EECS 122, Lecture 28

Today's Topic:

Internet QoS:

IntServ and DiffServ

Kevin Fall, kfall@cs.berkeley.edu

QoS/CoS in the Internet

- Provide differential traffic handling within the Internet (all at layer 3)
- Started as IntServ, focusing on per-flow QoS support, but has evolved to both IntServ and DiffServ (differential services)
- IntServ continuing focus on per-flow support; Diffserv focusing on class-based QoS (CoS) for aggregates of flows

Internet Integrated Services (IntServ)

- Provide guaranteed and “very good” service per flow at layer 3
- Major components
  - QoS requirement specifications
  - resource-sharing requirements
  - allowance for packet dropping
  - provision for usage feedback
  - signaling/provisioning

IntServ Architecture

- Guaranteed and controlled-load service
  - for delay-intolerant and tolerant apps
- Transmitter Token Bucket (TSPEC)
  - per-sender token bucket parameters
  - token rate (r), depth (b), and peak rate (p)
  - minimum policed unit (m)
    - smaller packets rounded up to this size
  - maximum packet size (M)

Controlled Load Service (2211)

- Approximate behavior of unloaded network:
  - using admission control (either analytic or measurement based), ensure the conformant traffic on CL flows receive good performance
  - conformant flows are those that do not exceed the TSPEC
  - nonconformant packets are handled best-effort (not explicitly dropped)

Behavior from CL Service

- Applications may assume
  - very high percentage of packets will be successfully delivered
  - very high percentage of the delay of packets will be near the minimum
- Restrictions
  - fragmented packets are handled best-effort
  - no quantitative bounds on delay or jitter

Time Scales for CL Service

- Assumptions of low queuing delay and little or no loss are given over long time scales
  - that is, over time scales larger than a “burst time” (time required for a flow’s max size data burst to be transmitted at the flow’s requested transmit rate)
  - over small time scales, occasional congestion may occur (normal)
  - over large time scales, problems are indicative of a failure of the CL service

Implementation Requirements

- CL service specification is very loose on implementation requirements:
  - no specific admission control algorithm
  - no specific buffer management or scheduling algorithm
- What it does say about handling nonconformant traffic:
  - continue CL service to conformant flows
  - nonconformants must not affect others
  - forward nonconformants as best-effort

Perspective

- CL service is targeted at adaptive real-time (loss tolerant) applications
- Minimum requirements makes implementation a less daunting task
- Possible implementations might include:
  - FCFS scheduling, 2 priority levels
  - WFQ or class based queuing (CBQ)
    - WFQ also provides policing function
  - (still need to use some admission control)

Guaranteed Service (2212)

- Provide bounds on bandwidth and maximum (queuing) delay w/no loss
  - does not address minimum or average queuing delay or delay-jitter
  - requires all hops along path to support GS to be effective
  - requires admission control, policing (edges of network), and reshaping (within network)
  - uses a TSPEC at sender, RSPEC at receiver

Receiver Specification (RSPEC)

- RSPEC includes <u>rate</u> R and <u>slack term</u> S
  - slack term is the difference (in microseconds) between the desired delay and the delay obtained using a reservation at rate R; can be used to reduce reservations
  - req’d: R >= r [sender avg rate], S >= 0
  - so, the maximum needed service rate (for a single receiver) at any particular node is min(p,R)
  - for R>p, less build-up in network (less delay)

A Word on Delay

- End-to-end delay comprises fixed delay (propagation, transmission [assuming fixed packet sizes]) and variable delay (queuing)
- Only the queuing delay is addressed by GS service, and it is provided based on the TSPEC and RSPEC (which do not include an explicit delay specification)
- Thus, the offered delay is *derived* from the SPECs and the network

The Fluid Model of Service

- GS (and other results we have seen) are based on the <u>fluid model</u> of service:
  - fluid service at rate R is essentially the service that would be provided by a <u>dedicated path of bandwidth R</u>
  - flow s service is completely <u>independent</u> of any other
- Level of service characterized by rate R and buffer B
  - only <u>bounded variation</u> from fluid model is ok

Deriving the Delay Bound

- Recall the idealized bound on delay for a leaky bucket constrained source (r,b) is:

$$D \le \frac{b}{R}; \qquad \text{if } R \ge r$$

- So, to cover non-idealized cases, add a rate-dependent and independent term:

$$D \le \frac{b}{R} + \frac{C}{R} + D;\ \text{if } R \ge r$$

Deriving the Delay Bound

- But these deviations are cumulative, so:

$$D \le \frac{b}{R} + \frac{C_{tot}}{R} + D_{tot};\ \text{if } R \ge r$$

$$C_{tot} = \sum_{i \in \{hops\}} C_i;\ D_{tot} = \sum_{i \in \{hops\}} D_i$$

- This bound is ok, but if we know *p* (source peak rate), we can tighten it a bit further by noting we won t get an instantaneous burst of size *b*…

When we know p…

- In the case R>=p, only a single packet looks like a burst, so:

$$D \le \frac{\boxed{M}}{R} + \frac{C_{tot}}{R} + D_{tot};\ \text{if } R \ge p \ge r$$

- But when R<p, we have a fraction of *b* in addition to the base packet M:

$$\boxed{M + (b-M)\frac{(p-R)}{(p-r)}}; p > R \ge r$$

The GS Maximum Delay Bound

- The GS max delay bound is then:

$$D = \frac{X}{R} + \frac{(M + C_{tot})}{R} + D_{tot}$$

$$X = \begin{cases} (b-M)(p-R)/(p-r) & p > R \ge r \\ 0 & R \ge p \ge r \end{cases}$$

- Ad mentioned, C, D are sums of max per-hop deviations from perfect fluid flow model: $C_{tot} = \sum_{i \in \{hops\}} C_i;\ D_{tot} = \sum_{i \in \{hops\}} D_i$

Policing and Reshaping in GS

- Policing
  - performed at edge of network
  - arriving traffic compared against TSPEC
- Reshaping (restore distorted traffic)
  - at all heterogeneous source branch points
    - in multicast, where an outgoing TSPEC might be less than the incoming one
  - at all source merge points
    - in multicast, where multiple sources on a tree or branch meet

Perspective

- Guaranteed Service makes substantial demands on internal network nodes
  - support for traffic re-shaping
  - bounded one-hop delay of b/R+C/R+D
- GS (as well as CL) require some form of signaling to communicate TSPEC (and RSPEC) info to network elements. No specific way is required by CL or GS, but the Internet standard protocol is RSVP…

ReSerVation Protocol (2205)

- RSVP is the IETF-specified reservation protocol
  - does not specify exact traffic or QoS metrics
  - specifies message formats and how they are exchanged
- In RSVP, receivers are responsible for requesting QoS instead of sender
- Conceived for use with IP multicast
- Establishes soft-state within switches

Reservation Set-UP

- RSVP interacts with routing tables to determine the end-to-end path
- RSVP uses soft state
  - note that routes might change
  - state times out (typically) after 30 seconds
- QoS is dynamic:
  - senders and receivers can modify their TSPECs and RSPECs
  - new (multicast) senders/receivers may arrive

Method of Operation

- Sender starts sending a PATH message
  - travels through network establishing source state in routers; subject to admission and policy control (authorization)
  - receivers respond with RESV (reservation request) messages, which flow back toward the sender
  - only reserves resources in 1 direction (S->R)
- RESV messages contain one of several styles of reservations

Reservation Styles

- Different senders in same RSVP session
  - source-specific (distinct) reservation
  - non-specific (shared) reservation
- Describing multiple senders (filter spec)
  - explicit list (list of sender by address)
  - wildcard (any sender in session)
- Reservation merging not allowed between distinct & shared or between explicit & wildcard reservations

Reservation Components

- RESV messages contain flow descriptors:
  - a FlowSpec and FilterSpec
  - FlowSpec contains QoS parameters
  - FilterSpec, together with a session identifier consisting of an address/port/proto tuple, defines the "flow" to receive the QoS
- FlowSpec generally contains
  - service class (e.g. CL, guaranteed, etc)
  - TSPEC, RSPEC parameters (formats are specific to the service class)

Reservation Merging

- For use with multicast, multiple reservations along different paths may be merged
- Generally, the merged FlowSpec is the “largest” of the merging FlowSpecs
- The merge procedure is service specific (e.g. controlled load and guaranteed service have their own)
- See RFCs 2205, 2210-2 for details

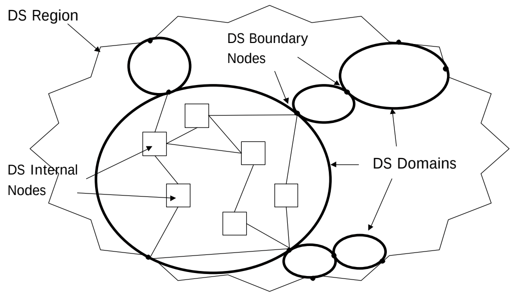
Perspective

- RSVP has been implemented, mostly in experimental applications
- The RSVP (and more generally, IntServ) effort has spawned two new efforts:
  - ISSLL (IntServ over specific link layers)
  - DiffServ
- DiffServ aims at providing support for differential service in the Internet, but is focusing more on traffic aggregates...

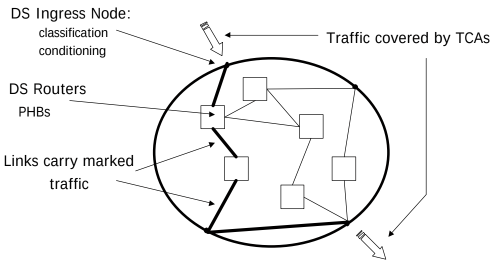
The DiffServ Architecture (2475)

- Goal is to provide scalable service differentiation in the Internet
  - traffic aggregation
  - complexity limited to the network edge devices (helps in building fast routers)
- Basic model
  - traffic entering a network is classified and possibly conditioned at the network boundary and assigned a behavior aggregate
  - forwarding defined per-aggregate (as a PHB)

It’s a DiffServ World After All...



Inside a DS Domain



TCA: traffic conditioning agreement
SLA: service level agreement

Traffic Classification and Conditioning

- Traffic is classified and conditioned on ingress or egress to/from a DS domain
  - classification involves the assignment of some DS codepoint (header value)
  - conditioning may involve shaping, policing, or remarking (changing the DSCP)
- Within a domain, a local source may act as its own ingress point, or the first-hop router near it may do so

Per-Hop Behaviors (PHBs)

- a description of the externally observable forwarding behavior of a DS node applied to a particular DS behavior aggregate
- the means by which nodes allocate resources; building block for diff services
- Simple example:
  - guarantee X % of link bw to aggregate Y
- Implemented via scheduling and buffer management (as we would expect)

Assured Forwarding (AF) PHB

- Four AF Classes
  - within each class, packets receive one of three <u>drop preference</u> values which must result in at least 2 distinct drop probabilities
  - no re-ordering allowed in same microflow (5 tuple of src/dst address and port + proto)
- Requirements on congestion behavior
  - allow short term bursts, but drop packets on persistent overload
  - (gee, sounds like RED, doesn t it?)

Expedited Forwarding (EF) PHB

- requires that the departure rate must equal or exceed some configurable rate
- if forwarding could lead to unlimited preemption (e.g. strict priority), regulation is required
- traffic exceeding regulation is <u>dropped</u>

Implementing the EF PHB

- Could use priority queues, subject to policing restriction
- Variants of WRR also possible
- Particular selection affects microflow delay or jitter, but these are not bounded by EF

Example services using PHBs

- The “Olympic” service (AF PHB):
  - better than best effort (BBE)
    - basically, a bw spec and spec for excess
  - gold, silver, bronze
    - e.g. net control, real-time, non-rt (perhaps)
- The virtual leased line (EF PHB):
  - fixed bw and delay parameters