Multicast Protocols

IGMP - IP Group Membership Protocol
DVMRP - DV Multicast Routing Protocol
MOSPF - Multicast OSPF

(see notes pages for some slides!)

IGMPv2 - Internet Group Management Protocol implements Group Membership

- All syst MC group 224.0.0.1
- Host sends:
  - 0x11 = Membership Query \([\text{General}(\text{GA}=0)/\text{Group spec}]\)
  - 0x16 = v2-Membership Report [Group]
  - 0x12 = v1-Membership Report
  - 0x17 = Leave Group [all routers mc g=224.0.0.2]

- IGMP runs directly on IP as protocol nr 2.
- TTL == 1 in all IGMP msgs
- Host will wait random[0...Max Resp Time] prior to response and will suppress its response if it sees another response to the same group
IGMPv3 adds selective reception from sources within a Group

Membership Query

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

Variants:
- General query: GA=0 and Nrof sources=0.
- Group specific query: GA=/=0, Nrof sources=0
- Group and source specific Query

0x22 = V3 Membership Report

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

---

Experimental routing protocols have been developed for MBone - an overlay MC Internet

<table>
<thead>
<tr>
<th>Shared tree</th>
<th>Source based trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bcast and Prune</td>
<td>Domainwide reports</td>
</tr>
<tr>
<td>PIM Sparse*</td>
<td>DVMRP</td>
</tr>
<tr>
<td>Core Based tree*</td>
<td>PIM Dense*</td>
</tr>
<tr>
<td></td>
<td>MOSPF</td>
</tr>
</tbody>
</table>

* Relies on Unicast routing protocol to locate MC-sources
Those that don’t, can route MC on routes separate from unicast routes.
Distance Vector Multicast Routing Protocol (DVMRP) is used for MC routing in the MBone

DVMRP header

- **Type** = 0x13
- **Code**
- **Checksum**
- **Reserved**
- **Minor vers** = 0xff
- **Major vers** = 3

[IP protocol=2=IGMP, TTL=1, TOS=internwrk cntrl]

- 'DVMRP Probe' [Code=1] for Neighbor discovery
- 'DVMRP Report' [Code=2] for route exchange
- 'DVMRP Prune' [Code=7] - cut a branch of MC tree
- 'DVMRP Graft' [Code=8] (join) - MC tree expansion
- 'DVMRP Graft Ack' [Code=9] - ack of graft reception

DVMRP is similar to RIP except that sources are like destinations in RIP

Probes are used for neighbor discovery

- 'DVMRP Probe' [list of neighbors on this i/f]

  - Probes are exchanged on tunnel and phys. i/fs
  - Mcasts are not exchanged until two-way neighbor relationship is established
  - Routers see each others versions -> compatibility
  - Keepalive --> fault detection, restart detection
    - sent each 10s, timeout set at 35s
  - If list of neighbors is empty - this is leaf ntwrk managed by IGMP

  Unicast [whole DVMRP routing table]
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’ \[\inf < \text{metric} < 2 \times \inf\]
comp. poisonous reverse, \(\inf = 32\).

Reports are processed:

- Adjusted metric = metric + interface cost
- If Metric < inf & Adjust metric ≥ inf
  - Set adjusted metric to inf
- If Route is new and Adj metric < inf
  - Add route to RT
  - Delete prune state of more general route
- Elseif Route exists
  - If Received metric < inf
    - Check if Designated forwarder status for \((S,G)\) changes
      - If Adjusted metric > existing metric
        - From same neighbor: update metric, Sch flash update for route
      - Elseif Adj.metric < existing metric
        - Update metric for the route
      - Elseif sender was different, update RT, schedule flash updates
      - Elseif Adj.metric = existing metric: refresh route ...
      - Elseif Received metric = inf ...
  - Elseif Inf < Received Metric < 2 * inf ...
Multicast algorithm

RPF - reverse path forwarding

Multicast [from=S, to=G]

Upstream interface = u

RPF i/f (S,G)

No

Yes

Mcast to list of i/f's

- At first mcast from RPF i/f a Forwarding Cache Entry [S,G]: (u,list...) is created using the DVMRP routing table
- List contains all downstream routers that have reported dependency on S
- Router is designated forwarder for downstream nodes
- If Designated forwarder becomes unreachable, Router assumes role of designated until it hears from a better candidate

List of dependent neighbors is used to minimise the MC tree

Multicast [from=S, to=G]

Cache = [S,G]: (u,list)

- Initially list may contain all mc i/fs but the upstream i/f
- Downstream address is removed from list if
  - leaf network and G \notin IGMP DB for this phys. network
  - downstream node has selected another designated forw
  - Prune received from all dependent neighbors on this i/f

Prune [S, G, lifetime]

Empty list

Remove Cache Entry
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
- Forw cache entries based on A are flushed
- Graft acks to A are flushed.
- Downstream dependencies are removed. If last, prune sent 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 Forw Cache entries for the Route are flushed
- During hold-down, the route may be taken back, if (<inf and = SAME) metric is received from SAME router
Prunes minimise the Mcast tree

- Prune \([S, (\text{netmask}), G, \text{Lifetime}]\)
- Multicasts ...
- 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
- Prune\([S, (m), G]\)
- 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

- Graft \([S, G]\)
- Graft Ack
- Graft is
  - always acknowledged => if no MCast, nobody is sending
  - if no Ack, is resent with exp. backoff retransmissions
  - forwarded upstream if necessary
Multicast routing example

Source based trees for G1
Shared Multicast tree for G1

![Diagram of Shared Multicast tree for G1]

Mbone overlay is based on WSs running DVMRP

![Diagram of Mbone overlay based on WSs running DVMRP]
MOSPF (Multicast Extensions to OSPF)

- The idea is: if the location of receivers is known to all routers, MC should be possible to exactly the receivers only!
- 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.
- limits the extent of multicast traffic to group members, something e.g. DVRMP cannot always do. Restricting the extent of multicast datagrams is desirable for high-bandwidth multicast applications or limited-bandwidth network links (or both).

MOSPF can be deployed gracefully

- Introduces multicast routing by adding a new type of LSA to the OSPF link-state database and by adding calculations for the paths of multicast datagrams.
- The introduction of MOSPF to an OSPF routing can be gradual - MOSPF will automatically route IP multicast datagrams around those routers incapable of multicast routing, whereas unicast routing continues to function normally.
- 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.
An MOSPF Routing Domain

E.g. $G_1 = 226.1.7.6$
E.g. expanding ring search (TTL).
Group m-LSA created and flooded when
e.g. host on 128.186.4.0 joins $G_1$.

Group-membership-LSA is created and flooded when an
IP user joins an MC-group using IGMP

<table>
<thead>
<tr>
<th>LS Age</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>E-bit. LS Type 6 (group-membership-LSA)</td>
</tr>
<tr>
<td>LS Type</td>
<td>226.1.7.6 (group $G_1$)</td>
</tr>
<tr>
<td>Link State ID</td>
<td>128.186.4.1 (router E)</td>
</tr>
<tr>
<td>Advertising Router</td>
<td>0x80000001</td>
</tr>
<tr>
<td>LS Sequence Number</td>
<td>0x3da9</td>
</tr>
<tr>
<td>LS Checksum</td>
<td>28 bytes</td>
</tr>
<tr>
<td>Length</td>
<td>2 (network)</td>
</tr>
<tr>
<td>Referenced LS Type</td>
<td>128.186.4.1 (128.186.4.0/24)</td>
</tr>
<tr>
<td>Referenced Link State ID</td>
<td>128.186.4.0/24</td>
</tr>
</tbody>
</table>
MOSPF calculates Shortest-path trees on demand

- 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.

Forwarding Cache Entry stores MC path routing info

<table>
<thead>
<tr>
<th>Source network, Group</th>
<th>Router or network for mcast reception</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>List of Interfaces,</td>
</tr>
<tr>
<td></td>
<td>mcasts must be sent</td>
</tr>
<tr>
<td></td>
<td>Metrics to nearest</td>
</tr>
<tr>
<td></td>
<td>group member</td>
</tr>
</tbody>
</table>

A Cache Entry may be deleted at any time -> will be recalculated on demand

- 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 can’t tell which are affected
  - Group-m-LSA: delete entries of that group
  - Hierarchy => the farther away the change is the fewer cache entries are deleted
On demand route calculations use Dijkstra’s SPF-algorithm

- Calculation is rooted on the source not the router as for unicast
- For a new mcast, every router performs the same calculation
- Stub networks do not appear in MOSPF calculation (e.g. router F)
- Tiebreaks for equal cost routes - previous hop router that has highest address is chosen (e.g. G over E)

Two level hierarchy aggregates both sources and group addresses

- In aggregation some info is lost --> sometimes mcasts are sent needlessly: C→G:to G1
- Presence of sources is reported by summary-LSA with MC-bit set: F to H→S3+S4 entry
- Area border router advertise Group-m-LSAs to bbone (B: G1, D,E,F;G1, C,D,E:G2) - no exact location
- Routers in non-bbone do not know location of group members

Wildcard mcast receiver receives all groups
Summary of Multicast Protocols for the Internet

<table>
<thead>
<tr>
<th>Shared tree</th>
<th>Source based trees</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bcast and Prune</td>
</tr>
<tr>
<td></td>
<td>Domainwide reports</td>
</tr>
<tr>
<td>PIM Sparse*</td>
<td>DVMRP</td>
</tr>
<tr>
<td>Core Based tree*</td>
<td>PIM Dense*</td>
</tr>
<tr>
<td></td>
<td>MOSPF</td>
</tr>
</tbody>
</table>

* Rely on Unicast routing protocol to locate MC-sources.
  • Those that don’t, can route MC on routes separate from unicast routes.
  • For Shared tree protocols an additional step of finding the Core or Rendesvous Point must be performed.
  • Directories are useful on service management level.