EE 122: End-to-end QoS

Ion Stoica (and Brighten Godfrey)
TAs: Lucian Popa, David Zats and Ganesh Ananthanarayanan
http://inst.eecs.berkeley.edu/~ee122/
(Materials with thanks to Vern Paxson, Jennifer Rexford, and colleagues at UC Berkeley)

Goals of Today’s Lecture

- Traffic characterization
  - Token bucket

- QoS models:
  - Integrated Services: End-to-end network “reservations”
  - Differentiated Services: First Class vs. Coach
Reserving Resources End-to-End

- Source sends a reservation message
  - E.g., “this flow needs 5 Mbps”
- Each router along the path
  - Keeps track of the reserved resources
    - E.g., “the link has 6 Mbps left”
  - Checks if enough resources remain
    - E.g., “6 Mbps > 5 Mbps, so circuit can be accepted”
  - Creates state for flow and reserves resources
    - E.g., “now only 1 Mbps is available”

How to Specify Bursty Traffic

- Option #1: Specify the maximum bit rate. Problems?
  - Maximum bit rate may be much higher average
  - Reserving for the worst case is wasteful
- Option #2: Specify the average bit rate. Problems?
  - Average bit rate is not sufficient
  - Network will not be able to carry all of the packets
  - Reserving for average case leads to bad performance
- Option #3: Specify the burstiness of the traffic
  - Specify both the average rate and the burst size
  - Allows the sender to transmit bursty traffic
  - … and the network to reserve the necessary resources
Characterizing Burstiness: Token Bucket

- **Parameters**
  - \( r \) – average rate, i.e., rate at which tokens fill the bucket
  - \( b \) – bucket depth (limits size of burst)
  - \( R \) – maximum link capacity or peak rate

- A bit can be transmitted only when a token is available

\[
\text{Maximum # of bits sent} = \frac{b \cdot R}{(R-r)}
\]

**Traffic Enforcement: Example**

- \( r = 100 \text{ Kbps}; b = 3 \text{ Kb}; R = 500 \text{ Kbps} \)

  (a) 3Kb
  
  \( T = 0 \) : 1Kb packet arrives

  (b) 2.2Kb
  
  \( T = 2 \text{ms} \) : packet transmitted
  
  \( b = 3\text{Kb} - 1\text{Kb} + 2\text{ms} \times 100\text{Kbps} = 2.2\text{Kb} \)

  (c) 2.4Kb
  
  \( T = 4\text{ms} \) : 3Kb packet arrives

  (d) 3Kb
  
  \( T = 10\text{ms} \) : packet needs to wait until enough tokens are in the bucket

  (e) 0.6Kb
  
  \( T = 16\text{ms} \) : packet transmitted
Source Traffic Characterization: Arrival Curve

- Arrival curve – maximum amount of bits transmitted during any interval of time $\Delta t$
- Use token bucket to bound arrival curve

Arrival Curve: Example

- Arrival curve – maximum amount of bits transmitted during any interval of time $\Delta t$
- Use token bucket to bound arrival curve
QoS Guarantees: Per-hop Reservation

- End-host: specify
  - arrival rate characterized by token bucket with parameters \((b, r, R)\)
  - the maximum tolerable delay \(D\), no losses
- Router: allocate bandwidth \(r_a\), buffer space \(B_a\) such that
  - no packet is dropped
  - no packet experiences a delay larger than \(D\)

![Diagram showing QoS guarantees](image)

Ensuring the Source Behaves

- Guarantees depend on the source behaving
  - Extra traffic might overload one or more links
  - Leading to congestion, and resulting delay and loss
  - Solution: need to enforce the traffic specification
- Solution #1: policing
  - Drop all data in excess of the traffic specification
- Solution #2: shaping
  - Delay the data until it obeys the traffic specification
- Solution #3: marking
  - Mark all data in excess of the traffic specification
  - … and give these packets lower priority in the network
**Integrated Services: 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
  - How routers deliver service

- Architecture for solution: *IntServ*
  - Provides service guarantees at a *per-flow granularity*

**Control Plane vs. Data Plane**

- Control plane:
  - How information gets to routers
- Data plane:
  - What routers do with that information when processing data packets
Control Plane: Resource Reservation

Sender sends specification of traffic profile
Control Plane: Resource Reservation

Path established (or perhaps admission control denies path)

The receiver accepts reservation request
Control Plane: Admission Control

Sender

Per-flow state (soft state)

Receiver

Sender

Per-flow state on all routers in path

Receiver
Data Plane

Sender

Per-flow classification on each router

Receiver

Data Plane

Sender

Per-flow classification on each router

Receiver
Data Plane

Admission Control

- Parameter-based: worst case analysis
  - Guaranteed service
  - Lower 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
5 Minute Break

Questions Before We Proceed?

Problems with IntServ

- Scalability: per-flow state & classification
  - 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
  - Contemporary settlements are primitive
    - Unidirectional, or barter
- User charging mechanisms: need QoS pricing
  - On a fine-grained basis
**Differentiated Services (DiffServ)**

- Give some traffic better treatment than other
  - Application requirements: interactive vs. bulk transfer
  - Economic arrangements: first-class versus coach
- What kind of better service could you give?
  - Fewer drops
  - Lower delay
  - Lower delay variation (jitter)
- How to know which packets get better service?
  - Bits in packet header
- Deals with traffic in aggregate
  - Provides weaker services
  - But much more scalable

**Diffserv Architecture**

- Ingress routers - entrance to a DiffServ domain
  - Police or shape traffic
  - Set Differentiated Service Code Point (DSCP) in IP header
- Core routers
  - Implement Per Hop Behavior (PHB) for each DSCP
  - Process packets based on DSCP
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

“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
Is Delay the Problem?

- 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
Example

- 10% premium traffic, 90% ordinary traffic
- Overall drop rate is 5%
- Give premium traffic 0% drops; ordinary traffic a 5.55% drop rate
- 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

DiffServ “Code Points”

- Use six of the ToS bits in IP packet header
- Define various “code points”
  - Alternative classes of service (drop probability and assured forwarding)
- Each code point defines a desired per-hop behavior
  - Description of the service the packet should get
  - Not a description of the router implementation of that service
Differentiated Service (DS) Field

- DS field encodes Per-Hop Behavior (PHB)
  - E.g., Expedited Forwarding (all packets receive minimal delay & loss)
  - E.g., Assured Forwarding (packets marked with low/high drop probabilities)

Edge Router

Data traffic

Per aggregate Classification (e.g., user)
Assumptions

- Assume two bits
  - P-bit denotes premium traffic
  - A-bit denotes assured traffic

- Traffic conditioner (TC) implement
  - Metering
  - Marking
  - Shaping

TC Performing Metering/Marking

- Used to implement Assured Service
- In-profile traffic is marked:
  - A-bit is set in every packet
- Out-of-profile (excess) traffic is unmarked
  - A-bit is cleared (if it was previously set) in every packet; this traffic treated as best-effort
**TC Performing Metering/Marking/Shaping**

- Used to implement Premium Service
- In-profile traffic marked:
  - Set P-bit in each packet
- Out-of-profile traffic is delayed, and when buffer overflows it is dropped

**Scheduler**

- Premium traffic sent at high priority
- Assured and best-effort traffic sent at low priority
  - Always drop OUT (best-effort) packets first
Control Path

- Each domain is assigned a Bandwidth Broker (BB)
  - Usually, used to perform ingress-egress bandwidth allocation
- BB is responsible to perform admission control in the entire domain
- BB not easy to implement
  - Require complete knowledge about domain
  - Single point of failure, may be performance bottleneck
  - Designing BB still a research problem

Example

- Achieve end-to-end bandwidth guarantee
## Comparison to Best-Effort & Intserv

<table>
<thead>
<tr>
<th></th>
<th>Best-Effort</th>
<th>Diffserv</th>
<th>Intserv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>Connectivity</td>
<td>Per aggregate isolation</td>
<td>Per flow isolation</td>
</tr>
<tr>
<td></td>
<td>No isolation</td>
<td>Per aggregate guarantee</td>
<td>Per flow guarantee</td>
</tr>
<tr>
<td></td>
<td>No guarantees</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service scope</td>
<td>End-to-end</td>
<td>Domain</td>
<td>End-to-end</td>
</tr>
<tr>
<td>Complexity</td>
<td>No setup</td>
<td>Long term setup</td>
<td>Per flow setup</td>
</tr>
<tr>
<td>Scalability</td>
<td>Highly scalable (nodes maintain only routing state)</td>
<td>Scalable (edge routers maintain per aggregate state; core routers per class state)</td>
<td>Not scalable (each router maintains per flow state)</td>
</tr>
</tbody>
</table>

## Discussion: Limited QoS Deployment

- End-to-end QoS across multiple providers/domains is **not** available today
- Issue #1: complexity of payment
  - Requires payment system among multiple parties
    - And agreement on what constitutes service
  - Diffserv tries to structure this as series of **bilateral** agreements ...
    - ... but lessens likelihood of end-to-end service
  - Architecture includes notion of “Bandwidth Broker” for end-to-end provisioning
    - Solid design has proved elusive
- Need infrastructure for metering/billing end user
Limited QoS Deployment, con’t

- Issue #2: prevalence of overprovisioning
  - Within a large ISP, links tend to have plenty of headroom
  - Inter-ISP links are \textit{not} over provisioned, however
- 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
- Key difference: intra-ISP vs. general end-to-end

\textbf{Exploiting Lack of End-to-End QoS}

- Suppose an ISP offers their own Internet service
  - E.g., portal (ala’ Yahoo) or search engine (ala’ Google)
- Then it’s \textit{in their interest} that performance to Yahoo or Google is \textit{inferior}
  - So customers prefer to use their value-added services
- ISP can
  - recognize traffic to competitor and demote it
  - charge competitor if they want well-provision paths
  - just not put effort/$ into high-capacity interconnects w/ other ISPs; congestion provides traffic demotion directly
- Works particularly well for \textbf{large} providers w/ lots of valuable content
Summary

- QoS models
  - Reservations & admission control
  - Descriptions of bursty traffic: token buckets
- IntServ provides per-flow performance guarantees
  - But lacks scalability
- DiffServ provides per-aggregate tiers of relative perf.
  - Scalable, but not as powerful
- Neither is generally available end-to-end today
- ISPs may manipulate what services receive what performance raises issues of: network neutrality