Lecture 15: Real-Time Delivery

15.0 Main Points

Motivation
Static schedules (ATM)
Delay adaptation (Internet)

15.1 Motivation

Real-time packet delivery -- packets must be delivered by deadline to be useful.

Best effort packet delivery -- packets have no deadline.

Two types of real time applications:

Hard real time: disaster if packets delivered late

Soft real time: packets useless if delivered late, but no disaster

To provide real time delivery, need **guarantees**: bandwidth, delay, error rate

Two parts to real time delivery

Admission control

Must request service in advance. Deny request if can't guarantee

Resource scheduling

How do you schedule resources, given a set of guarantees

How does real time traffic interact with best effort?

15.2 Static schedules (ATM)

Simple approach: each switch has schedule of which packets to send in each time slot, infinitely repeating.

At connection setup:

reserve bandwidth by reserving slots, buffer space at each router in path

delay is bounded by ~ 2 frames per switch
buffering is bounded by ~ 2 frames per switch

Best effort can fit in empty or unused slots.

15.3 Delay Adaptation

Internet

no connection setup (in IP)
no fixed routes (system can reroute at any time)
no buffer reservation
no switch schedules

How can we add real time support to the Internet?

Approach: Reserve bandwidth at routers, but without explicit schedules (e.g., use priority scheduling to favor real time traffic over best effort)

Ex: RSVP -- reservations are receiver driven

- send control packets along reverse path to reserve bandwidth
- refresh periodically . Soft state -- if route changes, old reservations time out; refresh reestablishes. Lose guarantee
- works with multicast (dynamic sharing of bandwidth among heterogeneous receivers)

Without explicit schedule, delay can be variable. Adapt !

Playback point estimation. Explicit timestamps with receiver buffering to absorb jitter.

How big should delay be?

Delay too big?

Delay too small?

Estimate delivery time, similar to TCP RTT estimation:

Ti = sender timestamp for packet i
Ai = arrival time at receiver for packet i

Di = delay for packet i = $|(Ai - Ti) - (Ai- - Ti-)|$

Use exponential weighted moving average to smooth estimate; adjust playback to compensate for changes in playback point.