CSE/EE 461 Lecture 22
Quality of Service

Tom Anderson
tom@cs.washington.edu
Peterson, Chapter 6.5

Quality of Service

- What kinds of service do different applications need?
  - Web is built on top of “best-effort” service
  - Other applications may need more
    - Internet telephone service (voice over IP)
    - Streaming audio/video
    - Real-time games
    - Remote controlled robotic surgery

- What mechanisms do we need to support these more demanding applications?
  - As with multicast, will need network to do more
An Audio Example

- Playback is a real-time service
  - audio must be received by a deadline to be useful
  - buffering can allow small variations in bw, delay

Tolerating Jitter with Buffering

- Insert variable delay before playout to give time for late samples to arrive
**Taxonomy of Applications**

- **Applications**
  - Real time
  - Elastic
    - Tolerant
      - Adaptive
      - Non-adaptive
    - Intolerant
      - Interactive
      - Interactive bulk
      - Asynchronous
  - Delay adaptive
  - Rate adaptive

Tolerance of loss/late packets  
Adaptivity to rate/delay change

---

**Adapting to Network Change**

- Adapt to changes in network behavior by exploiting application-specific techniques
Roadmap – Various Mechanisms

|简单易建，弱保证   | FIFO with Drop Tail | Classic Best Effort |
|复杂易建，强保证   | FIFO with RED      | Congestion Avoidance |
|                   | Weighted Fair Queuing | Per Flow Fairness |
|                   | Differentiated Services | Aggregate Guarantees |
|                   | Integrated Services | Per Flow Guarantees |

What’s in a Router?

- By convention, draw input ports on left, output on right. (But in reality a single physical port handles both directions.)
Model of a Router

Scheduling and Buffer Management

- Two different functions implemented at the queue

- A scheduling discipline
  - This is the order in which we send queued packets
  - Examples: FIFO or priority-based

- A buffer management policy
  - This decides which packets get dropped or queued
  - Examples: Drop tail, random drop, or per flow
FIFO with Tail Drop

- Arriving packet
  - Next free buffer
  - Next to transmit
  - Free buffers
  - Queued packets

- Arriving packet
  - Next to transmit
  - Drop

Incipient Congestion at a Router

- Sustained overload causes queue to build and overflow

Queue length

- Instantaneous
- Average

Time
Random Early Detection (RED)

- Routers monitor average queue and send “early” signal to source by dropping a packet

![Diagram showing MaxThreshold, MinThreshold, and AvgLen]

- Paradox: early loss can improve performance!

Red Drop Curve

- Start dropping a fraction of the traffic as queue builds
  - Expected drops proportional to bandwidth usage
  - When queue is too high, revert to drop tail
  - Nice theory, difficult to set parameters in practice

![Diagram showing P(drop) and Average Queue Length]
**RED Penalty Box**

- FIFO is not guaranteed (or likely) to be fair
  - If some hosts don’t play by the rules, they can grab more bandwidth
- Neither is RED
  - Senders can still ignore packet loss signals
- One solution: penalty box
  - Keep track of flows sending faster than average
  - Preferentially drop packets for those flows
  - After drop, verify that flow reduced its rate
    - If not, drop more of its packets

**Fair Queuing (FQ)**

- Fair Queuing is an alternative approach
  - Maintain one queue per traffic source (flow) and send packets from each queue in turn
    - Simulate round robin since packets are different sizes
  - Provides each flow with its “fair share” of the bandwidth, no matter what any flow does
  - However, bandwidth per flow can change if number of flows increases/decreases
- Weighted Fair Queueing (WFQ)
  - Proportionately increase rate given to certain flows
Fair Queuing

Round-robin service

Flow 1
Flow 2
Flow 3
Flow 4

Fair Queuing and WFQ

- Want to proportionally share bandwidth
  - At the “bit” level, but must send whole packets
- Approximate with finish times for each packet
  - finish (F) = arrive + length*rate; rate depends on # of flows, weight
  - Send in order of finish times, except don’t preempt transmission if a new packet arrives that should go first
Supporting QOS Guarantees

- Flowspecs. Formulate application needs
  - Need descriptor (token bucket) for guarantee
- Admission Control. Decide whether to support a new guarantee
  - Network must be able to control load to provide guarantees
- Signaling. Reserve network resources at routers
  - Analogous to connection setup/teardown, for router reservations
- Packet Scheduling. Implement guarantees
  - Various mechanisms can be used, e.g., explicit schedule, priorities, WFQ, …

Specifying Bandwidth Needs

- Problem: Many applications have variable demands
- Same average bandwidth, but very different needs over time
  - Example: MPEG compression rate depends on how much changes from frame to frame
Token Buckets

- Simple model
  - Reflects both average, variability over time
- Use tokens to send bits
- Avg bandwidth is $R$ bps
- Maximum burst is $B$ bits

Bucket size $B$ tokens
Fill rate $R$ tokens/sec
Sending drains tokens

Resource Reservation Protocol (RSVP)

Sender 1
Sender 2
PATH

R
PATH

R
R

RESV
(merged)

R
R

R
RESV

R
RESV

Receiver A
Receiver B
RSVP Issues

- RSVP is receiver-driven to be able to support multicast applications
- Only reserve resources at a router if there are sufficient resources along the entire path
  - both for average bandwidth and maximum bursts
- What if there are link failures and the route changes?
  - receivers periodically refresh by sending new requests toward sender
- What if there are sender/receiver failures?
  - reservations are periodically timed out

IETF Integrated Services

- Fine-grained (per flow) guarantees
  - Guaranteed service (bandwidth and bounded delay)
  - Controlled load (bandwidth but variable delay)
- RSVP used to reserve resources at routers
  - Receiver-based signaling that handles failures
  - Router can police that flow obeys reservation
- Priorities, WFQ used to implement guarantees
  - Router classifies packets into a flow as they arrive
  - Packets are scheduled using the flow’s resources
  - Flows with guaranteed service scheduled before controlled load, scheduled before best effort
IETF Differentiated Services

- A coarse-grained approach to QOS
  - Packets are marked as belonging to a small set of services, e.g., premium or best-effort, using the TOS bits in the IP header
- Marking policed at administrative boundaries
  - ISP marks 10Mbps (say) of your traffic as premium depending on your service level agreement (SLAs)
- Routers understand only the different service classes, not individual reservations
  - Use priority queues or WFQ for each class, not for each flow

Two-Tiered Architecture

Mark at Edge routers (per flow state, complex)

Core routers stay simple (no per-flow state, few classes)