Lecture 22

CS625: Advanced Computer Networks
Fall 2003

Wednesday, 08 October 2003

Bhaskaran Raman
CSE, IIT-Kanpur

http://www.cse.iitk.ac.in/users/braman/courses/cs625-fall2003/outline.html

Topic for Today

- Scalable Reliable Multicast (SRM) [FJM95]
- *Scribe for today?*
- *Further topics for semester?*

Reliable Multicast

- Example applications:
  - Software distribution
  - PPT slides in a video-conference
  - Distributed white-board (wb)
- For unicast, TCP provides reliability
- "One size does NOT fit all" in multicast:
  - Applications may or may not require ordering
  - One or more senders possible
  - Multiple senders possible for *same* data

Design Goals

- Give flexibility to application
  - Ordering, where data comes from, etc.
- Assume only datagram delivery from IP
- Adaptive parameters to algorithms

Design Challenges

- Unicast has "fate-sharing"
  - Sender or receiver can be responsible for reliability
  - TCP uses sender-based approach
- Sender-based approach has problems in multicast:
  - ACK implosion
  - Sender has to track set of receivers
  - Per-receiver state at sender
  - RTT? CWND? These are undefined
- Hence, receiver-based approach

Design Challenges (continued)

- Sequencing:
  - Per source or independent of source?
  - Sequence number wrap-around can happen
    - Problems with receiver joining late
    - Or intermediate partitions
- Solution: Application Layer Framing (ALF)
  - Unit of data defined by application
  - Example: "block-5 of file slides.ps"
    - FTP/TCP can also use this
  - In multicast, sequence number can now be per-source or independent of source
    - And, anyone who has the data can retransmit

The Reliable Multicast Protocol

- Each receiver keeps track of sequence numbers received
- Periodic session messages sent by each sender
  - With highest sequence number sent so far
  - And timestamp (for RTT estimation)
- On loss detection,
  - Multicast request message after random delay
  - Multicast response message after random delay

Random Timer Values

- Repair timer
  - Uniform in **[C1, C1+C2] X d(S,A)**
- Response timer
  - Uniform in **[D1, D1+D2] X d(B,A)**
- C1 high:
  - Larger repair delay
  - But, more suppression for farther nodes
- C2 high:
  - Larger response delay
  - But, more suppression for larger group sizes

Extensions

- Adaptive values of C1, C2, D1, D2
- Adapt based on
  - Observed delay in recovery
  - Observed duplicate requests
- Local recovery
  - Scoped repair/response messages
- Application specific adaptations
  - Different tolerance to delay and number of duplicates
- Congestion control?

Further topics...

- Overlay networks, and overlay multicast
- Reminder:
  - Assignment-2 assigned this Friday