Protocol Independent Multicast–Sparse Mode (PIM-SM): Motivation and Architecture

Stephen Deering

Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
deering@parc.xerox.com

Deborah Estrin

Computer Science Department/ISI
University of Southern California
Los Angeles, CA 90089
estrin@usc.edu

Dino Farinacci

Cisco Systems Inc.
170 West Tasman Drive,
San Jose, CA 95134
dino@cisco.com

Mark Handley

Department of Computer Science
University College London
Gower Street
London, WC1E 6BT
UK
m.handley@cs.ucl.ac.uk

Ahmed Helmy

Computer Science Department
University of Southern California
Los Angeles, CA 90089
ahelmy@catarina.usc.edu

Van Jacobson

Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, CA 94720
van@ee.lbl.gov

Ching-gung Liu

Computer Science Department
University of Southern California
Los Angeles, CA 90089
charley@catarina.usc.edu

Puneet Sharma

Computer Science Department
University of Southern California
Los Angeles, CA 90089
puneet@catarina.usc.edu

David Thaler

EECS Department
University of Michigan
Ann Arbor, MI 48109
thalerd@eecs.umich.edu

Liming Wei

Cisco Systems Inc.
170 West Tasman Drive,
San Jose, CA 95134
lwei@cisco.com

draft-ietf-idmr-pim-arch-04.ps

October 24, 1996

Status of This Memo

Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

Abstract

Traditional multicast routing mechanisms (e.g. DVMRP and MOSPF [1, 2]) were intended for use within regions where groups are widely represented or bandwidth is universally plentiful. When group members, and senders to those group members, are distributed *sparsely* across a wide area, these schemes are not efficient; data packets or membership report information are periodically sent over many links that do *not* lead to receivers or senders, respectively. This characteristic lead the Internet community to investigate multicast routing architectures that efficiently establish distribution trees across wide-area internets, where many groups are sparsely represented and where bandwidth is not uniformly plentiful due to the distances and multiple administrations traversed. Efficiency is evaluated in terms of the state, control message processing, and data packet processing required across the entire network in order to deliver data packets to the members of the group.

The Protocol Independent Multicast–Sparse Mode (PIM-SM) architecture:

(a) maintains the traditional IP multicast service model of receiver-initiated membership;

(b) uses explicit joins that propagate hop-by-hop from members' directly connected routers toward the distribution tree.

(c) builds a shared multicast distribution tree centered at a Rendezvous Point, and then builds source-specific trees for those sources whose data traffic warrants it.

(d) is not dependent on a specific unicast routing protocol; and

(e) uses soft-state mechanisms to adapt to underlying network conditions and group dynamics.

The robustness, flexibility, and scaling properties of this architecture make it well suited to large heterogeneous inter-networks.

This document motivates and describes the PIM-SM architecture. Companion documents describe the detailed protocol mechanisms for PIM-SM and PIM-DM, respectively [3, 4].

1 Introduction

This document describes an architecture for efficiently routing to multicast groups that may span wide-area (and inter-domain) internets. We refer to the approach as Protocol Independent Multicast-Sparse Mode (PIM-SM) because it is not dependent on any particular unicast routing protocol. Throughout this document we will use the shorter term PIM, to mean PIM-SM. When we are referring to the PIM Dense Mode protocol we will say PIM-DM explicitly.

The most significant innovation in this architecture is the efficient support of sparse, wide area groups. This *sparse mode* (SM) of operation complements the traditional *dense-mode* approach to multicast routing for campus networks, as developed by Deering [5, 6] and implemented in MOSPF and DVMRP [1, 2]. These traditional dense mode multicast schemes were intended for use within regions where a group is widely represented or bandwidth is universally plentiful. However, when group members, and senders to those groups, are distributed *sparsely* across a wide area, these schemes are not efficient; data packets (in the case of DVMRP) or membership report information (in the case of MOSPF) are occasionally sent over many links that do *not* lead to receivers or senders, respectively. The purpose of this work is to develop a multicast routing architecture that efficiently establishes distribution trees even when members are sparsely distributed. Efficiency is evaluated in terms of the state, control message, and data packet overhead required across the entire network in order to deliver data packets to the members of the group.

1.1 Definition of Terms (Glossary)

Following is a list of terms and definitions used throughout this document, in alphabetical order. This is a subset of the glossary list that appears in the protocol specification.

- **Asserts**. The process of choosing a single router to forward multicast packets ¿from a particular source onto a particular LAN segment. The need for Asserts arises when a LAN segment has multiple directly-connected routers with routes to the source.

- **Bootstrap router (BSR)**. A BSR is a dynamically elected router within a PIM domain. It is responsible for constructing the RP-Set and originating Bootstrap messages.

- **Candidate-BSR (C-BSR)**. A C-BSR is a router configured to participate in the BSR election and act as BSRs if elected.

- **Dense-mode (DM)**. A generic term referring to a multicast routing protocol that is optimized for dense groups. DVMRP, MOSPF, and Dense-mode PIM are examples.

- **Designated Router (DR)**. The DR is the highest IP addressed PIM router on a multi-access LAN. Normally, the DR sets up multicast route entries and sends corresponding Join/Prune and Register messages on behalf of directly-connected receivers and sources, respectively. The DR may or may not be the same router as the IGMP Querier. The DR may or may not be the long-term, last-hop router for the group, or a particular source that is sending to the group; a router on the LAN that has a lower metric route to the data source, or to the group's RP, may take over that role.

- **Incoming interface (iif)**. The iif of a multicast route entry indicates the interface from which multicast data packets are accepted for forwarding. The iif is initialized when the entry is created.

- **Join list**. The Join list is one of two lists of IP unicast addresses that is included in a Join/Prune message; each address refers to a source or RP. It indicates those sources or RPs to which downstream receiver(s) wish to join.

- **Last-hop router**. The last-hop router is the router which forwards multicast data packets to directly-connected member hosts. In general the last-hop router is the DR for the LAN. However, under various conditions described in this document a parallel router connected to the same LAN may take over as the last-hop router in place of the DR.

- **Member**. A host that desires to receive multicast datagrams for a group. This host need not be a sender to the group. A Member is synonymously called a *Receiver.*

- **Outgoing interface (oif) list**. Each multicast route entry has an oif list containing the outgoing interfaces to which multicast packets matching that entry should be forwarded.

- **Prune List**. The Prune list is the second list of IP unicast addresses that is included in a Join/Prune message. It indicates those sources or RPs from which downstream receiver(s) wish to prune.

- **PIM Multicast Border Router (PMBR)**. A PMBR connects a PIM domain to other multicast routing domain(s).

- **Rendezvous Point (RP)**. Each multicast group has a shared-tree via which receivers hear of sources. The RP is the root of this per-group shared tree, called the RP-Tree. Candidate-RPs are routers configured to participate as RPs for some (or all) groups.

- **RP-Set**. The BSR for a PIM region constructs a set of RP IP addresses based on Candidate-RP advertisements received. The RP-Set information is distributed to all PIM routers in a domain in a Bootstrap message.

- **Reverse Path Forwarding (RPF)**. RPF is used to select the appropriate incoming interface for a multicast route entry . The RPF neighbor for an IP address X is the the next-hop router used to forward packets toward X. The RPF interface is the interface to that RPF neighbor. In the common case this is the next hop used by the unicast routing protocol for sending unicast packets toward X. For example, in cases where unicast and multicast routes are not congruent, it can be different.

- **Route entry.** A multicast route entry is state maintained in a router along the distribution tree and is created, and updated based on incoming control messages, and in some cases data packets. The route entry may be different from the forwarding entry; the latter is used to forward data packets in real time. Typically a forwarding entry is not created until data packets arrive, the forwarding entry's iif and oif list are copied from the route entry, and the forwarding entry may be flushed and recreated at will.

- **Shared Tree (RP tree)**. The set of paths connecting all receivers of a group to its RP is the RP tree. A receiver on the RP tree receives packets from all sources of the group, except those sources that were pruned off the RP tree.

- **Shortest path tree (SPT)**. The SPT is the multicast distribution tree created by the merger of all of the shortest paths that connect receivers to the source (as determined by unicast routing).

- **Source**. A host that sends multicast datagrams to a group. A Source is not required to be a member. A Source is synonymously called a *Sender.*

- **Sparse Mode (SM)**. Sparse mode PIM uses explicit Join/Prune messages and Rendezvous points in place of Dense Mode PIM's and DVMRP's broadcast and prune mechanism.

- **Wildcard (WC) multicast route entry**. Wildcard multicast route entries are those entries that may be used to forward packets for any source sending to the specified group. Wildcard bits in the join list of a Join/Prune message represent either a (*,G) or (*,*,RP) join; in the prune list they represent a (*,G) prune.
- **(S,G) route entry**. (S,G) is a source-specific route entry. It may be created in response to data packets, Join/Prune messages, or Asserts. The (S,G) state in routers creates a source-rooted, shortest path (or reverse shortest path) distribution tree. (S,G)RPT bit entries are source-specific entries on the shared RP-Tree; these entries are used to prune particular sources off of the shared tree.
- **(*,G) route entry**. Group members join the shared RP-Tree for a particular group. This tree is represented by (*,G) multicast route entries along the shortest path branches between the RP and the group members.
- **(*,*,RP) route entry**. PMBRs join toward all RPs supporting non-local groups, within their PIM domain in order to pull packets generated within the region out to the borders of the region. The routers along the shortest path branches between the RP(s) and the PMBRs keep (*,*,RP) state and use it to determine how to deliver packets toward the PMBRs if data packets arrive for which there is not a longer match.

1.2 Background

In the traditional dense-mode IP multicast model, established by Deering [6], a *multicast address* is assigned to the collection of receivers for a multicast group. Senders simply use that address as the destination address of a packet to reach all members of the group. The separation of senders and receivers allows any host, member or non-member, to send to a group. A group membership protocol (IGMP) [7, 8] is used for routers to learn the existence of members on their directly attached subnetworks. This receiver-initiated join procedure has very good scaling properties; as the group grows, it becomes more likely that a new receiver will be able to splice onto a nearby branch of the distribution tree. A multicast routing protocol, in the form of an extension to existing unicast protocols (e.g. DVMRP, an extension to a RIP-like distance-vector unicast protocol; or MOSPF, an extension to the link-state unicast protocol OSPF), is executed on routers to construct multicast packet delivery paths and to accomplish multicast data packet forwarding.

In the case of link-state protocols, changes of group membership on a subnetwork are detected by one of the routers directly attached to that subnetwork, and that router broadcasts the information to all other routers in the same routing domain [9]. Each router maintains an up-to-date image of the domain's topology through the unicast link-state routing protocol. Upon receiving a multicast data packet, the router uses the topology information and the group membership information to determine the shortest-path tree (SPT) from the packet's source subnetwork to its destination group members. Broadcasting of membership information is one major factor preventing link-state multicast from scaling to larger, wide-area, networks — every router must receive and store membership information for every group in the domain. The other major factor is the processing cost of the Dijkstra shortest-path-tree calculations performed to compute the delivery trees for all active multicast sources [10] for all groups, thus limiting its applicability on an internet-wide basis.

Distance-vector multicast routing protocols construct multicast distribution trees using variants of Reverse Path Forwarding (RPF) [11]. When the first data packet is sent to a group from a particular source subnetwork, and a router receiving this packet has no knowledge about the group, the router forwards the incoming packet out all interfaces except the incoming interface. (Some schemes reduce the number of outgoing interfaces further by using unicast routing protocol information to keep track of child-parent information [6, 2].) A special mechanism is used to avoid forwarding of data packets to leaf subnetworks with no members in that group (also known as truncated broadcasting). Also if the arriving data packet does not come through the interface that the router uses to send packets to the source of the data packet, the data packet is silently dropped; thus the term Reverse Path Forwarding [11]. When a router attached to a leaf subnetwork, receives a data packet addressed to a new group, if it finds no members present on its attached subnetworks, it sends a prune message upstream towards the source of the data packet. The prune messages prune the tree branches not leading to group members, thus resulting in a source-specific shortest-path tree with all leaves having members. Pruned branches will "grow back" after a time-out period; these branches will again be pruned if there are still no multicast members and data packets are still being sent to the group.

Compared with the total number of destinations within the greater internet, the number of destinations having group members of any particular *wide-area* group is likely to be small. More importantly, bandwidth limitations, and therefore data and control message overhead, should not be ignored in a wide area context. In the case of distance-vector multicast schemes, routers that are not on the multicast delivery tree still have to carry the periodic truncated-broadcast of packets, and process the subsequent pruning of branches for all active groups. One particular distance-vector multicast protocol, DVMRP, has been deployed in hundreds of regions connected by the MBONE [12]. However, its occasional broadcasting behavior severely limits its capability to scale to larger networks supporting much larger numbers of groups, many of which are sparse.

1.3 Extending multicast to the wide area: scaling issues

The scalability of a multicast protocol can be evaluated in terms of its overhead growth with the size of the internet, numbers of receivers or sources per group, number of groups, and distribution of group receivers and senders. Overhead is evaluated in terms of resources consumed in routers and links, i.e., state, processing, and bandwidth.

Existing dense-mode link-state and distance-vector multicast routing schemes have good scaling properties only when multicast groups densely populate the network of interest, or when the overhead of dense-mode operation is negligible relative to the network resources. When most of the subnets or links in the (inter)network have group members, then the bandwidth, storage and processing overhead of broadcasting membership reports (link-state), or data packets (distance-vector) is warranted, since the information or data packets are needed in most parts of the network anyway. The emphasis of our work is to develop multicast protocols that will also efficiently support the sparsely distributed groups that are likely to be most prevalent in wide-area, multi-administration, inter-networks where resources must be used more conservatively.

1.4 Overhead and tree types

The examples in Figure 1 illustrate the inadequacies of dense-mode mechanisms when supporting sparse, wide area groups. There are three domains that communicate via an internet. There is a member of a particular group, G, located in each of the domains. There are no other members of this group currently active in the internet. If a traditional IP multicast routing mechanism such as DVMRP is used, then when a source in domain A starts to send to the group, its data packets will be broadcast throughout the entire internet. Subsequently all those sites that do not have local members will send prune messages and the distribution tree will stabilize to that illustrated with bold lines in Figure 1(b). However, periodically, the source's packets will be broadcast throughout the entire internet when the pruned-off branches times out.

Figure 1: Example of Multicast Trees

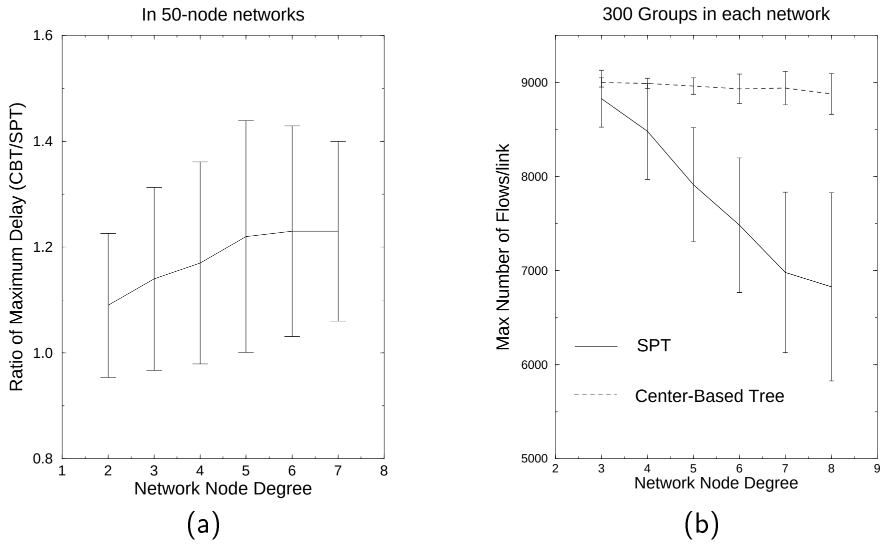

Figure 2: Comparison of shortest-path trees and center-based tree

Thus far we have motivated our design by contrasting it to the traditional dense-mode IP multicast routing protocols. The Core Based Tree (CBT) protocol [13] was proposed to address similar scaling problems in support of sparse-mode multicast. CBT uses a single delivery tree for each group, rooted at one of a small set of "core" routers and shared by all senders to the group. CBT does not exhibit the occasional broadcasting or flooding behavior of earlier protocols. However, CBT does so at the cost of imposing a single shared tree for each multicast group.

If CBT were used to support the example group, then a core might be defined in domain A, and the distribution tree illustrated in Figure 1(c) would be established. This distribution tree would also be used by sources sending from domains B and C. This would result in concentration of all sources' traffic on the path indicated with bold lines. We refer to this as *traffic concentration*. This is a potentially significant issue with any protocol that uses a single shared tree per group. In addition, the packets traveling from Y to Z will not travel via the shortest path used by unicast packets between Y and Z.

We need to know the kind of degradations a core-based tree can incur in average networks. David Wall [14] proved that the bound on maximum delay of an optimal core-based tree (which he called a *center-based* tree) is 2 times the shortest-path delay. To get a better understanding of how well optimal core-based trees perform in average cases, we simulated an optimal core-based tree algorithm over large number of different random graphs. We measured the maximum delay within each group, and experimented with graphs of different node degrees. We show the ratio of the CBT maximum delay versus shortest-path tree maximum delay in Figure 2(a). For each node degree, we tried 500 different 50-node graphs with 10-member groups chosen randomly. It can be seen that the maximum delays of core-based trees with optimal core placement, are up to 1.4 times greater than shortest-path trees. Note that although some error bars in the delay graph extend below 1, there are no real data points below 1 — the distribution is not symmetric, for more details see [15].

For interactive applications where low latency is critical, it is desirable to use the shortest-path trees to avoid the longer delays of an optimal core-based tree.

With respect to the potential traffic concentration problem, we also conducted simulations in randomly generated 50-node networks. In each network, there were 300 active groups all having 40 members, of which 32 members were also senders. We measured the number of traffic flows on each link of the network, then recorded the maximum number within the network. For each node degree between three and eight, 500 random networks were generated, and the measured maximum number of traffic flows were averaged. Figure 2(b) shows a plot of the measurements in networks with different node degrees. This experiment demonstrates situations in which CBT may exhibit significantly greater traffic concentrations.

It is evident to us that both tree types have their advantages and disadvantages. One type of tree may perform very well under one class of conditions, while the other type may be better in other situations. For example, shared trees may perform very well for large numbers of low data rate sources (e.g., resource discovery applications), while SPT(s) may be better suited for high data rate sources (e.g., real time teleconferencing). It would be ideal to flexibly support both types of trees within one multicast architecture, so that the selection of tree types becomes a configuration decision within a multicast protocol. A more complete analysis of these tradeoffs can be found in [15].

PIM is designed to address the two issues addressed above: to avoid the overhead of broadcasting packets when group members sparsely populate the internet, and to do so in a way that supports good-quality distribution trees for heterogeneous applications.

In PIM, a multicast router can choose to use shortest-path trees or a group-shared tree. The last-hop routers of the receivers can make this decision independently. A receiver could even choose different types of trees for different sources. In general, we recommend that routers be configured to join the shortest path tree for a source when the source's data rate exceeds a configured threshold.

The capability to support different tree types is the fundamental difference between PIM and CBT. There are other significant protocol engineering differences as well, the most significant of which is PIM's use of soft state reliability mechanisms. CBT uses explicit hop-by-hop mechanisms to achieve reliable delivery of control messages. As described in the next section, PIM uses periodic refreshes as its primary means of reliability. This approach reduces the complexity of the protocol and covers a wide range of protocol and network failures in a single simple mechanism. Although soft-state refreshing can introduce additional message protocol overhead, we introduce the notion of scalable timers to address such concerns.

1.5 Document organization

In the remainder of this document we enumerate the specific design requirements for wide-area multicast routing (section 2), summarize the architectural components and functions (section 3), enumerate several protocol engineering choices made in the design of PIM protocols (section 4), and consider the use of aggregation to address the scalability problem (section 5). Protocol details can be found in [3].

2 Requirements

We had several design objectives in mind when designing this architecture:

- **Sparse-Mode Regions** We define a sparse mode region as one in which

  (a) the number of networks/domains with group members present is significantly smaller than number of networks/domains in the region as a whole;

  (b) group members span an area that is too large/wide to rely on scope control; and

  (c) the region spanned by the group is not sufficiently resource rich to ignore the overhead of traditional schemes.

  Groups in sparse-mode regions are not necessarily "small"; therefore we must support dynamic groups with large numbers of participants (i.e. receivers and senders).

- **High-Quality Data Distribution**

  We wish to support low-delay data distribution when needed by the application. In particular, we avoid *imposing* a single shared tree in which data packets are forwarded to receivers along a common tree, independent of their source. Source-specific trees are superior when

  (a) multiple sources send data simultaneously and would experience poor service when the traffic is all concentrated on a single shared tree, or

  (b) the path lengths between sources and destinations in the shortest-path tree (SPTs) are significantly shorter than in the shared tree.

- **Routing Protocol Independence**

  The protocol should make use of existing unicast routing functionality to adapt to topology changes, but at the same time be independent of the particular protocol employed. This independence has another advantage that the multicast domain boundaries may extend beyond unicast domain boundaries. This allows network designers to take into consideration the multicast requirements and not to be burdened with unicast topology restrictions. We accomplish this by letting the multicast protocol make use of the unicast routing tables, independent of how those tables are computed.

- **Interoperability with dense mode protocols**

  We require interoperability with traditional RPF and link-state multicast routing, both intra-domain and inter-domain. For example, the intra-domain portion of a distribution tree may be established by some other IP multicast protocol, and the inter-domain portion by PIM; or vice versa. In some cases it will be necessary to impose some additional protocol or configuration overhead in order to interoperate with some intra-domain routing protocols.

- **Robustness**

  The protocol should be able to gracefully adapt to routing changes. We achieve this by

  (a) using *soft state* refreshment mechanisms,

  (b) avoiding a single point of failure by using an RP-Set, and

  (c) adapting along with (and based on) unicast routing changes to deliver multicast service so long as unicast packets are being serviced.

- **Scalability**

  We provide mechanisms for scaling with group and network size. These mechanisms address the forms of overhead: control messages and state. Bandwidth consumed by data packets is already minimized through the use of explicit-join sparse mode. Control message overhead can also be limited to a fixed percentage of the link bandwidth by adjusting the frequency of periodic messages on a link by link basis. This method of controlling overhead was proposed by Van Jacobson.

  State overhead can be managed in such a way that each router can unilaterally choose its own tradeoff point between the amount of state maintained and the amount of bandwidth consumed by unneeded flooding of multicast packets.

3 PIM Components and Functions: Overview

In this section we describe the architectural components of PIM. The detailed protocol mechanisms are described in [3].

As described, traditional multicast routing protocols were optimized for densely distributed groups or uniformly bandwidth-rich regions, and rely on data driven actions in all network routers to establish efficient distribution trees. In contrast, sparse-mode multicast constrains data distribution so that packets reach only routers that are on the path to group members. PIM differs from existing IP multicast schemes in two fundamental ways:

- Routers with local (or downstream) members join a sparse-mode PIM distribution tree by sending explicit Join/Prune messages; in dense-mode IP multicast membership is assumed and multicast data packets are sent until routers without local (or downstream) members send explicit prune messages to remove themselves from the distribution tree.

- Whereas dense-mode IP multicast tree construction is data driven, sparse-mode PIM must use per-group *Rendezvous Point* for receivers to "meet" new sources. Rendezvous Points (RP) are used by senders to announce their existence and by receivers to learn about new senders of a group. In SM, the shared-tree join state is stored in anticipation of data packets, whereas DM does not create state until a data packet arrives. The source-specific trees and associate state are data-driven in PIM, as in PIM-DM.

The shortest-path-tree state maintained in routers is roughly the same type as the multicast routing information that is currently maintained by routers running existing IP multicast protocols such as MOSPF, i.e., source (S), multicast address (G), outgoing interface set (*oif*), incoming interface (*iif*). We refer to this information as the multicast routing entry for (S,G). For all routers containing a (S,G) entry, their *oif*'s and *iif* together form a shortest-path tree rooted at S.

An entry for a shared tree can match packets from any source for its associated group if the packets come through the right incoming interface, we denote such an entry (*,G). A (*,G) entry keeps the same information a (S,G) entry keeps, except that it saves the RP address in place of the source address. There is a wildcard flag (WC-bit) indicating that this is a wild card entry, and an RPT-bit indicating that this is a shared tree entry.

Figure 3 shows a simple scenario of a sender and a receiver joining a multicast group via an RP. When the receiver wants to join a multicast group, its last-hop PIM router (A in fig 3) sends a Join/Prune message towards the RP for the group. If the last-hop router does not have RP information, it is considered an error. Processing of this message by intermediate routers sets up the multicast tree branch from the RP to the receiver. When sources start sending to the multicast group, the designated router (D in fig 3) sends a PIM-Register message, encapsulating the data packet, to the RP for that group. If

Figure 3: How senders rendezvous with receivers

the source's data rate warrants a source-specific tree, the RP responds by sending a Join/Prune message towards the source. Processing of these messages by intermediate routers (there are no intermediate routers between the RP and the source in fig 3) sets up a packet delivery path from the source to the RP.

If source-specific distribution trees are desired (based on the source's data rate or some other configuration parameter), the last-hop PIM router for each member eventually joins the source-rooted distribution tree for each source by sending a Join/Prune message towards the source, including the source in the Join list. After data packets are received on the new path, router B in fig 3 sends a PIM-prune message towards the RP, including the source S in the prune list. B knows, by checking the incoming interface in it routing table, that it is at a point where the shortest-path tree and the RP tree branches diverge. A flag, called SPT-bit, is included in (S,G) entries to indicate whether the transition from shared tree to shortest-path tree has completed. This minimizes the chance of losing data packets during the transition.

Each PIM router must be able to map a multicast group address to that group's RP (an IP address). To do so, an RP-Set is distributed to all PIM routers within a region, and each router runs the same hash function to map from group address to a particular RP in the RP-Set. In this way all routers within a PIM region map a particular group address to the same RP. The RP-Set is constructed and distributed by a dynamically-elected bootstrap router (BSR) within the region. Only a single RP is active for a group at any one point in time, and the BSR is responsible for keeping the RP-Set up to date. Therefore, all candidate RPs within the region send periodic advertisements (liveness indication) to the BSR.

PIM avoids explicit enumeration of receivers. In general, in many existing and anticipated applications, the number of receivers is much larger than the number of sources, and when the number of sources is very large, the average data rate tends to be lower (e.g. resource discovery). In any finite capacity network there is an upper bound on the data rate that any individual host can send or receive. Therefore there are fundamental bounds on the number of high data rate sources that can simultaneously send to the same group. However, there are no such bounds on the number of low datarate sources that can simultaneously send to the same group. If there are very large numbers of sources sending to a group, but the sources' average data rates are low, then it may be more efficient to support the group with a shared tree instead which has less per-source overhead; therefore we suggest triggering Shortest Path Tree (SPT) Join/Prune messages only after the last hop router has received a threshold datarate from the particular source. If sources are low data rate, these Join/Prunes will not be triggered and receivers will receive packets via the shared tree instead and no source specific tree state will be constructed. Issues of group-specific state proliferation and state aggregation are discussed further in section 5.

In summary, data packets from the source will travel to the RP in Register messages, and from the RP will travel to receivers via the distribution paths established by the Join/Prune messages sent upstream from receivers towards the RP. If the RP and receivers initiate shortest path tree Join/Prunes, the sources data packets will longest match on the source specific (S,G) state instead of traveling via the RP distribution tree. Some data packets will continue to travel from the sources to the RP in order to reach new receivers. Similarly, receivers will continue to receive some data packets via the RP tree in order to pick up new senders. However, when source-specific tree distribution is used, most data packets will arrive at receivers over a shortest-path distribution tree. At times when group participation is not changing, and all receivers have joined the shortest path tree(s), the RP can inform source(s) to stop sending data-encapsulating Register messages.

4 Protocol Engineering Design Features

In this section we describe engineering features embodied in the PIM protocols: robustness, interaction with other multicast protocols, and multicast service interfaces.

4.1 Robustness features

There are several areas in which PIM is designed for robustness.

4.1.1 Lost PIM messages

The protocol is fairly robust to lost control messages. If a PIM-Register message gets lost then data packets will continue to be encapsulated in subsequent PIM-Register messages until the first hop router receives a Register-stop message message from the RP. If a new Join/Prune message (carrying join information) is lost over an off-tree link (i.e. a link that is not already part of the mutlicast distribution tree), then for the remainder of the refresh period, packets will not be forwarded on the new path, causing join latency; or in the case of prune information, packets will continue to be forwarded until the refresh is sent, causing leave latency.

All outgoing-interface state that is cached is timed out after a period equal to '3.5' times the refresh period (e.g., default of 210 seconds for the default 60 second refresh interval). As in other multicast routing protocols, this longer timeout interval allows individual packets to be lost without adversely affecting the routing function. When a routing entry has no more outgoing interfaces it is scheduled to be deleted some time later and a prune can be sent upstream (if no prune is sent upstream the upstream state will eventually time out anyway since no Join/Prunes will be received to refresh the join state.) Initially PIM messages are configured to be refreshed every 60 seconds. However, in the future a scalable timer mechanism will be deployed in which the rate is a function of the amount of state in a router and link bandwidth (i.e., for lower speed links the rate will be slower and for higher speed links it may be higher).

4.1.2 Multiple Rendezvous Points and RP failure scenarios

If only a single RP were available to be used for a multicast group, group communication would be disrupted if the RP became unreachable. Assigning a set of available RPs greatly increases the robustness of the system. A small set of PIM routers within a domain are configured to act as Candidate RPs (C-RPs), and periodically send C-RP Advertisements to the elected BSR. At any point in time only a single RP is active for a group. However, when the BSR detects that a particular RP is no longer reachable, the BSR deletes the unreachable RP(s) from the RP-Set next distributed within the periodic Bootstrap message, and all PIM routers within the region rehash affected groups (i.e., those that were previously hashed to the now-unreachable RP).

4.2 Interaction with other multicast protocols

The basic difference between traditional IP multicast routing and PIM is that the former is completely data driven; we will refer to traditional IP multicast routing as "dense mode" for the purposes of this discussion. Four important behavioral differences result:

- Dense mode sends and stores explicit prune state in response to unwanted data packets. Sparse mode requires explicit joining; the default action is to not send data packets where they have not been requested.
- Sparse mode stores shared-tree join state in anticipation of data packets; Dense-mode routers do not store any state until data packets are sent (i.e. for active data sources). The difference is not very significant for active groups (i.e., PIM would have one additional tree active); however for idle groups dense mode has the advantage of having no state at all, whereas PIM would have state for the one shared-tree.
- Sparse mode relies on the concept of an RP for data to be delivered to receivers who request to join the group. Dense-mode groups do not require an RP; broadcast is used as the rendezvous mechanism.
- Sparse mode relies on periodic refreshing of explicit Join/Prune messages. Dense mode does not need to send prune messages periodically because of its data driven nature.

In simplified terms, the cost of dense mode is the default broadcast behavior and maintenance of prune state, whereas the cost of sparse mode is the need for RPs and RP-tree state for idle groups. If all members of a group are located within a bandwidth-rich region, the group may be supported in a strictly dense mode using scope control. However, such groups cannot include any members beyond the indicated scope, without imposing broadcast and prune overhead on the larger scope needed to reach the remote receiver. PIM is designed to address the more general problem of groups that are not a priori limited to intra-domain membership and may therefore span domains.

In the case of multi-access LANs, some interesting issues arise because of possibility of parallel routers forwarding duplicate packets onto the LAN. In SM we must be particularly careful with the operation of the RPtree because the RPF check that prevents routing loops is dependent on information stored in the router, and not based on the source address found in the packet header. As a result it is conceivable that a packet could be routed in elaborate loops because different routers are using different criteria for accepting the packet. To solve this problem each router on a multi-access LAN sends Assert messages when a data packet from a source arrives on the outgoing interface for the associated (S,G) or the (*,G) entry. All routers listen to Assert messages, compare the metrics included therein, and only one router remains the forwarder for that source to that LAN.

We also wish to interoperate with networks that do not have routers modified to generate and interpret PIM Join/Prune messages. We have to address two functions: pulling data out to the dense-mode cloud, and importing data into the PIM region from a dense mode region:

- In PIM, joining a distribution tree is not passive, routers with local members must take explicit join action to receive data packets. This creates problems when a dense-mode region, wishes to interoperate with PIM. To do so, one of two things must happen:

1. Either, PMBR's on the border between PIM and dense mode regions join to all of the PIM region's RPs to pull out all packets generated within the PIM region. Or,
2. The PMBR on the border of a dense mode region must receive some indication of membership within the dense mode cloud, and must generate PIM explicit Join/Prune messages to pull the data down to the dense mode cloud.

  The first of these two approaches is appropriate when the PIM region is a stub or multihomed and is connected to a dense mode backbone. The second of these two approaches is appropriate when the dense mode region is connecting to a PIM backbone.

- The PMBRs at the border between PIM and dense mode regions must act as DRs for the sources external to the PIM-SM domain. In other words the PMBR sets up source specific state and sends Registers on behalf of external sources.

The details of these mechanisms are described in [3, 16, 17].

4.3 Multicast service interface

The multicast interface for hosts is unchanged. Hosts need only learn about and communicate their interest in joining to multicast addresses.

5 Scaling and Aggregation

There are several motivations for aggregating source information; the most important are PIM message size and the amount of memory used for multicast routing entries.

One might consider using the highest level aggregate available for an address when setting up the multicast routing entry. This is optimal with respect to routing entry space. It is also optimal with respect to PIM message size. However, PIM messages will carry very coarse information and when the messages arrive at routers closer to the source(s) where more specific routes exist there will be a large fanout and PIM messages will travel towards all members of the aggregate which would be inefficient in most/many cases.

Traditional IP multicast routing (dense mode) does not have this problem since prune messages can carry most fine grain information which are triggered based on data packets. If the prune messages are lost, subsequent data triggers the prune. On the other hand, graft messages may be subject to the fan-out problem. In this case, they are sent as far as the message information takes it. The penalty is increased join latency.

If PIM is being used for inter-domain routing, and routers were able to map from IP address to domain identifier, then one possibility would be to use the domain level aggregate for a source in PIM messages (Autonomous System (AS) numbers or Routing Domain Identifiers (RDIs)). Then the PIM message would travel to the PMBRs of the domain and the PMBRs can use the internal multicast protocol's mechanism for propagating the join within the domain (e.g. send appropriate link-state advertisement in MOSPF or register a "local member" and do not prune in the case of RPF). However this approach requires that it is both possible and efficient to map from IP to domain address when processing data packets, as well as control packets.

We address the issues of control traffic and state scaling separately below. The detailed mechanisms have not yet been incorporated into the protocol specification as they are still being designed.

5.1 Containing control traffic overhead

To control the bandwidth consumed by periodic control messages, we adopt a technique proposed by one of the authors (Jacobson), called *scalable timers*. The timers controlling periodic refreshing of control messages are set such that the total overhead is a small fixed percentage of the link bandwidth.

Eventually, PIM should use the scalable timer approach; this approach was initially proposed by Van Jacobson and a detailed design and analysis was reported in [18]. In this approach the refresh interval is determined by the sender of the information. The sender can adjust the frequency of control messages (and therefore the timeout period at the control message receiver) depending upon the amount of state that it has to communicate, or refresh, over a particular link. It can thereby keep the amount of control traffic to some small percentage of the link bandwidth. In this case the receiver of the control messages may infer the appropriate refresh interval based on measurement of arriving control traffic, and set its timeout values accordingly.

In the absence of more experimentation with scalable timer mechanisms, the current PIM protocol specifies that the sender of control messages communication hold-time values explicitly. Therefore, a router tells its neighbors how long to keep it reachable by advertising the holdtime in PIM-Hello messages. Likewise, Join/Prune messages indicate how long state should be kept. This allows the sender to change its frequency without the receivers requiring any special configuration information.

5.2 Containing state overhead

PIM-SM maintains less source-specific state than do dense mode protocols. The more important issue faced by all existing multicast routing schemes is how to reduce the amount of group-specific state. This remains an open area of investigation.

6 Conclusions

We have presented a solution to the problem of routing multicast packets in large, wide-area internets. Our approach

(a) uses constrained, receiver-initiated, membership advertisement for sparsely distributed multicast groups;

(b) supports both shared and shortest path tree types in one protocol;

(c) does not depend on a particular unicast protocol; and

(d) uses soft state mechanisms to reliably and responsively maintain multicast trees.

The architecture accommodates graceful and efficient adaptation to varying types of multicast groups, and to different network conditions.

7 Acknowledgments

Tony Ballardie, Scott Brim, Jon Crowcroft, Paul Francis, Lixia Zhang and John Zwiebel provided detailed comments on previous drafts. The authors of CBT and membership of the IDMR WG provided many of the motivating ideas for this work and useful feedback on design details.

References

[1] J. Moy. Multicast extension to ospf. *Internet Draft*, September 1992.

[2] D. Waitzman S. Deering, C. Partridge. Distance vector multicast routing protocol, nov 1988. RFC1075.

[3] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol independent multicast - sparse mode (pim-sm): Protocol specification. *Proposed Experimental RFC*, September 1996.

[4] D. Estrin, D. Farinacci, A. Helmy, V. Jacobson, and L. Wei. Protocol independent multicast - dense mode (pim-dm): Protocol specification. *Proposed Experimental RFC*, September 1996.

[5] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, pages 85–111, May 1990.

[6] S. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, 1991.

[7] S. Deering. Host extensions for ip multicasting, aug 1989. RFC1112.

[8] W. Fenner. Internet group management protocol, version 2. *Internet Draft*, May 1996.

[9] J. Moy. Ospf version 2, oct 1991. RFC1247.

[10] J. Moy. Mospf: Analysis and experience. *Internet Draft*, July 1993.

[11] Y. K. Dalal and R. M. Metcalfe. Reverse path forwarding of broadcast packets. *Communications of the ACM*, 21(12):1040–1048, 1978.

[12] Ron Frederick. Ietf audio & videocast. *Internet Society News*, 1(4):19, 1993.

[13] A. J. Ballardie, P. F. Francis, and J. Crowcroft. Core based trees. In *Proceedings of the ACM SIGCOMM*, San Francisco, 1993.

[14] David Wall. *Mechanisms for Broadcast and Selective Broadcast*. PhD thesis, Stanford University, June 1980. Technical Report N0. 190.

[15] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. In *Proceedings of the 1994 international conference on computer communications and networks*, San Francisco, September 1994.

[16] S. Deering, D. Estrin, D. Farinacci, B. Fenner, V. Jacobson, M. Handley, D. Thaler, L. Wei, and A. Helmy. Interoperability mechanisms for pim-sm and dvmrp. *Internet Draft*, January 1996.

[17] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Pim multicast border router (pmbr) specification for connecting pim-sm domains to a dvmrp backbone. *Internet Draft*, September 1996.

[18] P. Sharma, D. Estrin, S. Floyd, and V. Jacobson. Scalable timers for soft state protocols. *Infocom 97*, June 1996.