Quality of Service (QoS)

- The Internet’s most contentious subject
  - Inside vs. Outside the Network (see P&D, pp. 519-520)

- The Internet’s most embarrassing failure
  - Many papers and proposals, but almost nothing accomplished
Today’s Lecture

- What “could be”, not what is
  - Today’s Internet does not have, nor soon will have, a reasonable QoS solution

- Focus on what one could accomplish with simple (and not-so-simple) mechanisms
  - Understand the basic concepts

- Current deployed mechanisms not discussed
  - Ugly hodge-podge of hacks

What’s the Problem?

- Internet gives all flows the same “best effort” service
  - No promises about when or whether packets will be delivered

- Not all traffic is created equal
  - Different “owners”, different application requirements
  - Some applications require service “assurances”

- How can we give traffic different “quality of service”? 
  - Thus begins the problem of QoS
Three Basic Problems

- Control how a link is shared:
  - Link sharing (discussed by Ion last lecture)

- Give some flows “assured” service
  - Integrated service (and perhaps differentiated service)

- Give some traffic preferential service
  - Differentiated service

A Different Taxonomy

- Giving better service can differ along three dimensions:
  - Relative versus absolute
  - Dropping versus delay
  - Flows versus aggregates

- Each choice requires different mechanisms
  - Intra-Router: Scheduling and dropping decisions
  - Inter-Router: Signaling protocols
Three Basic Questions

- How does a router service this packet?
  - Scheduling (various forms of priority and FQ)
  - Dropping (fancy versions of RED)

- How does the router know what to do with this packet?
  - Bits in packet header or explicit signaling

- How do we control the level of traffic?
  - Service level agreements (SLAs) or admission control

Back to Three Basic Problems

- Link sharing (last lecture): not covered today

- Integrated Services

- Differentiated Services

- Nice web site describing DiffServ and IntServ in succinct language: http://www.rhyshaden.com/qos.htm
Integrated Services

- Attempt to integrate service for “real-time” applications into the Internet
- Known as IntServ
- Total, massive, and humiliating failure
  - 1000s of papers
  - IETF standards
  - and STILL no deployment ...

Key Properties

- All assurances on a per-flow basis
- Traffic can be turned away

Note:
- Co-exists with best-effort service
- Similar mechanisms proposed for ATM (Asynchronous Transfer Mode) but
  - QoS central in ATM, best-effort an afterthought
  - Best-effort central in Internet, QoS an afterthought
Example: Video

Simplify by assuming that
Camera sends at a fixed rate

Circuit-Switched Networks

- Each packet experiences exactly the same delay
- Packet data is displayed as soon as it arrives
- Signal at receiving end is faithful representation
Internet

- Individual packets experience different delays
- Can’t treat network as a “wire”
- Application must adapt to network service

Router Effect on Delay

Delay variation or Jitter

Prob

Min e.g. 30ms 99%

Delay/latency
Router Effects on Traffic

Network Effects on Traffic
Network Effects on Traffic

Cumulative Bits

Source

Router 1

bits in the network

delay

Router n

Time

Cumulative Bits

Source

bits in the network

delay

Router n

Time
Network Effect on Delay

![Network Effect on Delay Diagram]

Choices

- Play back data upon arrival
  - Distorted signal

- Buffer data for a while (playback buffer)
  - Extra delay, less distortion

- Tradeoff depends on application (and use)
  - Noninteractive: absorb delay, eliminate all distortion
  - Interactive: absorb only a little delay, eliminate some distortion
Playback Buffer

Play back data a fixed time interval after it was sent

Playback Point

Nick McKeown
Adaptation

- Can move playback point as delays vary

- Moving playback point:
  - Increases distortion
  - But allows lower delays

Application Taxonomy
(Oversimplified and Fanciful)

- Elastic versus “real-time”
  - Traditional data apps are elastic
  - Streaming media are real-time

- RT intolerant versus RT tolerant
  - Intolerant applications need all data

- Tolerant nonadaptive versus tolerant adaptive
  - Not clear why any tolerant app couldn’t adapt

- Rate-adaptive versus delay-adaptive (or both)
Key Points

- Some apps don’t need to know maximal delay, just need it to be controlled
  - Tolerant, delay-adaptive applications will move playback point to reduce delay
  - Can absorb occasional outliers

- Some apps need to know maximal delay
  - Can’t tolerate loss or distortion
  - Need to fix playback point and so need a priori knowledge of delay bound
  - Bound is typically much worse than actual delays

Two Service Classes

- Controlled Load
  - Keep delays under control, but no bound

- Guaranteed Service
  - Explicit delay bound
**Process**

- Flow requests service from network
  - Service request specification (RSpec)
    - Controlled load: nothing
    - Guaranteed: service rate (can calculate delay)
  - Traffic specification (TSpec) (next slide)

- Routers decide if they can support request
  - Admission control

- If so, traffic is classified and scheduled at routers based on per-flow information

---

**Problem**

- How do you describe bursty traffic?

- Network needs some description of traffic

- But video source is bursty (due to coding)
  - Can’t predict in advance the exact behavior

- Describe “envelope” of traffic: rate and burstiness

- Bits sent between times s and t: $A(s,t) \leq \sigma + \rho(t-s)$
  - $\rho$: average rate
  - $\sigma$: burstiness
TSpec: The Token Bucket

\[ \rho : \text{average rate} \]
\[ \sigma : \text{burstiness} \]

\[ \text{Bits sent between times } s \text{ and } t: \ A(s,t) \leq \sigma + \rho(t-s) \]

Required Elements

- Reservation Protocol
  - How service request gets from host to network

- Admission control algorithm
  - How network decides if it can accept flow

- Packet scheduling algorithms (covered last lecture)
  - So routers can deliver service
Control Plane vs. Data Plane

- Control plane:
  - How information gets to routers

- Data plane:
  - What routers do with that information to data packets

Control Plane: Resource Reservation
Control Plane: Resource Reservation

Sender sends Tspec

Path established

Receiver
Control Plane: Resource Reservation

The receiver signals reservation request.

Control Plane: Admission Control

Per-flow state
Control Plane: Admission Control

Per-flow state on all routers in path

Data Plane

Per-flow classification on each router
Data Plane

Per-flow classification on each router

Data Plane

Per-flow scheduling on each router
Resource Reservation Protocol: RSVP

- Establishes end-to-end reservations over a datagram network
- Designed for multicast (which will be covered later)

- Sources: send TSpec
- Receivers: respond with RSpec Network
- Network: responds to reservation requests

PATH and RESV Messages

- Sender sends PATH messages
  - TSPEC: use token bucket
  - Set up the path state on each router including the address of previous hop (route pinning)
  - Collect path information (for guaranteed service)

- Receiver sends RESV message on the reverse path
  - Specify RSpec and TSpec
  - Sets up the reservation state at each router
Soft State

- Per session state has a timer associated with it
  - Path state, reservation state

- State deleted when timer expires

- Sender/Receiver periodically refreshes the state, resends PATH/RESV messages, resets timer

- Advantages:
  - No need to clean up dangling state after failure
  - Can tolerate lost signaling packets
  - Easy to adapt to route changes

Route Pinning

- Problem: asymmetric routes
  - You may reserve resources on $R \rightarrow S_3 \rightarrow S_5 \rightarrow S_4 \rightarrow S_1 \rightarrow S$, but data travels on $S \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow R$!

- Solution: use PATH to remember direct path from $S$ to $R$, i.e., perform route pinning
Admission Control

- Parameter-based: worst cast analysis
  - Guaranteed service
  - Low utilization

- Measurement-based: measure current traffic
  - Controlled load service
  - Higher utilization

- Remember that best-effort service co-exists
  - No need for IntServ traffic to achieve high utilization

IntServ Node Architecture