

# CSE 123b Communications Software

Spring 2003

Lecture 9: Multicast

Stefan Savage

## Midterm

- May 8<sup>th</sup> in class
- Covers material through 5/6 (Mobile IP)
  - Fair game: lectures, reading, homework, project #1
- Closed book, closed notes, but you can bring in one 8.5x11 sheet of paper with notes on it

May 5, 2003

CSE 123b – Lecture 9 – Multicast

2

## Last class

- Inter-domain routing
  - Routing between different organizations
  - Policy routing; not based on metrics
  - BGP
  - Economics of routing

May 5, 2003

CSE 123b – Lecture 9 – Multicast

3

## Today: Multicast routing

- Multicast service model
- Host interface
- Host-router interactions (IGMP)
- Multicast Routing
  - Distance Vector
  - Link State
  - Shared tree
- Limiters
  - Deployment issues
  - Inter-domain routing
  - Operational/Economic issues (SSM)

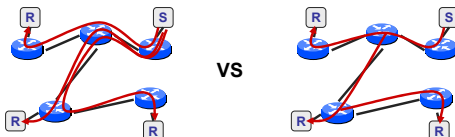
May 5, 2003

CSE 123b – Lecture 9 – Multicast

4

## Motivation

- Efficient delivery to multiple destinations (e.g. video broadcast)



- Network-layer support for one-to-many addressing
  - Publish/subscribe communications model
  - Don't need to know destinations

May 5, 2003

CSE 123b – Lecture 9 – Multicast

5

## IP Multicast service model

- **Communications based on groups**
  - Special IP addresses represent "multicast groups"
  - Anyone can join group to receive
  - Anyone can send to group
    - » Sender need not be part of group
  - Dynamic group membership – can join and leave at will
- **Unreliable datagram service**
  - Extension to unicast IP
  - Group membership not visible to hosts
  - No synchronization
- **Explicit scoping to limit spread of packets**

May 5, 2003

CSE 123b – Lecture 9 – Multicast

6

## Elements of IP Multicast

- **Host interface**
  - Application visible multicast API
  - Multicast addressing
  - Link-layer mapping
- **Host-Router interface**
  - IGMP
- **Router-Router interface**
  - Multicast routing protocols

May 5, 2003

CSE 123b – Lecture 9 – Multicast

7

## Host interface

- Senders (not much new)
  - Set TTL on multicast packets to limit "scope"
    - » Scope can be administratively limited on per-group basis
  - Send packets to *multicast address*, represents a group
  - Typically UDP-based transport
- Receivers (two new interfaces)
  - Join multicast group (group address)
  - Leave multicast group (group address)
  - Typically implemented as a socket option in most networking API

May 5, 2003

CSE 123b – Lecture 9 – Multicast

8

## Multicast addressing

- Special address range:
  - Class D (3 MSBs set to 1) 224.0.0.1- 239.255.255.255
  - Reserved by IANA for multicast
- Which address to use for a new group?
  - No standard
  - Global random selection
  - Per-domain addressing (MASC, GLOP)
- Which address to use to join an existing group?
  - No standard
  - Separate address distribution protocol (may use multicast)

May 5, 2003

CSE 123b – Lecture 9 – Multicast

9

## Link-layer multicast

- Many link-layers protocols have multicast capability
  - Ethernet, FDDI
- Translate IP Multicast address into LL address
  - E.g. Map 28 bits of IP MC address in 23bit Ethernet MC addresses
  - Senders send and receive on link-layer MC addresses
  - Routers must listen on all possible LL MC addresses
- Not an issue for point-to-point links

May 5, 2003

CSE 123b – Lecture 9 – Multicast

10

## Internet Group Management Protocol (IGMP)

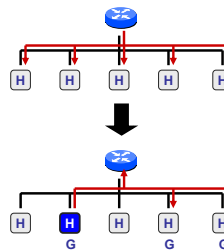
- **Goal: communicate group membership between hosts and routers**
- Soft-state protocol
  - Hosts explicitly inform their router about membership
  - Must periodically refresh membership report
  - Routers implicitly timeout groups that aren't refreshed
  - Why isn't explicit "leave group" message sufficient?
- Implemented in most of today's routers and switches

May 5, 2003

CSE 123b – Lecture 9 – Multicast

11

## How IGMP works (roughly)



- Router broadcasts *membership query* to 224.0.0.1 (all-systems group) with *t<sub>tl</sub>*=1
- Hosts start random timer (0-10 sec) or each group they have joined

- When a host's timer expires for group G, send *membership report* to group G, with *t<sub>tl</sub>*=1
- When a member of G hears a report, they reset their timer for G
- Router times out groups that are not "refreshed" by some host's report

May 5, 2003

CSE 123b – Lecture 9 – Multicast

12

## Multicast routing

- **Goal: build distribution tree for multicast packets**
  - Efficient tree (ideally, shortest path)
  - Low join/leave latency
- Several approaches
  - Distance Vector/Link State
    - » Leverage existing unicast routing protocols
  - Shared tree
    - » Unicast/multicast hybrids

May 5, 2003

CSE 123b – Lecture 9 – Multicast

13

## Multicast routing taxonomy

- **Source-based tree**
  - *Separate shortest path tree for each source*
- **Flood and prune (DVMRP, PIM-DM)**
  - » Send multicast traffic everywhere
  - » Prune edges that are not actively subscribed to group
- **Link-state (MOSPF)**
  - » Routers flood groups they would like to receive
  - » Compute shortest-path trees on demand
- **Shared tree (CBT, PIM-SM)**
  - *Single distributed tree shared among all sources*
  - Specify rendezvous point (RP) for group
  - Senders send packets to RP, receivers join at RP
  - RP multicasts to receivers; Fix-up tree for optimization

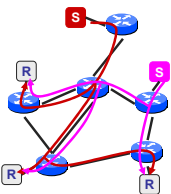
May 5, 2003

CSE 123b – Lecture 9 – Multicast

14

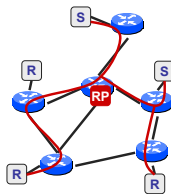
## Source-based vs Shared

Source-based tree



- Efficient trees; low delay, even load
- Per-source state in routers (S,G)
- Good for dense-area multicast

Shared-tree



- Higher delay, skewed load
- Per-group state only (G)
- Efficient for sparse-area multicast

May 5, 2003

CSE 123b – Lecture 9 – Multicast

15

## Flood and Prune (DV)

- Extensions to unicast distance vector algorithm
- Goal
  - Multicast packets delivered along shortest-path tree from sender to members of the multicast group
  - Likely have different tree for different senders
- Distance Vector Multicast Routing (DVMRP) developed as a progression of algorithms
  - Reverse Path Flooding (RPF)
  - Reverse Path Broadcast (RPB)
  - Reverse Path Multicast (RPM)

May 5, 2003

CSE 123b – Lecture 9 – Multicast

16

## Reverse Path Flooding (RPF)

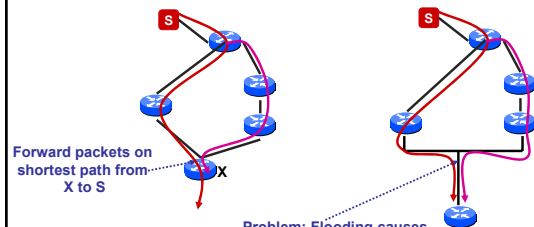
- Observation: Shortest-path multicast tree is subtree of shortest-path broadcast tree
- Approach: Use shortest-path broadcast tree
- Use **reverse path** to determine shortest path
  - Router forwards a packet **from S** iff received from the shortest-path link **to S**
  - Exactly what is in entry in forwarding table
    - » To reach S along shortest path, use link L
    - » If received packet from S on L, it came along shortest path
- How are packets forwarded?
  - **Flooding** – forward packets to multicast address out to all links except incoming link (hence reverse path flooding)

May 5, 2003

CSE 123b – Lecture 9 – Multicast

17

## Example: Reverse Path Forwarding



May 5, 2003

CSE 123b – Lecture 9 – Multicast

18

## Solution: Reverse Path Broadcast (RPB)

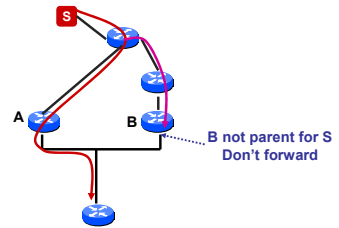
- Flooding vs. broadcast
  - With flooding, a single packet can be sent along an individual link multiple times
    - » Each router attached to link can potentially forward same packet
  - RPB sends a packet along a link at most once
- Approach: Define **parent** and **child** routers for each link
  - Relative to each link and each source S
  - Router is a parent for link if it has minimum path to S
  - All other routers on the link are children
  - Only parent router is allowed to forward multicast packets on link
- How to decide parent and children routers for link?
  - In routing updates; router determines if is parent

May 5, 2003

CSE 123b – Lecture 9 – Multicast

19

## Example: Reverse Path Broadcasting



May 5, 2003

CSE 123b – Lecture 9 – Multicast

20

## Reverse Path Multicast (RPM)

- Problem: Still **broadcasting** up to leaf networks
- Idea: Instead of **actively building tree**, use reports to **actively prune tree**
- Start with a full broadcast tree to all links (RPB),
  - Prune (S,G) at leaf if it has no members
    - Send Non-Membership Report (NMR) to prev-hop for S
- If all children of router R prune (S,G)
  - Send NMR for (S,G) to parent of R
- Soft-state management (must refresh NMR or rejoin)
- New group member sends graft (anti-prune) message

May 5, 2003

CSE 123b – Lecture 9 – Multicast

21

## Link State

- Use existing link-state routing algorithm (e.g. OSPF)
- Idea: include active groups in LSPs
  - Each router can compute shortest path tree from source to all destinations for any group
  - Trigger new flood on group membership change
- Performance issues
  - Expensive to precompute all (S,G) trees
  - Keep cache of trees and compute new trees on demand when new (S,G) packet arrives
  - Workload/topology dependant
- Best known example: MOSPF

May 5, 2003

CSE 123b – Lecture 9 – Multicast

22

## Shared tree approaches

- Unicast packets to Rendezvous Point (RP), which multicasts packet on shared tree
- Tree construction
  - Receivers send join messages to RP
  - Intermediate routers install state to create per-group tree
  - Key advantage is routers only store O(G) state
  - Potential optimizations: reroute to source-specific trees for local group members or high data-rate sources
    - Example: CBT, PIM-SM
- Issues
  - Delay, fault tolerance, RP selection

May 5, 2003

CSE 123b – Lecture 9 – Multicast

23

## IP Multicast today

- IP Multicast has generated 1000s of papers, but has not been widely deployed in the Internet...
- Why?
  - General deployment difficulties (Mbone)
  - Inter-domain multicast complexity
  - Economics of multi-source multicast

May 5, 2003

CSE 123b – Lecture 9 – Multicast

24

## Multicast evolution

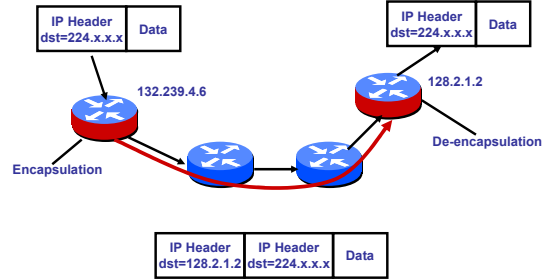
- How to deploy a new network-layer service?
  - Difficult to change router software
  - Difficult to change all routers
- Mbone (tunneling)
  - Special multicast routers (built from PCs/Workstations)
  - Construct virtual topology between them (overlay)
  - Run routing protocol over virtual topology
  - Virtual point-to-point links called **tunnels**
    - » Multicast traffic encapsulated in IP datagrams
    - » Multicast routers forward over tunnels according to computed virtual next-hop

May 5, 2003

CSE 123b – Lecture 9 – Multicast

25

## Tunnelling

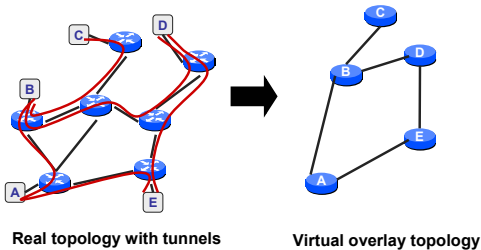


May 5, 2003

CSE 123b – Lecture 9 – Multicast

26

## Virtual overlay network



May 5, 2003

CSE 123b – Lecture 9 – Multicast

27

## Mbone Pro/Con

- Success story
  - Multicast video to 20 sites in 1992
  - Easy to deploy, no explicit router support
  - Ran DVMRP and had 100s of routers
- Drawbacks
  - Manual tunnel creation/maintenance
  - Inefficient
  - No routing policy (single tree)
  - Why would an ISP deploy a new mbone node?

May 5, 2003

CSE 123b – Lecture 9 – Multicast

28

## Inter-domain multicast routing

- Technical issues
  - How to exchange reachability information?
  - How to construct trees?
  - Who controls RP in shared tree?
- MBGP: reachability to multicast sources per prefix
- PIM-SM: shared tree multicast protocol
- MSDP: RP per group per AS, communication presence of group sources between RPs
- BGMP: alternative proposal, single shared tree with group addresses owned by individual ASs

May 5, 2003

CSE 123b – Lecture 9 – Multicast

29

## Economic issues

- ISP router migration cycle
  - Can't afford new routers on edge
- Domain independence
  - Do I want my customers MC controlled by an RP in a competitors domain?
  - Why run an RP for which I have no senders or receivers?
- Billing model
  - Inconsistent with input-rate-based billing
  - No group management (how big is group?)
- Group management
  - Who is in the group? Who can send? Security
- Network management
- Limited Multicast addresses

May 5, 2003

CSE 123b – Lecture 9 – Multicast

30

## Summary

---

- Multicast service model
  - One-to-many, anonymous communication
  - Simple host interface
- Per-source tree routing
  - Efficient trees
  - S\*G state explosion for large networks/groups
- Shared tree
  - More complex, fragile, hard to manage
  - Trees inefficient by as much as 2x
  - Only requires G state on routers
- Operational and Economic issues matter in deployment
- Killer app not found

## For next time...

---

- Mobile IP and Mobile routing; P&D 4.2.5