Multicast

Multicast Routing

- Unicast: one source to one destination
- Multicast: one source to many destinations
- Main goal: efficient data distribution
  - Avoid data duplication within network

Overview

- IP Multicast Service Basics
- Routing: MOSPF/DVMRP
- Reliability: SRM
- Overlay Multicast
Example Applications

- Broadcast audio/video
- Push-based systems (e.g., BGP updates)
- Software distribution
- Web-cache updates
- Teleconferencing (audio, video, shared whiteboard, text editor)
- Multi-player games
- Other distributed applications

IP Multicast Architecture

IP Multicast Service Model

- Each group identified by a single IP address
- Variable Size:
  - Groups of any size; sparse or dense
- Variable Location:
  - Members may be located anywhere on Internet
- Dynamic membership:
  - Members can join and leave at will
- Many-to-many
- Not only one-to-many
- No central state
- Group membership not known explicitly
- Analogy:
  - Each multicast address is like a radio frequency, on which anyone can transmit, and to which anyone can tune-in.

IP Multicast Addresses

- Class D IP addresses
  - 224.0.0.0 – 239.255.255.255
  - 1 1 1 0 Group ID
- How to allocate these addresses?
  - Well-known addresses: IANA
  - Transient addresses: e.g., by “SDR” program
    - Assigned and reclaimed dynamically,
IP Multicast API

- Sending – same as before
- Receiving – two new operations
  - Join(group)
  - Leave(group)
  - Receive multicast packets for joined groups via normal IP-Receive operation
  - Implemented using socket options

Multicast Router Responsibilities

- Learn of the existence of multicast groups (through advertisement)
- Identify links with group members
- Establish state to route packets
  - Replicate packets on appropriate interfaces
  - Routing entry:
    - Src, incoming interface
    - List of outgoing interfaces

Overview

- IP Multicast Service Basics
- Routing: MOSPF/DVMRP
- Reliability: SRM
- Overlay Multicast

Routing Techniques

- Basic objective – build distribution tree for multicast packets
- Link-state multicast protocols
  - Routers advertise groups for which they have receivers to entire network
  - Compute trees on demand
  - Example: MOSPF
- Flood and prune
  - Begin by flooding traffic to entire network
  - Prune branches with no receivers
  - Example: DVMRP
Multicast OSPF (MOSPF)

- Add-on to OSPF
  - Recall: flood routing announcements, each node gets entire topology
  - Now each router also keeps track of multicast group members
  - Routers mark link-state advertisement with groups that it has members for
- Source-based trees
  - Shortest paths to a node form a spanning tree
  - Routing algorithm augmented to compute shortest-path distribution tree from a source to any set of destinations
  - Packets from each source are forwarded on this tree

Impact on Route Computation

- Problems?
  - \(O(N^2)\) state: one tree per potential sender
  - Can’t pre-compute multicast trees for all possible sources
- One solution: Compute on demand
  - When first packet from a source S to a group G arrives
  - Slow if sources send infrequently
- Another solution: Shared trees
  - One tree per multicast group
  - Requires a rendezvous point
    - Unicast to RP, then RP multicasts it along tree
    - E.G., PIM Sparse Mode

Distance-Vector Multicast Routing

- Add on to DV routing (e.g., RIP)
  - Recall: each node locally determines shortest-path “next hop” for each destination
- Router forwards a packet if
  - The packet arrived from the link used to reach the source of the packet
    - Reverse path forwarding check (RPF)
    - Shortest-paths to a source form a spanning tree
  - If downstream links have not pruned the tree
    - Initially send to all routers then prune away branches
Reverse Path Forwarding

Prune

Graft

Overview

- IP Multicast Service Basics
- Routing: MOSPF/DVMRP
- Reliability: SRM
- Overlay Multicast
Multicast Transport Properties

- IP Multicast service guarantees?
  - Best effort
- What other properties would applications want?
  - Reliability
  - Congestion/Flow Control
  - In-order delivery
  - Etc.
- Why doesn’t IP Multicast provide these?
  - End-to-end principle: Can build other properties on top just like IP unicast
- SRM tackles reliability

Straw man Reliability Solutions

- Why not have each member ACK the sender?
  - ACK implosion: each packet sent generates $N$ ACKs!
  - Requires sender to track all receiver state
- Why not have each member NACK the sender?
  - If data rate is slow, may not know that we’re missing the last packet
  - Loss near the sender generates lots of NACKs; many receivers could share a bottleneck
  - SRM uses NACKs but in a more intelligent fashion

SRM Design Assumptions

- Example Application: digital whiteboard
- Many-to-many
  - Any one in the group can send
- Named data units
  - E.G., 0000 => “point (3,4)”, 0001 => “line (3,4)-(1,2)”
  - Each object sent has globally unique name
- Cooperative recovery
  - Any member can supply lost data to any other member
  - E.G., each member buffers all data

SRM Basic Operation

- Multicast periodic session messages telling everyone the “latest seqno”
  - Aside: can use these to estimate RTT between members
- Loss detected (missing seqno) => multicast repair request (NACK)
  - Request sent after a timer with time picked from uniform distribution:
    $2C_1d_{AB}$ \((0, C_1 + C_2)d_{AB}\)
  - Suppress request if we see a request and $i++$
  - => nodes closer to loss send request sooner (on expectation)
  - => first request likely to suppress others (with reasonable $C_1, C_2$)
- Receive repair request && we have the data item => multicast repair response
  - Request sent after a timer picked from uniform distribution:
    $D_1d_{AB}$ \((0, D_1 + D_2)d_{AB}\)
  - => nodes closer to requestor will respond sooner (on expectation)
- Goal: Have few repair request/responses for the entire group when loss
SRM Operation Example

Adaptive Parameter Adjustment

• Can trade-off higher delay for lower request/response duplicates

**Probabilistic Suppression:** Higher $C_2 \Rightarrow$ higher expected delay, but less likely to have duplicates
  - First request will likely reach all others before other request timers expire

**Deterministic Suppression:** Members with lower $C_1$ will likely send requests earlier
  - Mechanism 1: reduce $C_2$ when send request
    - $\Rightarrow$ members near persistent loss will send sooner
  - Mechanism 2: reduce $C_2$ when sent requests but still receive duplicate requests from members much farther from source
    - $\Rightarrow$ request more likely to reach far away members first

Adaptive Adjustment Algorithm

• After sending request:
  - Decrease $C_1$

• Before setting timer:
  - If sent request already && seen dup requests from further away:
    • Decrease $C_2$
    • Dup requests > $T$
    • Increase $C_2$
    • Dup requests < $T$ && request delay > $D$
        • Decrease $C_2$

• Converge on optimal delay-duplicate tradeoff
• Basically the same for $D_1$,$D_2$

Other Issues

• Local Recovery: Scoping recovery requests/replies
  - Basic algorithm multicast them to entire group
  - Administrative boundaries + TTLs can scope requests/replies

• Congestion control:
  - Assume fixed rate
  - Why not reduce rate to bottleneck link?
    • $\Rightarrow$ one bottlenecked receiver slows down the whole group
Overview

• IP Multicast Service Basics
• Routing: MOSPF/DVMRP
• Reliability: SRM
• Overlay Multicast

Failure of IP Multicast

• Real world:
  - Not widely deployed even after 15 years!
  - Use carefully — e.g., on LAN or campus, rarely over WAN
  - Largest deployment: MBONE, using IP-tunnels to connect domains
• IP Multicast failings:
  - Scalability of routing protocols
    • Extra router state required
    • Hard to manage
    • Who gets to set up groups and when?
  - Hard to implement TCP equivalent
    • As we just saw with SRM
  - Chicken-egg: No real applications
    • Hard to get applications to use IP Multicast without existing wide deployment
  - Economics, policy: Hard to get inter-domain support
    • Who pays for packet duplication?

Supporting Multicast on the Internet

At which layer should multicast be implemented?

End System Multicast

UCSD

Berkeley

MIT

CMU

CMU1

CMU2
Potential Benefits Over IP Multicast

- Quick deployment
- All multicast state in end systems
- Simplifies support for higher level functionality
  - Reliability, congestion control, etc.

Concerns with End System Multicast

- Reality: Many users behind asymmetric DSL/Cable modems
  - Not enough upload bandwidth to forward!
  - => Must be leafs in the multicast tree
- Key Metric: Resource Index
  - Forwarding capacity/total bandwidth demand
  - Measured ESM video groups have RI of 1-2...
  - => Building feasible tree is challenging (+ dealing with group dynamics, etc.)

Important Concepts

- Multicast provides support for efficient data delivery to multiple recipients
- Requirements for IP Multicast routing
  - Keeping track of interested parties
  - Building distribution tree
  - Broadcast/suppression technique
- Build reliability, congestion control, in-order delivery on top
  - Just like with TCP/IP, but harder...
- Difficult to deploy new IP-layer functionality
- End system-based techniques can provide alternative
  - Easier to deploy