Lecture 21b
Quality of Service

Peter Steenkiste
Departments of Computer Science and
Electrical and Computer Engineering
Carnegie Mellon University

15-441 Networking, Spring 2006
http://www.cs.cmu.edu/~prs/15-441

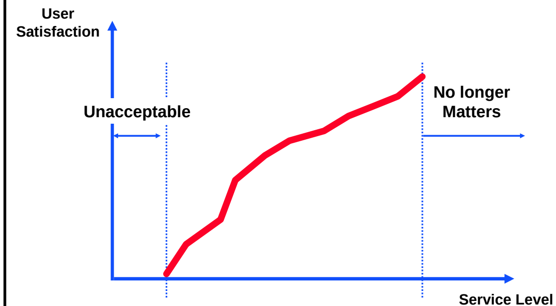
Outline

- What is quality of service?
- QoS principles and mechanisms.
- Example QoS service models.
- RSVP.

What is QoS?

- Current Internet supports best effort packet delivery only.
  - » sufficient for most applications, but some applications require or can benefit from a "higher" level of service
- "Higher" quality of service can mean that bounds are provided for one or more performance parameters.
  - » Bandwidth: fast data transfers, video
  - » Delay, jitter: telephony
  - » Packet loss, bit error rate: update services
- QoS can also mean that a user gets "better" treatment.
  - » But no guarantees are given

Performance versus Satisfaction



Quality of Service versus Fairness

- Traditional definition of fairness: treat all users equally.
  - » For example, max-min fairness: all users sharing the same bottleneck link get the same bandwidth
- QoS: treat users differently.
  - » For example, some users get a bandwidth guarantee, while others have to use best effort service
- The two are not in conflict.
  - » All else being equal, users are treated equally
  - » Unequal treatment is based on policies:
    - – Administrative policies: rank or position
    - – Economics: extra payment for preferential treatment

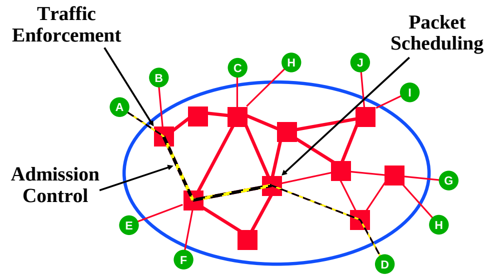
QoS Analogy: Surface Mail

- The defaults if "first class mail".
  - » Usually gets there within a few days
  - » Sufficient for most letters
- Many "guaranteed" mail delivery services: next day, 2-day delivery, next day am, ...
  - » Provide faster and more predictable service at a higher cost
  - » Providers differentiate their services: target specific markets with specific requirements and budgets
- Why don't we do the same thing in networks?

How to Provide QoS?

- Admission control limits number of users.
  - » You cannot provide guarantees if there are too many users sharing the same set of resources (bandwidth)
  - » For example, telephone networks - busy tone
  - » This implies that your request for service can be rejected
- Traffic enforcement limits how much traffic users can inject based on predefined limits.
  - » Make sure user respects the traffic contract
  - » Data outside of contract can be dropped (before entering the network!) or can be sent at a lower priority
- Scheduling support in the routers guarantee that users get their share of the bandwidth.
  - » Again based on pre-negotiated bounds

QoS Framework

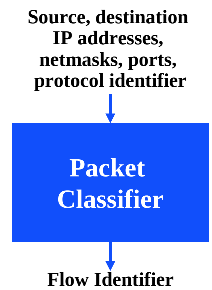


Some Simple QoS Mechanisms

- Classification.
  - » Packet filters
- Scheduling.
  - » Weighted fair queueing
  - » Hierarchical scheduling
- Traffic enforcement.
  - » Leaky buckets
  - » Shapers versus meters
- Admission control.

How Do We Distinguish Between Flows?

- For every packet we must be able to determine what flow (or user) it belongs to.
  - » First step towards giving it appropriate service
- Packet classifier takes as input the packet header and generates a flow identifier.
  - » The classifier has to be initialized, for example as part of admissions control
  - » Classification is a hard problem
- Later stages in the router can use the flow id to customize service.

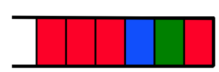


Sharing versus Isolation

- FIFO: sharing
  - » each traffic source impacts other connections directly
    - – e.g. malicious user can grab extra bandwidth
  - » the simplest and most common queueing discipline
  - » averages out the delay across all flows
- Priority queues: one-way sharing
  - » high-priority traffic sources have impact on lower priority traffic only
  - » has to be combined with admission control and traffic enforcement to avoid starvation of low-priority traffic
- WFQ: two-way isolation
  - » provides a guaranteed minimum throughput (and maximum delay)

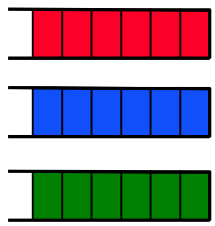
FIFO

- Simplest queueing discipline.
  - » Single FIFO
  - » No decision to be made (one choice)
- Widely used in today's routers.
- Key property: treats all *packets* equally.
  - » But does not necessarily treat all *users* equally
  - » Poorly implemented or malicious users can take bandwidth from other users



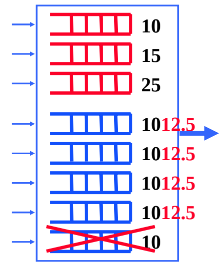
Priority

- Separate packets based on their “importance”.
  - » Can implement many policies: user, application, traffic quota, ..
- Scheduler always serves high priority queue first, if not empty.
  - » Very simple scheduler
- Risk is starvation.
  - » Low priority traffic may never get service
  - » Violates people’s assumption about best effort service
  - » Solution is admission control for higher priority traffic



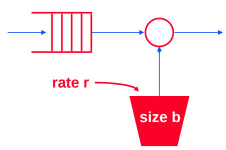
Weighted Fair Queueing WFQ

- Distributed bandwidth to flows according to some agreed upon distribution.
  - » Routers are given a set of weights during signaling
- WFQ can support bandwidth reservations.
  - » Adjust the weights so that guaranteed flows get their bandwidth
  - » Other flows share bandwidth according to some sharing formula (e.g. equally)
  - » Weights have to be adjusted when flows leave and enter



Traffic Enforcement: A Token Bucket

- We have to limit the traffic that each user can inject in the network while still allowing a certain degree of burstiness.
  - » Both rate and burstiness have to stay within agreed upon bounds
- Token bucket supports enforcement using two independent parameters.
  - » Long-term average rate
  - » Size of the largest burst
- Can be used to meter traffic, shape the traffic stream, or characterize the traffic.



QOS Admission Control

- Client submits request.
  - » description of traffic source
  - » service requested from network
  - » can include desirable and acceptable levels of service
- Traffic descriptors.
  - » Specifies a traffic service class
  - » set of parameters describing service or traffic
- Network checks whether the request can be satisfied and accepts or rejects the request.
  - » May have to check with the routers in the network
  - » May have to reserve resource on every router along the path that was selected for the flow

Outline

- What is quality of service?
- QoS principles and mechanisms.
- Example QoS service models.
- RSVP.

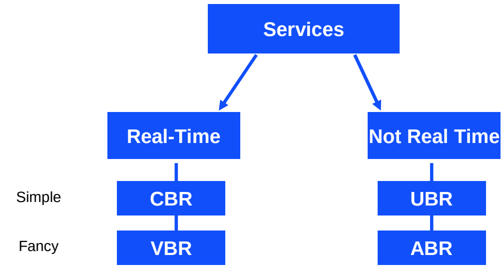
A Short History of Internet QoS

- Lots of initial research in the late 80s and early 90s.
  - » Often takes a telecommunications view of the network
- ATM QoS and IETF Integrated services were developed based on these results.
  - » Focus on per-flow, hard QoS
  - » Effort was driven by perceived application needs
- Focus later shifted to differentiated services.
  - » Focus is on QoS for flow aggregates, e.g. all the flows belonging to one customer
  - » The immediate user is really the network manager

Motivation
Application Types

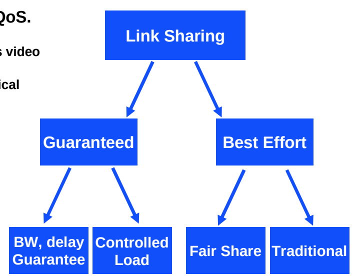
- Elastic applications.
  - » wide range of acceptable rates, although faster is better
  - » e.g. data transfers such as FTP
- Continuous media applications.
  - » lower and upper limit on acceptable performance
  - » sometimes called “tolerant real-time” since they can adapt to the performance of the network
    - – e.g. changing frame rate of video stream
    - – “network-aware” applications
- Real time applications.
  - » require hard limits on performance - “intolerant real-time”
  - » “unacceptable” means “very bad news”
  - » e.g. control applications, ..

ATM Model



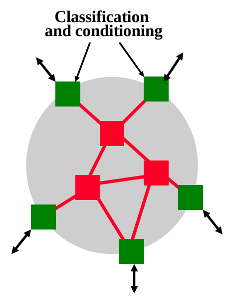
IETF Integrated Services

- Focus on per-flow QoS.
  - » Support specific applications such as video streaming
  - » Based on mathematical guarantees
- Many concerns.
  - » Complexity?
  - » Scalability?
  - » Business model?
  - » Charging?



IETF Differentiated Services:
Motivation and Design

- Do fine grain enforcement only at the edge of the network.
  - » Typically slower links at edge
  - » E.g. mail sorting in post offices
- Label packets with a type field.
  - » E.g. a priority stamp
- The core of the network uses only the type field for QoS management.
  - » Small number of types with well defined forwarding behavior
  - » Can be handled fast
- Example: expedited service versus best effort.
- Evolution rather than revolution.



Differentiated Services:
Discussion

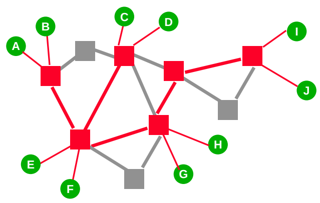
- DiffServ defines an architecture and a set of forwarding behaviors.
  - » It is up to the service providers to define and implement end-to-end services on top of this architecture
  - » Offers a more flexible service model: different providers can offer different services
- One of the main motivations for DiffServ was scalability.
  - » Keep the core of the network simple
- Focus of DiffServ is on supporting QoS for flow aggregates.
  - » Although architecture does not preclude more fine grain guarantees

Charging for Network Service

- Flat charge: you pay a flat fee, independent from usage.
  - » Garbage pick up (?)
- Usage charging: you pay for what you use.
  - » Water, electricity, ...
- Practice: you pay for the thickness of the pipe.
  - » Is equal to potential use or peak use
  - » Also needed: bilateral agreements between ISPs to deal with cross ISP traffic
- A lot of the early work on QoS would require usage based charging, if deployed.
  - » E.g., pay on a per connection basis
  - » Concerns for increased complexity and cost

Resource Reservation Protocol RSVP

- Signaling protocol that establishes connections in the Internet.
  - » IntServ, DiffServ
- Main goal: establish "state" in each of the routers so they "know" how they should treat flows.
  - » State = packet classifier parameters, bandwidth reservation, ..
  - » Uses periodic refresh to deal with failures and recovery
  - » Based on receiver initiated operation (like multicast)



RSVP Motivation and Goals

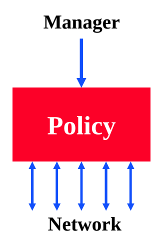
- Resource reservation mechanism for multi-point applications: video and voice conference, shared white board, ..
  - » A strong emphasis on multicast and large scale applications
- Accommodate heterogeneous receivers.
- Adapt to changing membership.
- Exploit application characteristics to optimize use of network resources.
- Allow receivers to switch "channels".
- Adapt to changes in underlying routes.
- Limit control overhead (scaling).

Key RSVP Properties

- Receiver initiates reservation by sending a reservation over the sink tree.
  - » assumes multicast tree has been set up previously
  - » uses existing routing protocol, but routers have to store the sink tree (reverse path from forwarding path)
- Soft state: Periodic path and reservation messages refresh information.
  - » adapts to changes routes and sources
  - » recovers from failures
  - » old information times out
- Temporal resource sharing: not all sources are active at the same time and not all receivers tune in to all active sources.
  - » Single shared reservation for all down stream receivers

Policy Issues

- Any QoS solution must include mechanisms for implementing a range of policies.
  - » Who gets to use what services?
  - » ISP: payment drives service
  - » In other environment, administrative policies are needed
- One components is to use a directory service for policy management.
  - » Managers can enter policies
  - » Network entities can retrieve the policies they need



Common Open Policy Service (COPS) Protocol

- Routers have a "Policy Enforcement Point".
  - » Is responsible for enforcing policies when a request for resources is handled
  - » Example: RSVP router performing policy based admission control
- COPS defines the interaction between the PEP and the policy server.
  - » PEP issues policy request through an RPC
  - » Interaction is based on long-term session that can have state at both ends
    - – asynchronous streaming of information to PEP
    - – caching information on the PEP