Internet Protocols
Fall 2005

Lectures 17-18
QoS
Andreas Terzis
Outline

• QoS
  – Fair Queuing
  – Intserv
  – Diffserv
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

- Want to control how a link is shared:
  - Link sharing
- Want to give some traffic better service
  - Differentiated service
- Want to give some flows “assured” service
  - Integrated service (and perhaps differentiated service)
Link Sharing

• First approach suggested by Nagle (1987)
• Round-robin among different flows
  – one queue per flow
Round-Robin Discussion

- Advantages: protection among flows
  - Misbehaving flows will not affect the performance of well-behaving flows
    - Misbehaving flow – a flow that does not implement any congestion control
    - FIFO does not have such a property
- Disadvantages:
  - More complex than FIFO: per flow queue/state
  - Biased toward large packets – a flow receives service proportional to the number of packets
Solution?

- Bit-by-bit round robin
- Can you do this in practice?
- No, packets cannot be preempted (why?)
- ...we can only approximate it
Fair Queueing (FQ)

- Define a **fluid flow** system: a system in which flows are served bit-by-bit
- Then serve packets in the increasing order of their deadlines
- Advantages
  - Each flow will receive exactly its fair rate
- Note:
  - FQ achieves max-min fairness
Max-Min Fairness

- Denote
  - $C$ – link capacity
  - $N$ – number of flows
  - $r_i$ – arrival rate
- Max-min fair rate computation:
  1. compute $C/N$
  2. if there are flows $i$ such that $r_i \leq C/N$, update $C$ and $N$
     \[ C = C - \sum_{i \text{ s.t } r_i \leq C/N} r_i \]
  3. if no, $f = C/N$, terminate
  4. go to 1
- A flow can receive at most the fair rate, i.e., $\min(f, r_i)$
Example

- $C = 10; \ r_1 = 8, \ r_2 = 6, \ r_3 = 2; \ N = 3$
- $\frac{C}{3} = 3.33 \Rightarrow C = C - r_3 = 8; \ N = 2$
- $\frac{C}{2} = 4; \ f = 4$

$$
\begin{array}{c}
\text{\textbf{f} = 4:} \\
\text{min}(8, 4) = 4 \\
\text{min}(6, 4) = 4 \\
\text{min}(2, 4) = 2
\end{array}
$$
Implementing Fair Queueing

• Idea: serve packets in the order in which they would have finished transmission in the fluid flow system
Example

Flow 1 (arrival traffic)

Flow 2 (arrival traffic)

Service in fluid flow system

Packet system

→ time

→ time

→ time

→ time
FQ Advantages

- FQ protect well-behaved flows from ill-behaved flows
- Example: 1 UDP (10 Mbps) and 31 TCP’s sharing a 10 Mbps link
Big Picture

- FQ does not eliminate congestion → it just manages the congestion
- You need both end-host congestion control and router support for congestion control
  - end-host congestion control to adapt
  - router congestion control to protect/isolate
Three Basic QoS Questions

• How does a router service this packet?
  – scheduling (various forms of priority and RR)
  – dropping (fancy versions of RED)
• How did the router know what to do with this packet?
  – bits in packet header or explicit signaling
• How can one control the level of traffic?
  – service level agreements (SLAs) or admission control
Integrated Services

• An attempt to integrate service for “real-time” applications into the Internet

• Known as IntServ

• A total, massive, and humiliating failure
  – 1000s of papers
  – IETF standards
  – and no deployment....
Key Differences

- All assurances on per-flow basis
- Traffic can be turned away
- Note:
  - all this co-exists with best-effort service
  - similar mechanisms proposed for ATM 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

Nick Mckeown
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 “wire”
- Application must adapt to network service
Router Effect on Delay

Prob

Delay/latency

Min

e.g. 30ms

Delay variation
or
Jitter

99%
Router Effects on Traffic

Cumulative Bits

Source

Router 1

bits in the network

delay

Time
Network Effects on Traffic

Delay's do not build up independently in each router.
Network Effects on Traffic

![Diagram showing cumulative bits over time for different routers, with note that the Service function at router 1 is the arrival function at router 2.](attachment:image.png)
Network Effects on Traffic

Cumulative Bits

Source

delay

bits in the network

Router n

Time

CS 349/Fall05
Network Effect on Delay

Prob

Min

e.g. 200ms

99%

Delay variation or Jitter

Delay/latency

Nick Mckeown

CS 349/Fall05
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

Cumulative Bits

Source

Destination

Playout buffer

Playout delay

Playback Delay

Time
Playback Point
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: $A(s,t) \leq \sigma + \rho(t-s)$
TSpec: The Token Bucket

- \( \rho \): average rate
- \( \sigma \): burstiness

Bits sent between times \( s \) and \( t \): \( A(s,t) \leq \sigma + \rho(t-s) \)

Nick Mckeown

CS 349/Fall05
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 (next lecture)
  – so routers can deliver service
Control Plane versus Data Plane

• Plane as in geometry, not airplane

• 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

CS 349/Fall05 40
Control Plane: Resource Reservation

Path established

CS 349/Fall05
Control Plane: Resource Reservation

The receiver signals reservation request
Control Plane: Admission Control
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

Sender

Receiver

CS 349/Fall05
Resource Reservation Protocol: RSVP

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

- 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
The Big Picture

Network

Sender
Receiver

PATH Msg

CS 349/Fall05
The Big Picture
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
Advantages of IntServ

- Precise QoS delivered at flow granularities
  - good service, given exactly to who needs it

- Decisions made by hosts
  - who know what they need
  - not by organizations, egress/ingress points, etc.

- Fits multicast and unicast traffic equally well
Disadvantages of IntServ

- **Scalability**: per-flow state, classification, etc.
  - we goofed, bigtime
  - aggregation/encapsulation techniques can help
  - can overprovision big links, per-flow ok on small links
  - scalability can be fixed, but no second chance

- **Economic arrangements**:
  - need sophisticated settlements between ISPs
  - right now, settlements are primitive (barter)

- **User charging mechanisms**: need QoS pricing
Differentiated Services

- Some traffic should get better treatment
  - application requirements: interactive vs bulk transfer
  - economic arrangements: first-class versus coach
- What kind of better service could you give?
  - measured by drops, or delay (and drops)
- How do you know which packets to give better service to?
  - bits in packet header
Traffic Limitations

- Can’t give all traffic better service!
- Must limit the amount of traffic that gets better service
- Service Level Agreements (SLA)
  - source agrees to limit amount of traffic in given class
  - network agrees to give that traffic “better” service
    - for a price!
  - economics play an important (fatal?) role in QoS
DiffServ “Code Points”

- Use six of the ToS bits in IP packet header
- Define various “code points”
- Each code point defines a desired per-hop behavior
  - a description of the service the packet should get
  - not a description of the router implementation of that service
“Expedited Forwarding”

- Give packet minimal delay and loss service
  - e.g., put EF packets in high priority queue

- To make this a true “absolute” service,
  - all SLAs must sum to less than the link speed
  - unlikely

- More likely, a way to assure relatively low delay
Is Delay the Problem?

- With RED, most queues are small
- Packets are dropped when queue starts to grow
- Thus, delays are mostly speed-of-light latency
- Service quality is mostly expressed by drop-rate
- Want to give traffic different levels of dropping
“Assured Forwarding”

- Packets are all serviced in order
  - makes TCP implementations perform well
- But some packets can be marked as low-drop and others as high-drop
  - think of it as priority levels for dropping
- Can be implemented using variations of RED
  - different drop probabilities for different classes
Example

- 10% premium traffic, 90% ordinary traffic
- Overall drop rate is 5%
- Can give premium traffic 0% drops, and ordinary traffic a 5.55% drop rate
- Can get a large improvement in service for the small class of traffic without imposing much of a penalty on the other traffic
  - count on SLAs to control premium traffic
Advantages of DiffServ

• Very simple to implement
• Can be applied to different granularities
  – flows
  – institutions
  – traffic types
• Marking can be done at edges or by hosts
• Allows easy peering (bilateral SLAs)
DiffServ Peering

- Ingress routers
  - Police/shape traffic
  - Set Differentiated Service Code Point (DSCP) in DiffServ (DS) field
- Core routers
  - Implement Per Hop Behavior (PHB) for each DSCP
  - Process packets based on DSCP
Disadvantages of DiffServ

• Service is still “best effort”, just a better class of best effort
  – except for EF, which has terrible efficiency
  – all traffic accepted (within SLAs)
• Some applications need better than this
  – certainly some apps need better service than today’s Internet delivers
  – but perhaps if DiffServ were widely deployed premium traffic would get great service (recall example)
  – nonetheless, let’s plunge ahead....
What You Need to Know

• Three kinds of QoS approaches
  – Link sharing, DiffServ, IntServ

• Some basic concepts:
  – differentiated dropping versus service priority
  – per-flow QoS (IntServ) versus per-aggregate QoS (DiffServ)
  – Admission control: parameter versus measurement
  – control plane versus data plane
  – controlled load versus guaranteed service
  – codepoints versus explicit signaling

• Various mechanisms:
  – playback points
  – token bucket
  – RSVP PATH/RESV messages
Factors Limiting QoS Deployment

- Prevalence of overprovisioning
  - if all links are only at 40% utilization, why do you need QoS?
  - lore says that inter-ISP links are not overprovisioned
- Primitive inter-ISP financial arrangements
  - QoS requires financial incentives to enforce tradeoffs
  - Current peering arrangements are not able to carry these incentives through in a meaningful way
    - must agree on pricing and service
    - currently agree on neither!
- End-users not used to pricing/performance options
QoS Debates

• Is overprovisioning enough?
  – if so, is this only because access links are slow?
  – what about Korea, Japan, and other countries with fast access links?
  – Disconnect: ISPs overprovision, users get bad service

• Is differentiated services enough?
  – can one really deliver reliable service just using relative priorities?
  – is EF service a viable option?

• It all depends on adaptability of applications