Components of the IP Multicast Architecture

Internet Group Management Protocol (IGMP)

IGMP Version 1 Message Format

Mapping to Link-layer Multicast Addresses

Link-layer Transmission/Reception
**IGMP Goal**

Designate one router as IGMP “querier”

**Approach**

It asks all hosts to get at least one response per active group, with examples of soft state (periodically query), so occasional losses are okay on each link, one router is elected the “querier”

**How IGMP Works**

- Querier periodically sends a Membership Query message to the all-systems group (224.0.0.1), with TTL = 1 on receipt, hosts start random timers (between 0 and 10 seconds) for each multicast group to which they belong.
- IGMPv2 adds explicit leave messages.
- Query interval is typically 60 -- 90 seconds.
- When a host’s timer for group G expires, it sends a Membership Report to group G, with TTL = 1, other members of G hear the report and stop their timers.
- Routers hear all reports, and time out nonresponding groups.

**IGMP Implications**

- In normal case, only one report message per group present is sent in response to a query (routers need not know who all the members are, only that members exist).
- Query interval is typically 60 -- 90 seconds.
- To reduce join latency, when a host first joins a group, it sends one or two immediate reports (unsolicited responses).

**IGMP Version 2**

- Standard querier election method specified.
- Version and type fields merged into a single field.
- Complexity of election modified
- Backward compatible with version 1
- Changes from version 1:
  - New message and procedures to reduce “leave latency”
  - IGMPv2 adds explicit leave messages.
  - Query interval is typically 60 -- 90 seconds.
  - When a host’s timer for group G expires, it sends a Membership Report to group G, with TTL = 1.
  - Other members of G hear the report and stop their timers.
  - Routers hear all reports, and time out nonresponding groups.

**CS551 Multicast Routing**

Bill Cheng

[http://merlot.unl.edu/cs551-712](http://merlot.unl.edu/cs551-712)
Components of the IP Multicast Architecture

**Hosts**
- IGMP (Internet Group Management Protocol)

**Routers**
- Multicast routing protocols (various)

**Service Model**
- Multicast service model makes it hard to locate receivers

Multicast Routing

**Dynamic Join/Leave**
- Flood data packets to entire network, or

**Options so far (not very efficient)**
- Tell routers about all possible groups and receivers so they can create routes (trees)
- Separate shortest path tree (SPT) for each sender
- Flood and prune
- Rendezvous options

Rendezvous Options

- Specify rendezvous (or meeting place) to which sources send initial packets, and receivers join
- Requires mapping between multicast group address and meeting place

Early Routing Techniques

- prune branches with no receivers
- Link-state multicast protocols
- Compute trees on demand
- Examples: DVMRP, PIM-DM, PIM-SM

Source-based Trees

- Output link determined from input link, multicast address, and source address
- Single tree shared by all members
- Share tree shared by all receivers
- Examples: CBT, PIM-SM

Multicast Tree Taxonomy

- Separate shortest path tree (SPT) for each sender
- Examples: MOSPF, PIM-DM

Source-based trees

- Shared trees
- Shared tree rooted at group core/rendezvous point
- Examples: DVMRP, PIM-DM, PIM-SM

Multicast Routing can build different types of distribution trees

Multicast Routing

- They can create routes (trees)
- Look at routes around the periphery of the network, leaving leaves open
- If a link failure occurs, a new route will be established

Source-based model makes it hard to locate receivers

Service model
A Shared Tree

A Shared Tree

output link determined from input link & multicast address

shortest path trees - low delay, better load distribution

Source-based trees

more state at routers (per-source state)
efficient for dense-area multicast
higher delay (bounded by router OK? traffic concentration)
shortest path trees - low delay, better load distribution
Source-based trees

Who Can Send?

Express (previous)

Only one node can send (others must make their own request)
Single-source

Model used by most multicast applications
Anyone (Deering’s service model)

Multicast Status

but not always stable
Some commercial use (applications)
but not always widely used in research
Multicast widely used on LANs

Who Can Send?

EXPRESS

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?

Who Can Send?
defines IP multicast service model
e.g., best effort, packet based, anonymous groups
Lays foundation for IP multicast

Key Ideas

Characterizing Groups

Cost analysis

Link-state extensions (OSPF)
Distance-vector extensions (DVMRP)
Extended shortest-link
Several algorithms
Enhanced addressing (IPv6)
Dynamic group addressing
Path to multicast service model
Lay's Foundation for Multicast
Phase 2: Prune

Computer Communications - CSCI 551

Copyright © William C. Cheng

Phase 3: Graft

Computer Communications - CSCI 551

Copyright © William C. Cheng

Phase 4: Steady State

Computer Communications - CSCI 551

Copyright © William C. Cheng

Sending Data in DVMRP

Computer Communications - CSCI 551

Copyright © William C. Cheng

DVMRP Pros and Cons

Computer Communications - CSCI 551

Copyright © William C. Cheng

Link-State Multicast Routing

Computer Communications - CSCI 551

Copyright © William C. Cheng

Realized as MOSPF (Multipoint Open Shortest-Path First)
(new like L2A routing)

It is easy to graft a new leaf into the tree

Pros

- Works well with many receivers are passive
- Simple

Cons

- Overhead is per-receiver, receivers are passive
- Works poorly with many groups

Basic idea: treat group members (receivers) as new links

Basic idea: treat group members (receivers) as new links

DVMRP Pros and Cons

Computer Communications - CSCI 551

Copyright © William C. Cheng

Pros

- Works well with many receivers
- Overhead is per-sender
- Receivers are passive

Cons

- Works poorly with many groups
- Every send in every group floods the net
- Every send in every group floods the net
- Misses are expensive if only needed at some places
- If only needed at some places

Realized as MOSPF (Multipoint Open Shortest-Path First)
(new like L2A routing)

Pros

- Works well with many receivers are passive
- Simple

Cons

- Overhead is per-receiver, receivers are passive
- Works poorly with many groups
- Works poorly with many groups
- Misses are expensive if only needed at some places
- Misses are expensive if only needed at some places
Multicast: add membership information to "link state"

Each router computes multicast tree for each active source, builds forwarding entry with outgoing interface list.

Link state: Each router floods link state advertisement

Z computes shortest path tree from S1 to X and Y (lazily, when it gets a data packet for group)

W, Q, R, each do same thing as data arrives at them

Z has network map, including membership at X and Y

Link state advertisement (T) with new membership (R3) may require incremental computation and addition of interface to outgoing interface list (Z)

Overhead: all these inactive nodes must keep multicast states

MOSPF Pros and Cons

Pros

- works well with many senders
- no per-sender state

Cons

- works poorly with many receivers
- no point-to-point
- works poorly with many groups
- lots of information goes places that don't want it
- lots of information goes places that don't want it
- link-state scales with respect to number of links
- many links cause frequent changes
- lots of information goes places that don't want it
- lots of information goes places that don't want it

CS551 Multicast Routing: PIM

[Deering96a]

Bill Cheng
Want a multicast routing protocol that works well with sparse users.

**Key Ideas**

**Computer Communications - CSCI 551**

Copyright © William C. Cheng

Use a single shared tree; fix one host as rendezvous point.

flooding and pruning

With source-based trees senders and receivers meet by:

LS distribution of group and receiver state

How do we solve the problem?

Rendezvous

establish a meeting place: center, core or rendezvous point

trade-off: shared trees can be inefficient

Rendezvous Point (RP)

How to Build A Shared Tree

Basic protocol steps

Routers with local members may initiate data-driven switch to source-specific shortest path trees

PIM Protocol Overview

Incoming interface (iif): interface from which multicast packet is accepted and forwarded

Outgoing interface list (oif list): interfaces out of which multicast packets are forwarded

Rendezvous Point (RP): used in PIM as alternative to broadcast

Designated Router (DR): one router per multi-access LAN elected to track group membership, and then Join/Prune

PIM Terminology

Shared tree:

reverse-shortest-path tree rooted at RP

Source-specific tree:

reverse-shortest-path tree rooted at source. Also referred to as Shortest Path Tree (SPT)

Entry:

multicast forwarding state

Reverse-path forwarding (RPF) check:

checks if a packet arrived on the interface used to reach the source of the packet

source-specific or shared tree

entry: multicast forwarding state for a particular source. Also referred to as source-specific path (SPT)

Shared tree: reverse-shortest-path tree rooted at RP

Soft state: periodic state-driven refreshes, time-out idle state

See PIM v.2 Specification (RFC2362)

Basic protocol steps

Trading off shared trees can be inefficient

Key Ideas

How do we solve the problem?

Source distribution of groups and receiver state:

With source-based trees routers and receivers meet:

Want a multicast routing protocol that works well with sparse shared trees: fix one host as rendezvous point

PIM Terminology (Cont.)

simply send a message towards the RP use the unicast routing table to get there

Quiet easy if you have a RP

Quite easy if you have a RP

Quite easy if you have a RP
Multicast Distribution Tree Example (DVMRP)

Source-specific forwarding entry:
incoming: 1
outgoing: 2, 3

R4 register to be a receiver, not sending

PIM Example: Build Shared Tree

Shared tree after R1, R2, R3 join

Join message toward RP

(*,G) (*,G) (*,G)

Note protocol independence (no DVMRP or MOSPF)

How Do Routers Know RPs?

 RP information is flooded through the network

RP information is flooding something!

Hash based on group address

function to pick a unique RP for the group

If there are multiple RPs, each router uses the same hash

If you are on the tree, you just send it as with other multicast protocols

If you are not on the tree (say, you are a sender but not a group member), the packet is tunneled to the RP that sends it

PIM: Sending Data

this makes central placement of RP important

If you are not on the tree (say, you are a sender but not a group member), the packet is tunneled to the RP that sends it

If you are on the tree, you just send it as with other multicast protocols

Data Encapsulated in Register

S1 unicast encapsulated data packet to RP in Register

RP decapsulates, forwards down shared tree

PIM Example: Sending Data On The Tree

R4 sends data

(*,G) (*,G) (*,G) (*,G) (*,G)

PIM: Sending Data
RP May Ask High-rate Src to Join (Cont...)

Build source-specific distribution tree

Forward Packets on "Longest Match" Entry

Prune S1 off Shared Tree to Avoid Duplicates

Discussion

Interest in multicast motivated by audio and video apps
RP flooding limits scalability

Context

PIM was part of a larger body of work in multicast routing

Forward source-specific tree

Join messages toward S1

Build source-specific distribution tree

Prune S1 off shared tree

where iif of S1 and RP entries differ

Multicast status

Management of multicast is hard

Multicast deployment is ad-hoc

Improvement in multicast scalability compared to DVMRP and MOSPF

Prune if S1 is shared tree

Multicast is an intra-domain routing protocol

S1 Source (S1)-specific distribution tree

Multicast is an intra-domain routing protocol

Shared tree

Prune S1 off shared tree

Interest in multicast motivated by audio and video apps

BGMP & MSDP

Subsequent work developed inter-domain multicast

Management of multicast is hard

Multicast deployment is ad-hoc

Improvement in multicast scalability compared to DVMRP and MOSPF

Reachability

Some source (S) packets may be dropped by any RP (if they can be dropped by any source, they should not be dropped by any RP)

Then may ask high-rate Src to join

Join messages toward S1

Build source-specific distribution tree

Prune S1 off shared tree

where iif of S1 and RP entries differ

Multicast status