Broadcast and Multicast Routing

Daniel Zappala

CS 460 Computer Networking
Brigham Young University
Group Communication

- How can the Internet provide efficient group communication?
  - send the same copy of a data stream (e.g. TV show, teleconference) to a group of users
  - need to find where everyone is located (routing)
  - need to avoid sending a separate copy to everyone
**Choices**

- **unicast**: send a separate copy of each packet to each host
- **broadcast**: send one copy of each packet, the network will replicate it and deliver it to all hosts
  - **broadcast provides efficient network flooding**
- **multicast**: send one copy of each packet, the network will replicate it and deliver it to only those hosts that want it
  - **multicast provides efficient group communication**

Duplicate creation/transmission
Broadcast
Broadcast

- send a copy of each packet to all your neighbors
  - need to eliminate duplicates
  - sequence numbers: drop a sequence number previously seen
  - reverse path forwarding: accept the packet only on the incoming interface used to send packets to the source
Spanning and Steiner Trees

- **spanning tree**
  - connect all routers in the entire Internet
  - easy to build a minimum cost tree

- **Steiner Tree**
  - connect only those routers with multicast members for a particular group
  - NP-complete (one of the original 21!)
  - many different heuristics, but often centralized
  - not used in practice: complex, hard to create a good and practical decentralized algorithm
Multicast Service Model
Internet Multicast Service Model

- classic model - developed by Steve Deering
- logical multicast group - a collection of hosts
- any host can join/leave the group at any time
- any sender can send to the group at any time
- no network report of group membership
Group Membership and Routing

- need IGMP to report group membership from hosts to routers
- need multicast routing protocol to get data from any sender to current set of group members
IGMP: Internet Group Management Protocol

- **host**: sends IGMP report when application joins multicast group
  - application uses IP_ADD_MEMBERSHIP socket option
- **router**: sends IGMP query at regular intervals
  - needs only one active member to respond
**IGMP Messages**

- **type**
  - 0x11 = membership query, can be sent to all groups (group address set to zero) or a specific group
  - 0x16 membership report
  - 0x17 leave group

- **maximum response time**: bound on member response time
  - set timer between 0 and max
  - first timer to fire responds
  - other members hear report and suppress their own
<table>
<thead>
<tr>
<th>IGMP Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>version 1</strong></td>
</tr>
<tr>
<td>• router sends membership query on LAN to all hosts</td>
</tr>
<tr>
<td>• hosts respond with membership report for all groups</td>
</tr>
<tr>
<td>• randomize delay before responding</td>
</tr>
<tr>
<td>• implicit leave by not responding to query</td>
</tr>
<tr>
<td><strong>version 2</strong></td>
</tr>
<tr>
<td>• group-specific query</td>
</tr>
<tr>
<td>• leave group message</td>
</tr>
<tr>
<td>• host that responds to query can send leave</td>
</tr>
<tr>
<td>• router then sends group-specific query to check if any other hosts are members</td>
</tr>
<tr>
<td><strong>version 3</strong></td>
</tr>
<tr>
<td>• source-specific joining</td>
</tr>
<tr>
<td>• source-specific pruning</td>
</tr>
</tbody>
</table>
Multicast Routing Problem

- find a tree (or trees) connecting routers that have local group members
DVMRP
DVMRP: Distance Vector Multicast Routing Protocol

- first multicast routing protocol
- developed by Steve Deering as part of his dissertation at Stanford
- **reverse path forwarding**

```
1  if packet received on link used to send packets to source:
2       send packet on all other links
3  else:
4    ignore packet
```

- **pruning**
  - if a router has no local members and no downstream routers with members, send prune message upstream to cancel forwarding
- **flood and prune**: reverse path forwarding + pruning
Reverse Path Forwarding
Flood and Prune
DVMRP

- **soft state**
  - prune state at routers eventually times out and is deleted
  - multicast packets flooded down that branch again
  - routers must prune again unless there are new members

- **grafting**
  - if a new member joins, router can send a graft message to cancel prune state

- **implementation**
  - initially run on Sun workstations using mrouted
  - built the MBone: a set of hosts that connect to each other using tunnels and run mrouted on the Internet’s first virtual network
  - later implemented in commercial routers
CBT
CBT: Core-Based Trees

- Tony Ballardie, UK
- builds a single multicast tree shared among all group members
  - avoid flooding
  - send multicast data only to group members
- core-based tree
  - determine a single router to act as the core
  - routers with members send a join message via unicast to core
  - intermediate routers intercept message and create a branch of a tree
  - forms a shortest-path tree, rooted at the core
Core-Based Tree
Explicit Join

- only those routers with members need to join
- by default other routers don’t get data
- can use soft state (refresh join periodically) or an explicit teardown
Not Shortest Path!
PIM
PIM: Protocol-Independent Multicast

- Steve Deering, Deborah Estrin, Van Jacobson, others
- shortest-path trees with explicit joining
  - members first join a center-based tree to hear about new sources (bootstrapping)
  - sources unicast data to center, which relays to members
  - members can then join a separate, shortest-path tree for each source
Switch to Shortest-Path Tree
Basic Multicast Routing Problem

- who are the sources?
  - flood data from all sources (DVMRP – not scalable)
  - join to a core-based tree, sources give data to core (CBT, not shortest paths)
  - use core-based tree for rendezvous, switch to shortest paths once sources reveal themselves (PIM)

- PIM is still not scalable
  - still need to flood identity of core to entire Internet
Where Are We Now?
Multicast Development

- DVMRP, MBone
- MOSPF (OSPF with multicast routing)
- interoperability
- CBT (not widely deployed)
- PIM (more scalable)
- BGMP (hierarchical multicast routing)
- SSM (source-specific multicast)
SSM: Source-Specific Multicast

- Hugh Holbrook, Stanford (Cheriton, same advisor as Deering)
- real problem: multicast address allocation
  - each group needs a unique address
  - only 28 bits of addresses
  - randomization runs into birthday problem rather quickly – need global coordination
- easy solution: each source has its own 28-bit address space
  - identify group as combination of source-group instead of just group
- build shortest-path trees using explicit join
Joining and SSM Tree
Differences from PIM

- Join(S, G) instead of Join(G)
  - G is no longer globally unique
- Change Internet’s multicast model: only one source per group
  - New source can relay through primary source
  - Members can then join source-specific tree
- Permanent multicast addresses
  - Previously had to allocate and deallocate from a shared space
  - Now we can advertise SSM (S, G) addresses on TV, web pages
  - “Join multicast.cnn.com for our live video feed”
Why Don’t We Have Deployed Multicast Services?

- **SSM**
  - consensus as best solution
  - easy to deploy Internet-wide: scalable
  - easy transition from PIM’s sparse mode

- **status**
  - Cisco has implementations in its routers
  - Sprint actually has multicast services you can buy
  - companies deploy multicast internally and via multicast VPNs
  - used for pushing to content caches

- **but ...**
  - no “killer” application (television distribution?)
  - no demand from public
  - nobody watches live events synchronously any more
    (except for General Conference and the Super Bowl)
  - no incentive for ISPs to deploy (no extra charges)
Application-Layer Multicast

- *if the ISPs won’t deploy it, then we will run it ourselves*
- run multicast in application layer
  - organize members into a virtual network using TCP connections
  - can build on top of some Peer-to-Peer overlays (e.g. Chord, Tapestry)
  - penalty: higher delay, less efficient
- advantages
  - build on top of TCP
  - host A sends to B and C, B sends to D and E, C, sends to F, G and H ...
  - get reliability, flow control, congestion control for free
  - these are problems that native multicast protocols have a very hard time solving adequately
- essentially back to the MBone, but with automatic configuration, application-specific software