PIM: An architecture for wide-area multicast routing

- Stephen Deering (Xerox),
  Deborah Estrin (USC/ISI),
  Dino Farinacci (cisco),
  Van Jacobson (LBL),
  Ching-Gung Liu (USC),
  Liming Wei (USC)

- Ack(ideas): Tony Ballardie, Paul Francis, IDMR WG
- Ack($): cisco, Sun, ARPA, DOE, NSF

- Extended tutorial slides available:
  catarina.usc.edu:~ftp/pub/estrin/pim.tutorial.dec94.ps

Outline

- Motivation for sparse mode multicast
- Overview of sparse mode operation (example)
- High level PIM design summary: Sparse and Dense Mode, Center-based and Shortest path trees, Unicast-routing protocol independent
- Description/Examples of PIM functions
  - Dense Mode
  - Operation over multiaccess LANs: Assert messages
  - Sparse mode over multiaccess LANs
  - Adaptive soft state behavior
  - Multiple RPs
  - PIM/non-PIM
  - Scaling
- Open Issues and Status

Traditional Distance Vector multicast (e.g. DVMRP/Mbone)

- Reverse Path Forwarding (RPF) check on data:
  did it come on shortest path?
- "Flood" toward all possible receivers
- Non-members' routers send "prunes" upstream; stored in S,G state
- Time-out prune state and repeat flood periodically
- Grafts for lower join latency

DV Multicast: Data Packet Distribution and Prune State Overhead OFFtree

Traditional LS Multicast

- Membership information flooded inside LS Advertisements
- Compute and store distribution tree next hops in S,G; data driven
- LS Advertisements flooded periodically
- Tree next hops recomputed if needed

LS Multicast: Membership Information Distribution Overhead

Traditional multicast good for local/campus area

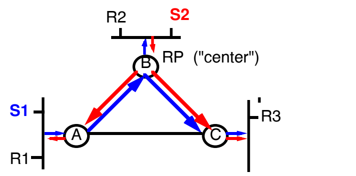
- Flooding is efficient (and simple) means to distribute data or membership when:
  - (A) Overhead of flooding is negligible relative to network resources, OR
  - (B) Most parts of the network have group members
  - Referred to as "Dense Mode" multicast
- (A) becomes less likely as network size and area increases
- For a given group size, (B) becomes less likely as network size increases

"Sparse Mode" multicast for wide-area

- Wide area internets have more "Sparsely" represented groups
  - Sparseness = Total network links/Total mcast tree links
- Scalability Requirement: Limit BW and State OFFtree
  - Avoid flooding (of data or membership info)
- Sparse Mode Approach:
  - Assign per-Group Rendezvous Point(s) (RPs) at which members and senders "meet"
  - Data only sent where requested; routers with downstream members build tree with Explicit Joins toward RP and then toward Sources
  - Explicit Join limits BW and state OFFtree

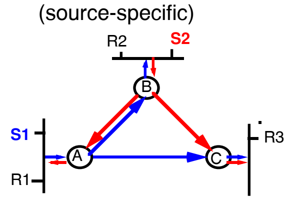
Explicit joining means we can dynamically choose Tree Type

Center/Core-based tree
(shared by sources)



* not min-delay paths (S1 to R3)
* traffic concentration (B to C)
* no Src-specific state

Shortest-path trees
(source-specific)



* min-delay paths
* distributed load (A to C)
* Src-specific tree state

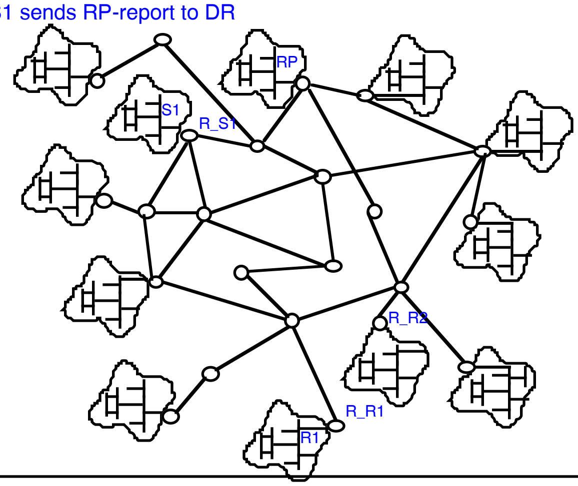
We want both; depends upon traffic characteristics.

Sparse Mode Protocol Operation: Overview

- Receivers' Designated Routers (DRs) join to RP
- Senders' DRs Register with RP
- RP forwards data from new sources and to new members
- Members can join directly to Senders (and prune off RP Tree)
- Data packets travel down S,G and *,G distribution trees
- Lots of details left out...see spec

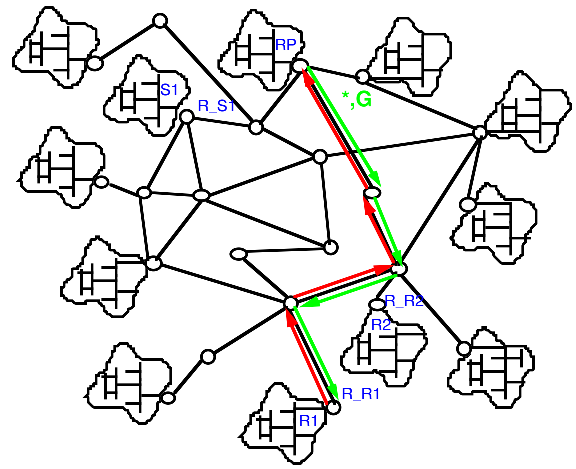
1. R1 sends IGMP-report, RP-report (G:RP-list) to DR
2. S1 sends RP-report to DR

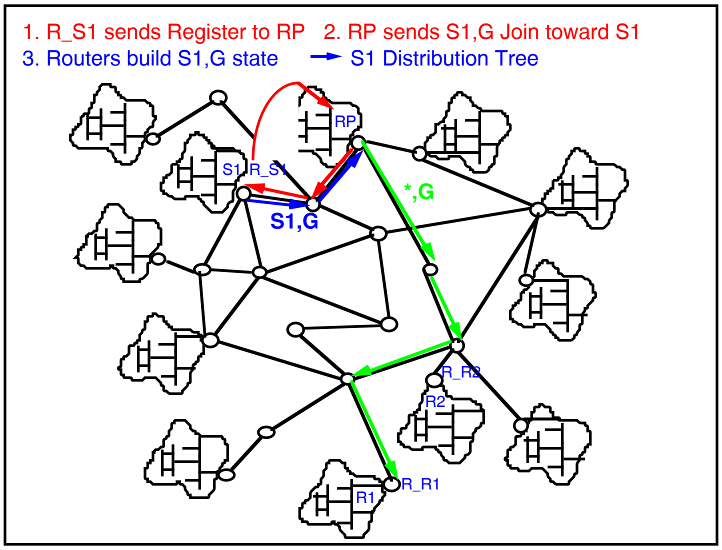
3. G is detectable as RP-based group

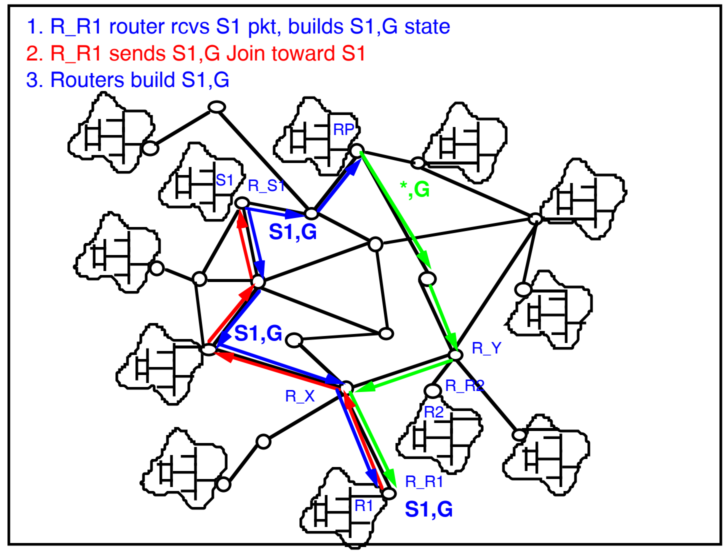


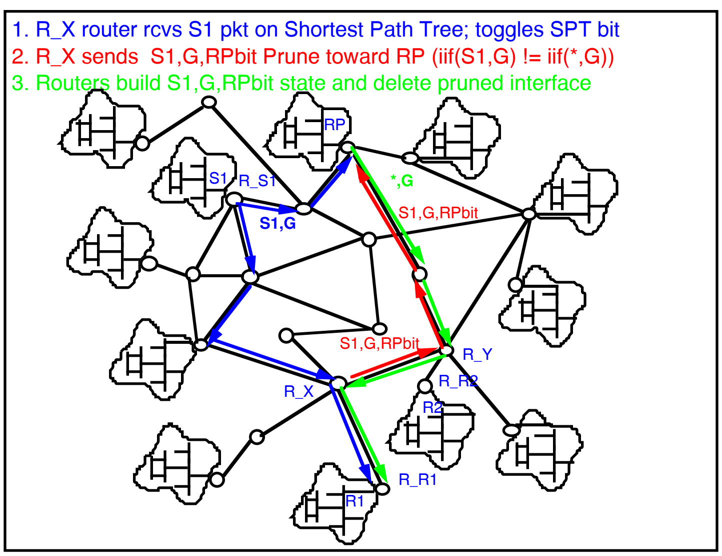
1. R_R1 sends any-source Join toward RP
2. Routers build *,G state

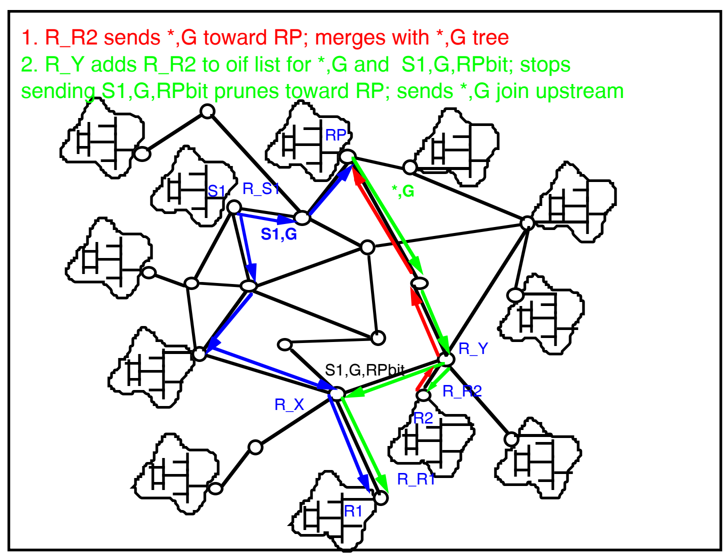
Control Msgs
RP-based Distribution Tree



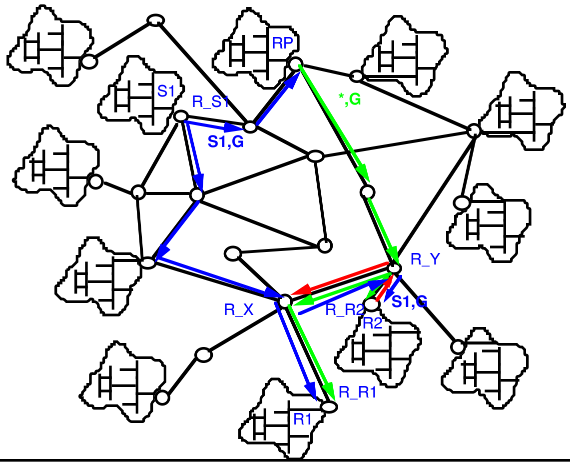

1. R_S1 sends Register to RP
2. RP sends S1,G Join toward S1
3. Routers build S1,G state
S1 Distribution Tree
RP
S1
R_S1
S1,G
*,G
R_R2
R2
R_R1
R1


1. R_R1 router rcvs S1 pkt, builds S1,G state
2. R_R1 sends S1,G Join toward S1
3. Routers build S1,G
RP
S1
R_S1
S1,G
*,G
S1,G
R_Y
R_X
R_R2
R2
R_R1
R1
S1,G

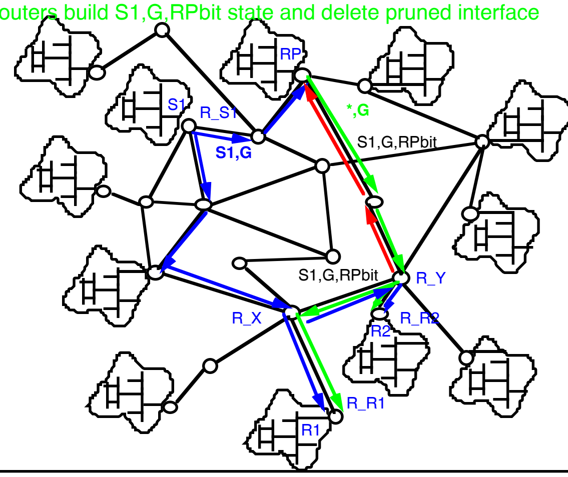

1. R_X router rcvs S1 pkt on Shortest Path Tree; toggles SPT bit
2. R_X sends S1,G,RPbit Prune toward RP (iif(S1,G) != iif(*,G))
3. Routers build S1,G,RPbit state and delete pruned interface
RP
S1
R_S1
S1,G
*,G
S1,G,RPbit
S1,G,RPbit
R_Y
R_X
R_R2
R2
R_R1
R1


1. R_R2 sends *,G toward RP; merges with *,G tree
2. R_Y adds R_R2 to oif list for *,G and S1,G,RPbit; stops sending S1,G,RPbit prunes toward RP; sends *,G join upstream
RP
S1
R_S1
S1,G
*,G
R_Y
S1,G,RPbit
R_X
R_R2
R2
R_R1
R1

1. R_R2 rcvs S1 pkt, builds S1,G
2. R_R2 sends S1,G Join toward S1
3. R_Y resets RPbit and iif in S1,G; sends S1,G Join



1. R_Y rcvs S1 pkt on shortest path tree; toggles SPT
2. R_Y sends S1,G,RPbit Prune toward RP (iif(S1,G) != iif(*,G))
3. Routers build S1,G,RPbit state and delete pruned interface



Dynamic Behavior

- All members hear from New Sources via RP
- New Members hear from all sources via RP
- Last hop router controls switch to shortest path tree (and timeout)
- Periodic refresh of Join/Prune messages capture topology changes and state loss
- S,G state built and kept alive in response to data packets
- *,G state build and kept alive by receivers in anticipation of data

Do we still need "Dense Mode" (Traditional IP style) Multicast ?

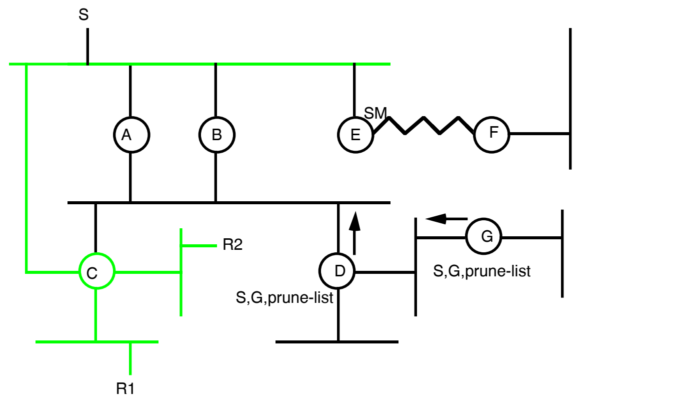
	Dense Mode	Sparse Mode
Join State	None	Explicit
Prune State	Explicit	None
State Created	by data traffic on demand	*,G kept alive by receivers
Protocol Action	Prune S,G	Join RP,G Join S,G Prune S,G

YES: Dense mode is simple and efficient for local area (especially multi-access)

PIM Dense Mode Operation

- Similar to DVMRP
- RPF check, "flood" data
- Prunes sent upstream from leaves; prune state built for S,G
  - Leaf router detection
  - Prune over-rides
  - Asserts to eliminate duplicates over parallel paths
- Explicit (reliable) graft messages and timers undo prune state

PIM Dense Mode Example



* S packets initially flooded over all links
* Routers with no local members and no downstrem neighbors send Prunes upstream
* Steady state packets forwarded only over green links
* Prune state stored offtree

Design Options

- Q: Center-based or Shortest Path Trees?
  - A: Both--depends on traffic characteristics
- Q: Sparse or Dense Mode?
  - A: Both--DM for strictly-local groups; SM for wide-area groups
- Q: DV or LS based routing?
  - A: Either--indifferent to how unicast routing tables are computed

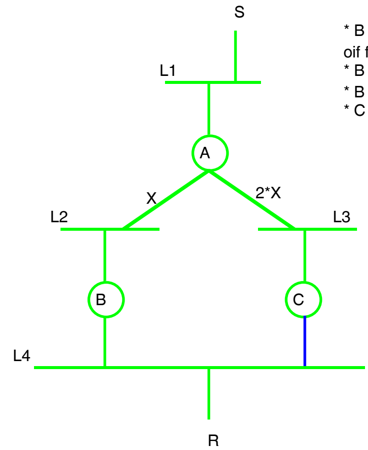
Unicast Routing Protocol Independence

- Can build loop-free multicast using RPF (Reverse Path Forwarding) as basic building block
  - RPF check: did data packet arrive on shortest path ?
  - No other routing information needed to build multicast trees
- RPF information is available from ANY unicast routing protocol's routing TABLE
- Thus the name..."Protocol Independent Multicast" (PIM)--Sparse and Dense Mode with common packet formats and operations

Examples of Detailed PIM Behavior

- Assert Messages in PIM-DM
- PIM SM-DM treatment
- Assert Messages in PIM-SM
- Adaptive soft state mechanism
- Multiple RPs
- PIM-SM interaction with non-PIM protocols
- Scaling technique

Assert Messages in PIM-DM

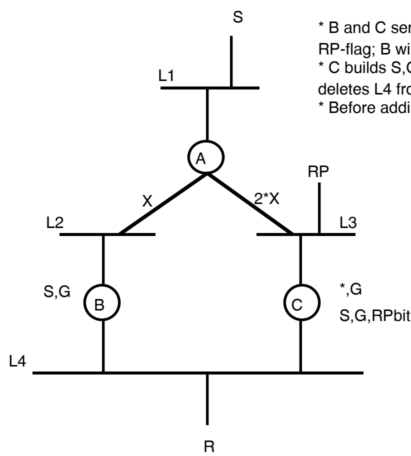


* B and C see pkt from S on L4, their oif for S,G
* B and C each send Assert(S,metric)
* B wins Assert
* C prunes L4 from oif list for S,G

SM-DM

- Router Interfaces configured as Dense or Sparse Mode from forwarder/upstream perspective
  - DM: e.g., resource availability, strictly-local groups
  - SM: e.g., resource limits, wide-area groups
- Wide area groups treated in SM everywhere; independent of interface configuration
- Local area/scoped groups treated in DM over DM interfaces; not sent over SM interfaces without explicit join state.
- Assert messages on LANs resolve overlap in *,G and S,G trees

Assert Messages in PIM-SM



* B and C send Asserts for S,G; C sets RP-flag; B wins Assert
* C builds S,G,RPbit state; copies *,G oif, deletes L4 from oif and flags it
* Before adding back L4, do Assert

Adaptive Soft state mechanism

- Periodic refreshes of all control messages
- Built and triggered based on local state and events
- Maintenance of state is the same as creation of state: change in underlying network requires no special action
- Adaptive to changing topologies
  - When change in unicast route to X
  - Update iif for all X,Gi, and all *,Gi where X is RP

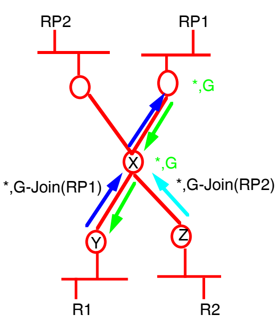
Multiple RPs: Example

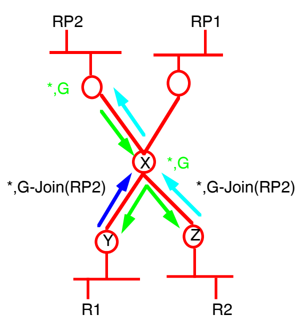
- Each source Registers to each RP
- Each receiver Joins to a single RP
- RP-reachability messages sent downstream by each RP
  - Routers process RP-reachability messages hop-by-hop
  - If RP-reachability matches RP and iif for *,G: update *,G entry timer; forward
  - If RP-reachability has higher value than RP in *,G, overwrite RP in *,G; forward
- Receivers converge to common upstream RP

1. R1 does *,G join to RP1
2. R2 does *,G join to RP2

3. X stores single RP address per *,G entry; chooses highest numbered RP from all *,G-Joins

4. X forwards RP-reachability msgs from RP2 downstream; Y will update RP addr in *,G

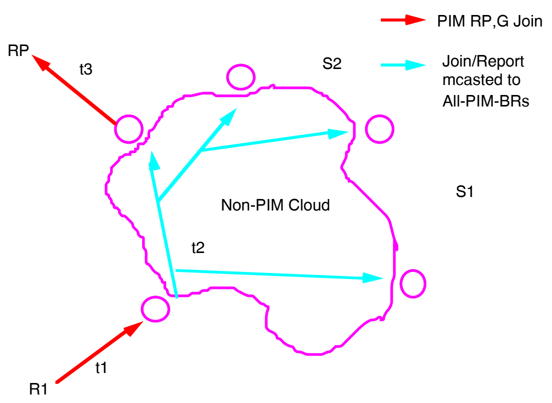




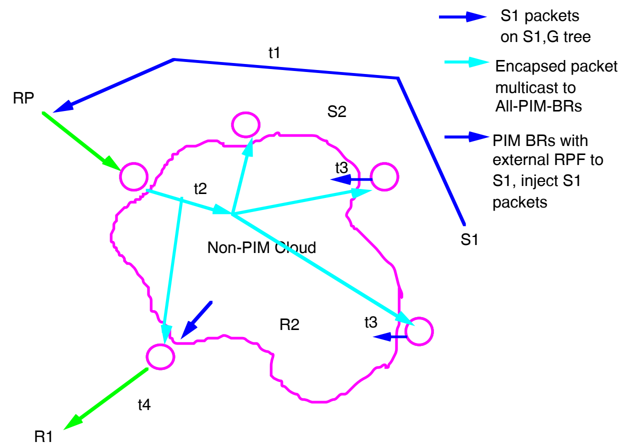
PIM-SM Interaction with DVMRP and MOSPF

- Pull down data to non-PIM cloud
  - *,G Join received by PIM-BRx that reaches RP via NON-PIM cloud
    - PIM-BRx mcasts Join(G,RP) to all-PIM-BRs
    - PIM-BRs with external route to RP send *,G Join toward RP
  - S,G Join received by PIM-BRx that reaches S via NON-PIM cloud
    - PIM-BRx sends S,G Join to all-PIM-BRs
    - Every PIM-BR that has external shortest path to S, sends S,G Join toward S.
- Propagate RP-tree data through non-PIM-DM cloud
  - Receiving PIM-BR multicasts packet to all-PIM-BRs
  - Every PIM-BR that has external shortest path to S, injects packet onto non-PIM links

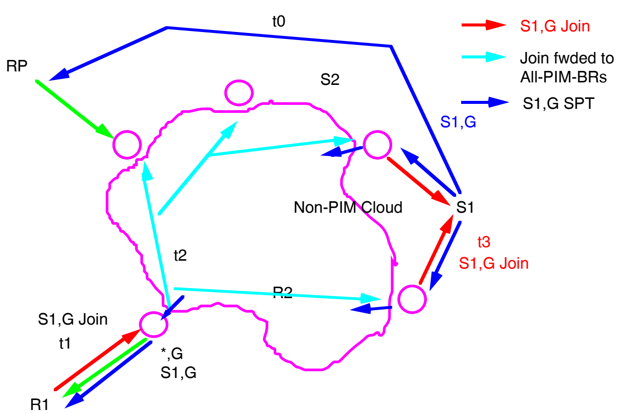
PIM-SM to DVMRP/MOSPF: Joins



PIM-SM to DVMRP/MOSPF: Data Packets



PIM-SM to DVMRP/MOSPF: SPT switch



Scaling

- Control traffic
  - limit to small % of link bw; hbh timers
- State
  - BW-state tradeoff can be made unilaterally
    - Center of net could drop state and build "DM-like", data-driven, S,G entries; initialize oif list with all interfaces (BW overhead)
    - Center must still propagate specific joins to support more specific state (less BW overhead) upstream
    - Center must encapsulate/decapsulate at boundaries
    - RPF check on outer header, Lookup match on inner header.

Open Issues

- Transition architecture
  - Multilevel Multicast
  - Standardize Level1 to Level2 messages
  - Allow heterogeneous Level1 protocols
  - Run a single, common Level2 protocol and make Level1 Boundaries solid
- IGMP host interface under revision
- Scaling mechanisms for state and control messages
- Support for Reservation Protocols and Policy: Explicit Routing to obtain alternate paths

Status

- Original Specification March 94 Internet Draft
- Revised Specification available as Internet Draft
- Architecture Document available as Internet Draft
- cisco Implementation (Dino Farinacci)
- Prototype under development at USC (Charley Liu, Puneet Sharma)
- PIMSIM simulator (Liming Wei)
- Tutorial slides available in catarina.usc.edu:~ftp/pub/estrin/pim.tutorial.dec94.ps