Multicast Transport

CSE 561 Lecture 14, Spring 2002.
David Wetherall

Overview

- Last time: multicast routing
  - How to get packets from a sender to a set of receivers
- This time: multicast transport
  - What's the equivalent of "TCP" for multicast?
- Case study of two multimedia apps and protocols
  - *vic*, video conferencing; key issue: congestion control
  - *wb*, a shared whiteboard; key issue: reliability

Multicast Congestion Control

- What are the bandwidth needs of multicast applications?
  - E.g., software distribution versus conferencing
  - They must still be matched to the network even if not elastic.

- Key Issue: Heterogeneity
  - Different receiver bandwidths mean no single answer is sufficient
  - So how do we match receivers w/o separate unicasting?

- Key Approach: Layered Coding
  - Send at several rates and let receivers select the best
  - Rates can carry separate or layered information

RLM (McCanne 95)

- How do receivers select “the best” layers?
  - Want to avoid overwhelming the network
- One solution:
  - Imagine if routers implemented priority drop (and FQ) …
  - Source could just send and “best layers” would fall out
  - But routers are best effort drop-tail with one class of service!
- RLM approach:
  - Have receivers learn (by join experiments) what layers suit them
  - Implement using one IP multicast group per layer

Binary Exponential Backoff (BEB)

- Ethernet collisions are the classic example
  - Double interval over which retransmission timer is chosen
  - Reset interval once successful

- The technique is generally useful for adapting to an environment (e.g., network conditions)
  - TCP timeouts
  - RLM
  - SRM
  - Damping flapping links (BGP, AutoNet skeptics)?

Application Level Framing (ALF)

- Clark and Tennenhouse (SIGCOMM'90)
- A design principle that calls for applications to send data in terms of units meaningful to them and lower layers to preserve these boundaries.

- Why?
  - Consider lost/reordered data
  - Consider manipulation inside the network ☺

RLM Discussion

- Scalability
  - Shared join experiments
  - One receiver can learn when a layer will fail, but not succeed.

- Security
  - What are the interactions?

Multicast Reliability

- Scaling problems: why is multicast reliability hard, different from TCP?
  - Straightforward use of ACKs doesn't scale
  - Nor do NACKs due to *implosion*
  - Centralized retransmissions become a scaling bottleneck
  - Receiver orientation if IP multicast semantics
- Approaches to distribute work and hence scale
  - Use all group members for error recovery
  - Randomization (to avoid implosion)
  - FEC / parity coding (one retransmission for different losses)

SRM (Floyd et. al. 96)

- Approach is to *distribute* retransmissions over group
  - Challenge is to minimize repair requests/responses
- Consider different topologies:
  - Chain – use network distance to suppress duplicates
  - Star – use randomization to suppress duplicates
  - Trees – a mixture
- Adaptive learning
  - Tune timer parameters to network conditions

SRM Discussion

- Scalability
  - What are the problems?
  - How well does local recovery work?
  - Do we need network support for local recovery?

- Security
  - Cooperation is an underlying assumption
  - What about incentives?