Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.892 Fall 1999 **Quality of Service in the Internet** November 15, 1999

Overview. QoS architectures & mechanisms. Integrated services, RSVP, differentiated services.

1 QoS architectures & mechanisms

- Problem: how to do better than current best-effort network in terms of delay, jitter, throughput, and error rate.
- Why do we care?
  1. To accomodate some interesting applications.
  2. Economic incentive for further Internet development.
  3. To prevent a "tragedy of the commons."
- Define the following:
  1. Architecture. The framework for the Internet that will scalably extend th best-effort architecture and service model.
  2. Mechanism. What techniques are implemented in the network to provide this.
  3. Policy. Under what condition is each mechanism used.
  4. Behavior. What is the observable consequence of a mechanism.
- Why is Internet QoS hard?
  - At odds with current architecture: dumb-network and smart-ends.
  - Scalability.
  - Deployment.
  - Somewhat at odds with reasons for Internet's efficiency, statistical multiplexing.
  - Heterogeneity of the Internet (many orders of magnitude of performance variation).
  - Handling multicast.
- Architectural choices:
  - Reservation-based.
  - Priority-based.

  These impact the service model and the way in which applications signify what they want.

2 Integrated services

- Extends the Internet architecture in a compatible way. Provides QoS facilities, controlled link sharing, and handles multicast.
- Extended service model defines externally visible *behavior* and provides service classes.
- A reference implementation defines a reservation setup protocol like RSVP.
- Integrated services defines three types of service:
  1. Guaranteed service.
  2. Controlled load service.
  3. Best-effort.

2.1 Guaranteed service

- RFC 2212. For real-time applications with tight delay requirements, inelastic applications.
- Assured level of bandwidth, mathematical bound on end-to-end delay, no queueing losses for conforming packets.
- To use guaranteed service, application specifies its traffic spec (TSpec) and a reservation spec (RSpec).
- Incoming traffic must be subject to traffic policing or traffic shaping.
  1. Policing: Treat non-comforming traffic as best-effort.
  2. Shaping: bursty traffic is shaped according to TSpec.

  A *token bucket* can be used to do both.
- What is TSpec?
  1. Max burst size, b.
  2. Average rate, r.
  3. Max size, M.
  4. Min size, m.
  5. Peak rate, P

  Traffic not conforming to (σ, ρ) model subject to a token bucket, which produces in any time interval T, a number of bits, $A(T) = \min(p \times T, b + r \times T)$.

- What is RSpec?
  1. Reservation rate, R. R must be $geqr$, the TSpec rate.
  2. Slack delay, S. S is the difference between desired delay and end-to-end delay obtained at reservation R.

2.2 Controlled-load service

- RFC 2211. For adaptive, real-time traffic. Service model is that of a best-effort traffic in an unloaded network.
- Specification: Only TSpec. No RSpec. Want to *share* jitter and post-facto delay, not *isolate* delays. This is motivated by wanting to reduce playback buffer sizes. (CSZ92 paper.)
- Non-conforming traffic treated as best-effort.

2.3 IntServ components

- Packet classifier. What class does each packet belong to?
- Packet scheduler. May police/shape too.
- Reservation setup protocol. Protocol to reserve resources for a specified QoS (RSpec). Need to know TSpec too to shape/police.
- Admission control. Decide whether to allow a flow or not.
- Queueing in IntServ.
  1. WFQ or equivalent for guaranteed class. Parekh-Gallager delay bound can then be provided.
  2. Want sharing for controlled-load class (CSZ calls this predicted service; think of controlled load as a more formal version of predicted service with a TSpec).
  3. CSZ92 presents a FIFO+ scheduling scheme for predicted service.
  4. RFC doesn't actually say anything about mechanism, only about behavior.

3 Reservation setup using RSVP

- RFC 2205, standards track.
- Requirements:
  1. Unicast and multicast.
  2. Heterogeneous receivers in multicast.
  3. Dynamic membership changes.
  4. Efficiency. (Of course!)
  5. Connectionless model.
- Principles.
  1. Receiver-initiated. Reservation is requested, maintained, and torn down by receivers.
  2. Reservation state is *soft-state.*
  3. Two-pass. Forward PATH message and reverse RESV message.
  4. Application-independent (opaque information transport)

5. Independent of routing protocol. (but route pinning?)

- RSVP defines a *session* and a *flow descriptor.*
- Session: destination of data flow.
- Flow descriptor has flow spec and filter spec. Flow spec is (TSpec, RSpec). Filter spec defines classifier at router.
- First step: PATH message sent downstream with TSpec. Router creates or updates state containing previous hop, sender TSpec. Restarts cleanup timer associated with soft state. Forwards PATH on.
- Second step: Receiver(s) sends RESV with RSpec. Analyze for admission control. Forward to previous hop.

4 Differentiated services

- Biggest difference from IntServ: doesn't pre-define end-to-end service model. Defines only *per-hop behaviors (PHB).*
- Scalable architecture: complexity at edges, interior is simple.
- Picture at edge:

```
                          +--------+
                          |        |-------------------+
                   +----->| Meter  |                   |
                   |      |        |--+                |
                   |      +--------+  |                |
                   |                  V                V
             +------------+      +---------+      +----------+
             |            |      |         |      | Shaper/  |
packets ====>| Classifier |=====>| Marker  |=====>| Dropper  |=====>
             |            |      |         |      |          |
             +------------+      +---------+      +----------+
```

Mark, drop, meter, shape constitute *traffic conditioning.*

- DS codepoint: value of DS field.
- Interior: PHB based only on DS codepoint. At edge, classification plus conditioning.