Multicast Protocols

IGMP – IP Group Membership Protocol
DVMRP – DV Multicast Routing Protocol
MOSPF – Multicast OSPF
PIM – Protocol Independent Multicast

Multicast in local area networks

Multicast addresses
IGMP – Internet Group Membership Protocol
Multicast addresses

32 bits

<table>
<thead>
<tr>
<th>1110</th>
<th>28 bits - multicast group address</th>
<th>1111</th>
<th>experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 224.0.0.0 - 239.255.255.255 | Local network control |
| 224.0.0.0 - 239.255.255.255 | All systems |
| 224.0.0.0 - 239.255.255.255 | All routers |
| 224.0.0.0 - 239.255.255.255 | All OSPF routers |

- Sender does not need to belong to G.
- Address space is flat.

Multicast in broadcast networks

- In broadcast networks only one copy should be sent of a multicast packet
- Some broadcast network support group addresses
  - E.g. Ethernet
  - Group address is based on the IP address
    - Place low-order 23 bits of multicast address into low-order 23 bits of MAC address 01-00-5E-00-00-00
    - No ARP required
- Point-to-point links need no special arrangements
Routers discover multicast receivers using IGMP

- IGMP = Internet Group Membership Protocol
- Version 2 defined in RFC-2236, version 3 in RFC-3376
- Runs directly over IP (protocol type 2)
- Used locally within a network
  - TTL=1 in all IGMP messages
- Router with lowest IP address is active on a network
- Routers do not need to know the exact members, only whether there are members for a specific group

IGMPv2 - Internet Group Management Protocol

<table>
<thead>
<tr>
<th>Type</th>
<th>Max Resp Time</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Address (GA)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Sent to "all systems" multicast address 224.0.0.1
- "Leave group" message sent to "all routers" address 224.0.0.2

Host waits random [0...Max Resp Time] prior to response and suppresses its response if it sees another response to the same group.

Router

- 0x11 = Membership Query
  - [General (GA=0) / Group specific (GA≠0)]
- 0x16 = V2 Membership Report [Group]
- 0x12 = V1 Membership Report
- 0x17 = Leave Group
- 0x11 = Membership Query [Group specific]

Are there other members left?
IGMPv3 adds selective reception from sources within a group

<table>
<thead>
<tr>
<th>Type=0x11</th>
<th>Max Resp Time</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>Number of Sources (N)</td>
<td></td>
</tr>
</tbody>
</table>

Source Address

0x11 = Membership Query

Variants:
- General query (GA=0 and number of sources=0)
- Group specific query (GA≠0 and number of sources=0)
- Group and source specific query (GA≠0 and number of sources>0)

0x22 = V3 Membership Report

Can exclude listed sources within a group or include only listed sources within a group

MBone
**MBone – an overlay multicast Internet**

- Multicast backbone (MBone) was deployed to support research
  - Enable multicast applications without waiting for full availability of multicasting standards
- Started in 1992
- Uses tunnels to link multicast islands
  - Previously as source routed packet
  - Now with encapsulation
- Uses DVMRP and IGMP

**MBone overlay is based on workstations running DVMRP**

Tunneling is used to bypass unicast sections of the Internet
Experimental routing protocols have been developed for MBone

<table>
<thead>
<tr>
<th>Tree type</th>
<th>Shared tree</th>
<th>Source based trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Center based tree</td>
<td>Flood and prune</td>
</tr>
<tr>
<td>Protocols</td>
<td>PIM Sparse*</td>
<td>DVMRP</td>
</tr>
<tr>
<td></td>
<td>Core Based Tree*</td>
<td>PIM Dense*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOSPF</td>
</tr>
</tbody>
</table>

* These rely on unicast routing protocol to locate multicast sources.
  (The other ones can route multicast on routes separate from the unicast routes)

DVMRP – Distance Vector Multicast Routing Protocol
DVMRP – Distance Vector Multicast Routing Protocol

- First multicast protocol in the Internet (1988)
- Distance vector routing protocol similar to RIP
  - Except that sources are like destinations in RIP
- Routers maintain separate multicast routing tables
- Uses the reverse-path-forwarding (RPF) algorithm
- Routers exchange
  - Distance in hops (reverse path distance)
  - IP address and mask of source
- Tunnels explicitly configured with
  - Destination router
  - Cost
  - Threshold

DVMRP is used for multicast routing in the MBone

- DVMRP messages are IGMP messages (IP protocol=2=IGMP, TTL=1)

DVMRP header:

<table>
<thead>
<tr>
<th>Type=0x13</th>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>Minor vers =0xff</td>
<td>Major vers = 3</td>
</tr>
</tbody>
</table>

Version 3 (1997) presented in this course
Probes are used for neighbor discovery

**Probes** are exchanged on tunnel and physical interfaces. They contain the list of neighbors on the interface.
- If empty, this is a leaf network managed by IGMP.
- Multicasts are not exchanged until a two-way neighbor relationship is established.
- Routers see each other's versions and capability flags for compatibility.
- Keepalive for fault detection and restart detection: sent every 10s, timeout set at 35s.

---

**DVMRP uses the concept of dependent downstream routers**

- DVMRP uses the route exchange as a mechanism for upstream routers to determine if any downstream routers depend on them for forwarding from particular source networks.
  - Implemented with ”poison reverse”:
  - If a downstream router selects an upstream router as the best next hop to a source, it echoes back the route with a metric = original metric + inf.
Route reports are used to build the source based trees

Each DVMRP router periodically (60s) broadcasts to its neighbors
- the list of pairs (source, metric)
- source aggregation according to CIDR may be used

- The receiving MC router calculates the previous hop on each source's multicast path = the DVMRP router that reports shortest distance from the source
- If equal distance ⇒ choose smallest IP address

DVMRP Report \[\text{inf} < \text{metric} < 2*\text{inf}\]
(compare to poisonous reverse, \(\text{inf}=32\))

The multicast algorithm of DVMRP is based on Reverse Path Forwarding (RPF)

- At first multicast from RPF interface a Forwarding Cache Entry \([S,G]:(u,\text{list})\) is created using the DVMRP routing table
  - The list contains all downstream routers that have reported dependency on \(S\)
- The router is designated forwarder for downstream nodes
- If the designated forwarder becomes unreachable, another router assumes the role of designated until it hears from a better candidate
List of dependent neighbors is used to minimize the multicast tree

- Initially list may contain all multicast interfaces but the upstream interface
- Downstream address is removed from list if
  - It is a leaf network and G is not in IGMP DB for this phys. network
  - Downstream node has selected another designated forwarder
  - Prune received from all dependent neighbors on this interface

Prunes minimize the multicast tree

- If known dependent neighbor
  - If mask and mask=sent mask with (S,G)
    - Prune all sources in network (S, mask)
  - If prune is already active
    - reset timeout to new value
  - If all dependent neighbors have sent prunes
    - If no group members on the mc-interface
      - Remove u from all Forwarding Cache entries
    - If last u
      - Send prune

- If Mcasts keep arriving (3s)
  - Resend Prune with exponential backoff = double interval each time
  - Remove Cache Entry
Grafts are used to grow the tree when a new member joins the group

- The graft is always acknowledged
  - if no multicast, nobody is sending
- If no ack is received, the graft is resent with exponential backoff retransmissions
- The graft is forwarded upstream if necessary

On probe timeout caches are flushed

- All routes learned from A → hold-down
- All downstream dependencies ON A are removed
- If A was designated forwarder, a new one is selected for each (source, group) pair
- Forwarding cache entries based on A are flushed
- Graft acks to A are flushed.
- Downstream dependencies are removed.
  - If last, send prune upstream
Route hold-down is a state prior to deleting the route

- Routes expire on report timeout or when an infinite metric is received
- An alternate route (that in RIP caused temporary loops) may exist
- Routers continue to advertise the route with inf metric for 2 report intervals – this is the hold-down period
- All forwarding cache entries for the route are flushed
- During hold-down, the route may be taken back, if
  - metric < inf, and
  - metric = SAME, and
  - received from SAME router

PIM – Protocol Independent Multicast
PIM – Protocol Independent Multicast

- Most popular multicast protocol
- Two modes of operation
  1. Dense mode
  2. Sparse mode
- Independent of any particular unicast routing protocol
- Uses unicast routing table
  - Simple protocol
  - Assumes the links are symmetric
  - No tunnels
- Messages sent in IGMP packets

PIM Dense Mode

- For dense multicast groups
  - Dense: The probability is high that a small randomly picked area contains at least a group member, e.g. LAN
- Based on RPF / "flood-and-prune"
- Principle similar to DVMRP
  - Simpler
  - Less efficient
PIM-DM implementation of RPF

Receive multicast packet $P$ from source $S$ to group $G$ on interface $U$

- Equal-cost multipath
  - no
  - yes

- $U$ used to send packets to $S$
  - no
  - yes

- $M(n, S, G) = 1$ for all $n$
  - no
  - yes

For all $n$ where $M(n, S, G) \neq 1$: Forward $P$ on interface $n$

$M$ is a cache for prune messages:

- $M(n, S, G) = 1$ if a prune($S, G$) has been received on interface $n$
- Least recently used entries may be dropped

PIM-DM – Pruning

- R = receiver
- prune

PIM-DM – Pruning on broadcast networks

- Prune messages sent to "all-routers" (224.0.0.2)

```
1. multicast
2. prune
3. delay the prune
4. join
5. cancel the prune
```

PIM-DM – Resolving multicasts received on multiple path

```
A and C receive packets for (S,G) from another router
assert(S,G,d_A)
assert(S,G,d_C)

if d_{other router} < d_{my distance}:
prune interface
```

PIM Sparse Mode

- RFC 2362
- Uses the center-based tree algorithm
- Evolved from the Core-Based Tree (CBT) protocol
- Rendezvous point (=center) connects the receivers with the senders
- Receivers must explicitly join

PIM-SM route entries

- Route entry includes
  - source address
  - group address
  - incoming interface
  - list of outgoing interfaces
  - timers, flags
- Packets match on the most specific entry
  - (S,G) – a specific source in a specific group
  - (*,G) – all sources in a specific group
  - (*, *, RP) – all groups that hash to a specific RP
PIM-SM example (1)

- Join packets are sent toward the RP
  - Address=G, Join=RP, wildcard (WC) bit, RP-tree (RPT) bit, Prune=(empty)
- Intermediate routers set up (*, G) state and forward the join

PIM-SM example (2)

- Senders send packets to RP encapsulated in register messages
- RP resends packets on the tree
- RP may construct a (S,G) entry, and send periodic joins to the sender
PIM-SM example (3)

- If the last-hop router (K and A) sees many packets from the source, it can switch from a shared tree to a shortest path tree for (S,G).
- It sends a join directly to the source, and prunes the previous path.

![Diagram](image)

PIM-SM example (4)

- Copies of the packets are still sent to RP.
- Join/prune messages are sent periodically for each route entry.

![Diagram](image)
Selection of Rendezvous Point

- A small group of routers configured as bootstrap routers candidates
- One of them selected as bootstrap router (BSR) for the domain
- BSR periodically sends Bootstrap messages through the domain
- A set of routers are configured as candidate RPs
  - typically same as candidate BSRs
- Candidate RPs periodically unicast Candidate-RP-Advertisements to the BSR, which includes them in the Bootstrap message
  - Candidate RP’s own address
  - Optional group address and mask length
- The RP is selected by a hash function from the valid candidate RPs
  - All routers use the same hash functions, therefore all routers select the same RP for a given group

PIM-SM can interoperate with DVMRP and other multicast protocols

- PIM Multicast Border Routers (PMBR) connects PIM-SM with other multicast protocols
Considerations

- PIM can switch from sparse mode to dense mode
  - Controlled by a parameter, which defines when the group is dense enough
- The RP may be a single point of failure
- The RP may be a bottle-neck

MOSPF – Multicast extensions to OSPF
MOSPF – Multicast extensions to OSPF (1)

• Idea: if the location of receivers is known to all routers, multicast should be possible to exactly the receivers only!
• MOSPF is an extension of OSPF, allowing multicast to be introduced into an existing OSPF unicast routing domain.
• Unlike DVMRP, MOSPF is not susceptible to the normal convergence problems of distance vector algorithms.
• MOSPF limits the extent of multicast traffic to group members only
  – Desirable for high-bandwidth multicast applications or limited-bandwidth network links (or both).

MOSPF – Multicast extensions to OSPF (2)

• Unlike OSPF, MOSPF does not support multiple equal-cost paths
• MOSPF calculates the source-based trees on demand
• MOSPF can be, and is in isolated places, deployed in the MBONE. A MOSPF domain can be attached to the edge of the MBONE, or can be used as a transit routing domain within the MBONE’s DVMRP routing system.
• Defined in RFC 1584
MOSPF can be deployed gracefully

- Introduces multicast routing by
  - adding a new type of LSA to the OSPF link-state database
  - adding calculations for the paths of multicast packets
- The introduction of MOSPF to an OSPF routing can be gradual
  - Multicast capability marked with a M-bit in the option flag
  - Routers without multicast capability are ignored in calculating multicast routes ⇒ MOSPF will automatically route IP multicast datagrams around routers incapable of multicast routing
  - No tunnels ⇒ there may be a unicast path, but no multicast path

An MOSPF Routing Domain

E.g. G1 = 226.1.7.6
Group m-LSA created and flooded when e.g. host on 128.186.4.0 joins G1.
Group-membership-LSA is created and flooded when a user joins an multicast group using IGMP

LS Type 6 = Group Membership LSA:

<table>
<thead>
<tr>
<th>LS age</th>
<th>Options=E</th>
<th>LS type=6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link state ID</td>
<td>226.1.7.6 (group G1)</td>
<td></td>
</tr>
<tr>
<td>Advertising router</td>
<td>128.186.4.1 (router E)</td>
<td></td>
</tr>
<tr>
<td>LS sequence number</td>
<td>0x80000001</td>
<td></td>
</tr>
<tr>
<td>LS checksum</td>
<td>0x0</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Referenced LS type</td>
<td>2 (network)</td>
<td></td>
</tr>
<tr>
<td>Referenced Link State ID</td>
<td>128.186.4.1</td>
<td></td>
</tr>
</tbody>
</table>

MOSPF calculates shortest-path trees on demand

• A separate tree is calculated for each combination of group and source
• Result is stored in Multicast Forwarding Cache Entry
• Hierarchy reduces the number of calculations

• Lines with label A are pruned when removing redundant shortest paths.
• Lines with label B are pruned when removing links that do not lead to G1
The Multicast Forwarding Cache Entry stores multicast path routing info

- For each source network and group:
  - When network conditions change paths are recalculated:
    - Cache entries must be deleted, when changed LSAs are received:
      - Router-LSA, Network-LSA (on router or link failure or cost change) ⇒ Delete all entries since it is not possible to tell which are affected.
      - Group-Membership-LSA ⇒ Delete entries of that group.
    - Hierarchy ⇒ The farther away the change is the fewer cache entries are deleted.
    - When the first packet arrives to a multicast group, the routes are recalculated.

<table>
<thead>
<tr>
<th>Router or network for multicast reception</th>
<th>List of interfaces, multicasts must be sent</th>
<th>Metrics to nearest group member</th>
</tr>
</thead>
</table>

On demand route calculations use Dijkstra’s shortest path first algorithm

- Calculation is rooted on the source
  - not in the current router as for unicast
- For a new multicast, every router performs the same calculation
- Stub networks do not appear in MOSPF calculation
  - e.g router F
- For equal cost routes, the previous hop router with the highest address is chosen
  - e.g. G over E
Summary of Multicast Protocols for the Internet

<table>
<thead>
<tr>
<th>Tree type</th>
<th>Shared tree</th>
<th>Source based trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Center based tree</td>
<td>Flood and prune</td>
</tr>
<tr>
<td>Protocols</td>
<td>PIM Sparse*</td>
<td>DVMRP</td>
</tr>
<tr>
<td></td>
<td>Core Based tree*</td>
<td>PIM Dense*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOSPF</td>
</tr>
</tbody>
</table>

* These rely on unicast routing protocol to locate multicast sources. (The other ones can route multicast on routes separate from the unicast routes)

- For shared tree protocols an additional step of finding the Core or Rendezvous Point must be performed.
- Directories are useful on service management level.