Outline of developments since 3/94 IETF

- IGMP RP Report from hosts to leaf routers: G:RP_list and SPT bit (Dino, yesterday)
- Interface mode bit in Router Query message (Farinacci, yesterday)
- Operation on multi-access networks, e.g. Assert processing (Farinacci, yesterday)
- Management tools (Farinacci, Jacobson, McLoughrie, yesterday)
- PIM SM-DM interaction and integration (Estrin)
- Plans for integrated PIM Spec (Estrin)
- Aggregation of state (Jacobson)
- Open Issues: PIM-non-PIM interaction details, scalable timers
- Implementation status and plans: cisco, USC

PIM SM-DM Interaction: Overview

- Dense Mode goals: store PRUNE state for ACTIVE sources
- Sparse Mode goals: store JOIN state for EXPLICIT receivers

Integration/Interaction goals

- Accommodate strictly-DM behavior for local groups
- Accommodate SM and DM interfaces in a single router
- Accommodate any combination of SM and DM along any multicast tree branch:
    - SM Joins propagate through DM clouds--pull down data from/thru SM
    - Encapsulate packets traveling on RP-tree with Src-Address=RP--flood RP tree data thru DM
- Consistent with aggregation mechanisms

Assumptions

- PIM routers will run combined SM/DM code
- SM-DM interaction can take advantage of modified routers within DM clouds
- Non-PIM clouds will rely only on modified BORDER routers at the cost of some additional overhead (detailed design post IETF)
- Wide-area (non-local) groups have RP(s); all leaf routers obtain G:RP_list mapping.

SM/DM Interfaces

- Each OUTGOING interface CONFIGURED as SM or DM
  - Configure based on resource availability and administrative policy
  - DM if: some strictly-local groups, BW overhead of flood-and-prune OK, minimize non-transient state
- Router SETS corresponding Mode bits in Router Query Messages sent out each OUTGOING interface
- Downstream router FLAGS INCOMING interface as SM or DM according to received Router Query Message.

Processing incoming control messages:

- IGMP Report
  - If G:RPlist mapping exists: create state, add oif, or reset oif timer for *,G
- Join/Prune Message
  - Create state, add oif, or reset oif timer for Sources in Join list
  - Create negative cache state or delete oif for Sources in Prune list w/ RP bit set
  - If Join arrived on incoming interface for existing S,G entry, and sender of Join has higher IP address, supress Join (clear Joiner-Bit)
  - Set Sparse Bit in state according to Sparse Bit in Source Address

State Management: SM vs. DM

- If Join arrived on DM interface, AND RPF interface is DM: can delete S,G state
- If either interface is SM, keep S,G state in order to generate periodic directed joins.

Generating outgoing control messages

- Send Join toward all sources with active S,G state if Joiner and Sparse Bits set.
- Send Prune toward RP when RPF interface is SM AND:
  - Negative cache S,G entry exists (and Joiner bit set)
  - S,Gs have different iif than *,G
- No need to Send prunes up DM interfaces: data packets will be dropped and data-triggered prune sent.

Processing Transit Data Packets

- RPF check always on "outer" header Src_Addr
- If RPF fails, but arriving interface is iif for *,G and IP-Protocol is NOT RP-Encaps: Encapsulate packet with outer header Src_Addr set to RP (found in *,G entry); ELSE DROP
- If RPF checks and IP-Protocol is RP-Encaps: use inner header S,G to lookup or create state
  - If RP-bit cleared, drop packet and send S,G Prune toward RP (indicates unencapsulaetd S,G packets received on SPT); else forward to longest match oif list
- If IP-Protocol is NOT RP-Encaps: use outer header S,G to lookup or create state
- Send triggered Prune when unwanted data packets received on DM interface

Processing Data Packets from/to directly connected hosts

- From directly connected host:
  - If no S,G state, some Register-Stops not received, and G:RPlist exists, encapsulate in Register to RP(s)
- To directly connected members:
  - If longest match is *,G, and SPT bit set: create S,G and send Join toward S; If RPF is the same as *,G, set RP-bit for S,G.

Multi-Access Networks

- Join/Prune messgaes multicasted to all-routers on multi-access networks with target address set to next hop router toward S (or RP)
- Asserts, Prune-overrides as in Dense Mode
- Data packets multicasted to all-routers address
- If local members exist, send decapsulated version of packets ALSO with TTL=1; hosts dont have to decapsulate.

Summary/Evaluation:

- Benefits:
  - DM routers maintain state for active data sources
  - DM routers can exist on Source and RP trees
  - Protocol behavior simple variant on DM/SM
  - RP encapsulation facilitates RP-tree aggregation
- Costs:
  - All data packets require added check in each router on RP tree: one check for RPF, second for forwarding if encapsulated
  - On Multimode trees, Downstream of first DM interface:
    - DM routers generate and carry more control messages: Joins
    - All data packets on RP tree encapsualted
    - Leaf multi-access networks on RP tree carry two copies of each data packet

New Spec: Integrates DM and SM

- State: G,S, iif, RPF IP_Addr, oif, Exp_Timer, Sparse bit, Local_members, RP, Register-stop-all,...
- Timers: Periodic Join, oif, entry, neighbor, RP_entries, Assert, ...
- Message types: Join/Prune, Register, Graft&Ack, Router Query, RP-Reachability, Assert, Register-Stop
- Processing incoming control messages
- Generating outgoing control messages: triggered by timer, data, or control messages
- Data Packet Processing
- Special operations on MultiAccess networks
- Special operations for Leaf Routers
- Pseudocode

Aggregation Overview

- SM eliminates data packet overhead on off-tree routers at the expense of additional state and control message overhead
- Aggregation goals:
  - Avoid state that grows with G*S
  - Allow routers to make unilateral and dynamic tradeoffs between bandwidth and state: high speed backbones different from periphery links.

Aggregation Mechanisms

- Default S,* trees:
  - Router creates S,* entry with union of S,G oif lists
  - Extra data packets sent over non-overlapping links/branches of S,G trees
  - Pass through S,G specific Join/Prune so upstream need not aggregate
- Default RP,* trees:
  - Router creates RP,* entry with union of RP,G oif lists
  - RP encapsulation used on RP-tree anyway
  - Extra data packets sent over non-overlapping links/branches of RP,G trees
  - Pass through RP,G specific Join/Prune so upstream need not aggregate

Aggregation Mechanisms (cont.)

- Dense Mode:
  - Router keeps S,G (entries alive only for active data sources
  - Extra data packets sent over off-tree DM links/branches until pruned
  - May be combined with S,* default trees
- S-Aggregate, G entries
  - Aggregate S,G entries across Sources that belong to common aggregate
  - Cannot send S_Aggregate Joins because of fannout
  - Pass through S,G specific Join/Prune so upstream need not aggregate

Aggregation Open Issues

- Periodic S,G specific control messages still sent through aggregating regions
- Extent of bandwidth overhead of aggregation mechanisms depends on group membership locality.