Reliable Multicast Protocols

Mark Handley
ACIRI
mjh@aciri.org

Overview

- **One Size Doesn't Fit All**
  - ACK vs NACK
  - One-to-many or Many-to-many
  - Congestion Control
  - Router Support
- **Solutions**
  - SRM
  - RMTP-II
  - PGM

One Size Doesn't Fit All

- **Does your application:**
  - Need to know that everyone got the data?
  - Need to constrain differences between receivers?
  - Need to scale to large numbers of receivers?
  - Need to be totally reliable?
  - Need many interacting senders?
  - Need to work in the public Internet?
  - Need to work without a return path (eg satelite)?

- **The answers to these questions constrain the design space.**

ACK vs NACK

- **Two fundamentally different approaches to RM:**
  - ACK: Have everyone acknowledge the data.
  - NACK: Have members inform when data is missing.

- **If you need absolute knowledge that someone is missing the data, you probably need an ACK-based protocol.**

- **NACK-based protocols scale better. Often you don't care if someone didn't get all the data.**

ACK-based mechanisms

- **ACK-implosion is an issue.**
  - Without router support, this limits scaling.
  - Can combine many ACKs into a packet.
  - Scaling also limited by link or member failure
    - still require a way to eject a failed member or everything just hangs.
- **ACK mechanisms are often combined into a tree-based protocol**
  - Allows aggregation of ACK messages.
  - Still need to deal explicitly with failures.

NACK-based mechanisms

- **NACK-implosion is an issue.**
  - NACK-suppression solves this.
  - Timer based solutions (SRM) and sender-triggered solutions (NTE) both work well.

Reliability

- **Several mechanisms:**
  - Repeated Transmission (open loop)
  - Retransmission (NACK or lack of ACK)
  - Packet Level FEC (pro-active)
  - Packet Level FEC (re-active)
  - Application Level Redundancy

- **Often these can be combined.**

One-to-Many Vs Many-to-Many

- **One-to-many:**
  - Stock price feeds
  - Software distribution
- **Many-to-many:**
  - Games
  - Collaboration Tools

One-to-Many Vs Many-to-Many

- **One-to-many applications are much simpler:**
  - One data source.
  - Others may be able to retransmit signed data.

- **Many-to-many applications are complex:**
  - Consistency issues tied to distribution model.
  - Trust model is tied to distribution model.
  - Who sends the data affects its semantics.

Congestion Control

- **This is the biggest problem with multicast.**
  - The IETF will not allow RM standards without effective congestion control.
  - Multicast congestion control is difficult!
- **The IRTF Reliable Multicast Research Group is developing a congestion control mechanism for one-to-many bulk-transfer type applications.**
  - Based on a model of the throughput TCP would achieve across a network path with the same loss rate and round trip time.
  - Still research, but looks promising.
- **Perhaps pass this to the IETF in Q3 1999?**

Router Support

- **Most current RM protocols work end-to-end.**
  - The routers just forward packets.
- **Some RM protocols build trees.**
  - Hosts in the middle retransmit and aggregate feedack.
  - The routers just forward packets.
- **What if the routers themselves could help out?**
  - They could behave like RM helpers.
  - RM protocols might work better?
  - Transport state in routers may make this expensive?

A Brief Tour of Some Protocols

- **SRM: Scalable Reliable Multicast**
  - NACK-Based
- **RMTP-II: Reliable Multicast Transport Protocol - II**
  - ACK-Tree Based
- **PGM: "Cisco's PGM"**
  - NACK-Based with Router Support

- **Packet-level FEC.**

SRM: Scalable Reliable Multicast

- **Many senders send data.**
- **Receivers notice loss:**
  - gaps in sequence space
  - periodic messages detect tail loss
- **Receivers multicast a NACK to the group.**
  - random timers suppress duplicates.
- **Anyone can multicast a retransmission to the group.**
  - random timers suppress duplicates.

RMTP-II

- **One sender sends data.**
- **Receivers are arranged into a tree**
  - sender
  - intermediate nodes
  - leaf receivers
- **All receivers and send ACKs for every packet**
  - these go to intermediate nodes.
  - intermediate nodes retransmit downtree.
  - intermediate nodes aggregate ACKs
  - aggregates flow up the tree to sender.

PGM

- **One sender sends data.**
- **Receivers notice loss:**
  - gaps in sequence space
- **Receivers multicast a NACK to the group.**
  - A local PGM router confirms the NACK
  - Remembers the interface it arrived on.
- **PGM Router forwards a NACK upstream.**
  - NACK travels PGM-hop by PGM-hop back to the sender pinning up temporary state on the way.
- **Sender retransmits the missing packet.**
  - Packet travels back down the trail of state left by the NACK.
  - State it torn down.
  - Packet only goes to receivers that need it.

Packet Level FEC

- **Can be applied to most RM protocols.**
- **Greatly reduces the need for retransmissions.**

- **Problem:**
  - Packet loss is not spatially correlated.
  - With a large group, almost every packet is lost by someone.
  - Retransmission load can be high.

- **Solution:**
  - Don't retransmit the data, retransmit a parity packet.

Reliable Multicast and the IETF

- **There's an IRTF Research Group on Reliable Multicast:**
  - http://www.east.isi.edu/rm
  - Main task: congestion control
- **There's an IRTF Research Group on Secure Multicast**
  - Closed group.
- **There are no IETF standards for RM yet.**
  - Congestion Control and Security are gating functions.
- **Expect the IETF to form one or more RM working groups in the next few months as the congestion control work starts to mature.**