e.g. leaky bucket)
- shaper may force flow to conform to bucket parameters or drop some traffic



Per-Hop Behaviors

- forwarding behaviors performed at core routers
- resulting service must be measurable
- expedited forwarding
 - rate given to a class must equal or exceed a specified rate
 - provides a logical link with a minimum guaranteed rate to a particular class of traffic
- assured forwarding
 - divides traffic into four classes, each with a minimum guaranteed rate
 - each class has three drop preference categories
 - provides a way for customers to allocate traffic into classes and to exceed traffic specifications while also specifying which packets get dropped first

Integrated Services

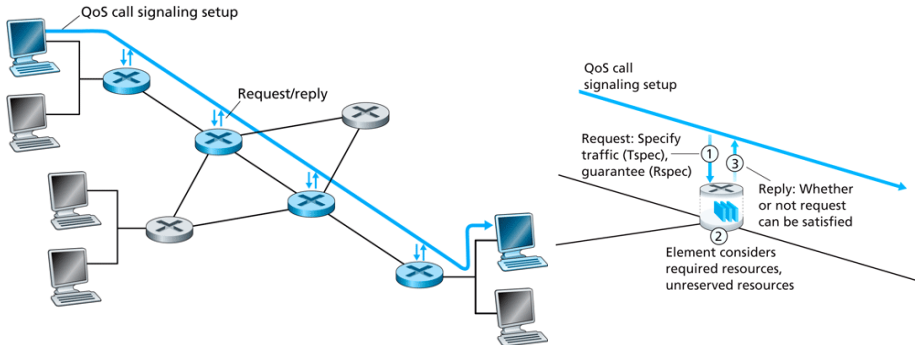
Integrated Services

- goal: provide quality of service guarantees to individual connections on the Internet
- major components
 - scheduling: meet performance requirements *at each hop*, e.g. with Weighted Fair Queueing
 - admission control: decide whether there are enough resources for the connection *at each hop*
 - resource reservation: a setup protocol to get approval from admission control and reserve resources with the scheduler *along a path*
 - QOS routing: a routing protocol that can find a path with available resources
 - pricing and billing: charge users for the resources they use

RSVP

- Resource ReSerVation Protocol: connection setup protocol for the Internet
- features
 - resource reservations for multicast trees
 - receiver-oriented: group member initiates and maintains reservation on the tree
 - soft state: reservations must be refreshed or else they will expire
- source must specify a traffic specification (e.g. leaky token bucket parameters) and a resource specification (e.g. minimum bandwidth, maximum delay)
- network provides guaranteed bandwidth + leaky bucket = guaranteed delay

The Call Setup Process



Retrospective

Retrospective

- We've been trying to provide QoS on the Internet for 20 years – why have we failed?
 - to provide end-to-end integrated or differentiated services, all ISPs between two hosts must provide the service and have service agreements between each other (e.g. agree to carry traffic and give it the specified service, for a fee)
 - need to police, shape, schedule, admit traffic – extra deployment and management challenges
 - need to bill users, probably by volume of traffic
- under low or moderate load, we can get most of the benefits of QoS using best-effort service!
- provisioning is the easy way out