Protocol Independent Multicast (PIM): Motivation and Architecture

Stephen Deering (Xerox), Deborah Estrin (USC), Dino Farinacci (Cisco), Van Jacobson (LBL), Ching-gung Liu (USC), Liming Wei (USC)[1]

draft-ietf-idmr-pim-arch-00.ps

March 1994

Abstract

Existing multicast routing mechanisms were intended for use within regions where a group is 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 occasionally sent over many links that do *not* lead to receivers or senders, respectively. We have developed a multicast routing architecture that efficiently establishes distribution trees across wide area internets, where many groups will be sparsely represented. Efficiency is measured 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.

Our Protocol Independent Multicast (PIM) architecture: a) maintains the traditional IP multicast service model of receiver-initiated membership; b) can be configured to adapt to different multicast group and network characteristics; c) is not dependent on a specific unicast routing protocol; and d) 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 internetworks.

This document motivates and describes the PIM architecture. Companion documents describe the protocol mechanisms for both sparse and dense mode PIM[1, 2].

Status of This Memo

[1] Given the length of this author list, it seems appropriate to identify the roles played by each of the authors, who are listed in alphabetical order. Jacobson proposed the original idea of sending join messages toward discovered sources as a means of supporting sparse multicast groups. The detailed architecture and supporting protocols were developed as a collaborative effort of Deering, Estrin, Farinacci, and Jacobson. More recently Liu identified and fixed several critical protocol bugs as part of his implementation effort, and Wei provided data to support the need for shortest path distribution trees (SPT) and contributed to protocol development as part of his simulation effort. Estrin, Liu, and Wei were supported by grants from the National Science Foundation and Sun Microsystems.

1 Introduction

This document describes an architecture for efficiently routing to multicast groups that span wide-area (and inter-domain) internets. We refer to the approach as Protocol Independent Multicast (PIM) because it is not dependent on any particular unicast routing protocol.

The architecture proposed here complements existing multicast routing mechanisms such as those proposed by Deering in [3, 4] and implemented in MOSPF and DVMRP[5, 6]. These traditional 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 group members, 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 some or all members are sparsely distributed. Efficiency is measured 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.

This document motivates and describes the PIM architecture. Companion documents describe the protocol mechanisms for both sparse and dense mode PIM[1, 2].

1.1 Background

In the traditional IP multicast model, established by Deering[4], 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[7] 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[8]. 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[9], 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)[10]. 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.[2] A special mechanism

[2]Some schemes reduce the number of outgoing interfaces further by using unicast routing protocol information to keep track of child-parent information[4, 6].

is used to avoid forwarding of data packets to leaf subnetworks with no members in that group (aka 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[10]. 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 will send 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[11]. 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.

1.2 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, size of groups, number of groups, size of sender sets, and distribution of group members. Overhead is measured in terms of resources consumed in routers and links, i.e., state, processing, and bandwidth.

Existing link-state and distance-vector multicast routing schemes have good scaling properties only when multicast groups densely populate the network of interest. 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 proposed 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 internetworks.

1.3 Overhead and tree types

The examples in Figure 1 illustrate the inadequacies of the existing mechanisms. 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.

Thus far we have motivated our design by contrasting it to the traditional dense-mode IP multicast routing protocols. More recently, the Core Based Tree (CBT) protocol[12] was proposed to address similar scaling problems. CBT uses a single delivery tree for each group, rooted at a "core" router and shared by all senders to the group. As desired for sparse groups, CBT does not exhibit the occasional broadcasting or flooding behavior of earlier protocols. However, CBT does so at the cost of imposing a

Figure 1: Example of Multicast Trees

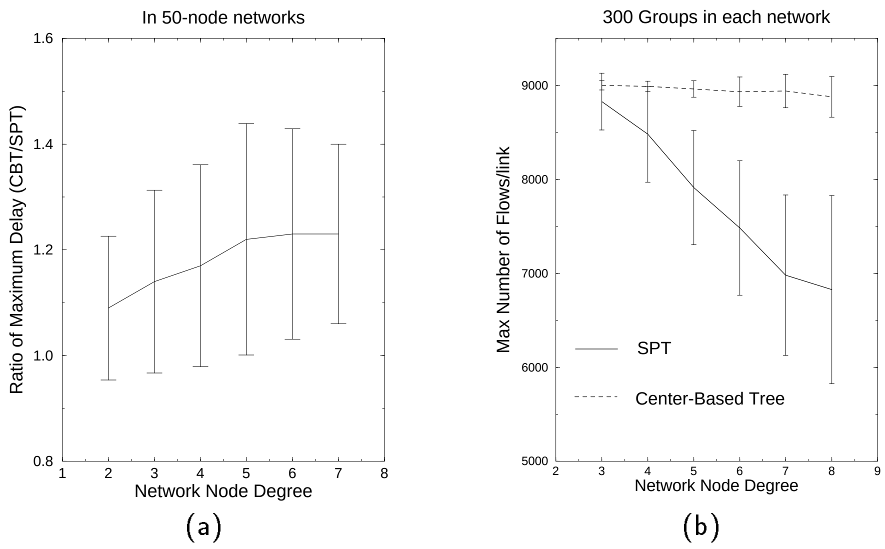

Figure 2: Comparison of shortest-path trees and center-based tree

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 CBT, or any protocol that imposes 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[13] 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 vs 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 around 100% to 140% that of the shortest-path trees[3].

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. It is clear from this experiment that CBT exhibits 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 tress 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).[4] 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.

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 group can choose to use shortest-path trees or a group-shared tree. The first-hop routers of the receivers can make this decision independently. A receiver could even choose different types of trees for different sources.

The capability to support different tree types is the fundamental difference between PIM and CBT. There are other significant protocol engineering differences as well. Two obvious engineering tradeoffs are:

(a) **Soft state vs. explicit reliability mechanism:** 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. On the other hand, it can introduce additional message protocol overhead.

(b) **Incoming interface check on all multicast data packets:** If multicast data packets loop the result can be severe; unlike unicast packets, multicast packets can fan out each time they loop. Therefore we assert that all multicast data packets should be subject to an incoming interface check comparable to the one performed by DVMRP and MOSPF.

1.4 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, and discuss open issues (Section 5). Protocol details can be found in [1].

[3] 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 [14].

[4] A more complete analysis of these tradeoffs can be found in[14].

2 Requirements

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

- **Efficient Sparse Group Support:**

  We define a sparse group as one in which (a) the number of networks/domains with group members present is significantly smaller than number of networks/domains in the Internet, (b) group members span an area that is too large/wide to rely on scope control; and (c) the internetwork spanned by the group is not sufficiently resource rich to ignore the overhead of current schemes. Sparse groups are not necessarily "small"; therefore we must support dynamic groups with large numbers of receivers.

- **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 Independent:**

  The protocol should rely on 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 do not have to map directly to 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.

- **Accommodate Dense Mode Behavior:**

  In those parts of the network where a group is densely represented, we wish to be able to operate in a mode that is more similar to traditional IP multicast, i.e., data driven default to send with explicit pruning.

- **Interoperability:**

  We require interoperability with traditional RPF and link-state multicast routing, both intra- 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. 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.

  In support of this interoperation with existing IP multicast, *and* in support of groups with very large numbers of receivers, we should maintain the logical separation of roles between receivers and senders.

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

Figure 3: How senders rendezvous with receivers

3 PIM Components and Functions: Overview

In this section we describe the architectural components of PIM. The detailed protocols developed to realize these functions are described elsewhere.[1, 2].

As described, traditional multicast routing protocols which were designed for densely populated groups, rely on data driven actions in all network routers to establish efficient distribution trees; we refer to such schemes as *dense mode* multicast. In contrast, *sparse mode* multicast tries to constrain the data distribution so that a minimal number of routers in the network receive it. PIM differs from existing IP multicast schemes in two fundamental ways:

(a) routers with local (or downstream) members join a PIM sparse mode distribution tree by sending explicit join messages; in dense mode IP multicast, such as DVMRP, 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.

(b) whereas dense mode IP multicast tree construction is data driven, PIM must use per-group *Rendezvous Point(s)* 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.

The shortest path tree state maintained in routers is roughly the same as the forwarding 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)[5]. We refer to this forwarding information as the multicast forwarding entry for (S,G).

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 shared tree entry.

Figure 3 shows a simple scenario of a receiver and a sender joining a multicast group via an RP. When the receiver wants to join a PIM multicast group, its first hop PIM router (A in fig 3) sends a PIM-Join message toward one of the RPs advertised for the group. Processing of this message by intermediate routers sets up the multicast tree branch from the RP to the receiver. When sources start sending to

[5] For all routers containing a (S,G) entry, their oif's and iif's together form a shortest path tree rooted at S.

the multicast group, the first hop PIM-router (D in fig 3) sends a PIM-Register message, encapsulating the data packet, to the RP(s) for that group. The RP responds by sending a join toward 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(s).

If source-specific distribution trees are desired, the first hop PIM router for each member eventually joins the source-rooted distribution tree for each source by sending a PIM-Join message towards the source. After data packets are received on the new path, router B in fig 3 sends a PIM-prune message toward the RP[6].

One or more Rendezvous Points (RPs) are used *initially* to propagate data packets from sources to receivers. An RP may be any PIM-speaking router that is close to one of the members of the group, or it may be some other PIM-speaking router in the network. A sparse mode group, i.e., one that the receiver's directly connected PIM router will join using PIM, is identified by the presence of RP address(es) associated with the group in question. The mapping information may be configured or may be learned through another protocol mechanism (e.g., a new IGMP message used by hosts distribute information about RPs to their local routers).

PIM avoids explicit enumeration of receivers, but does require enumeration of sources. If there are very large numbers of sources sending to a group but the sources' average data rates are low, then one possibility is to support the group with a shared tree instead which has less per-source overhead. If shortest path trees are desired then when the number of sources grows very large, some form of aggregation or proxy mechanism will be needed; see section 5. We selected this tradeoff because 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 summary, once the PIM-Join messages have propagated upstream from the RP, data packets from the source will follow the S,G distribution path state established. The packets will travel to the receivers via the distribution paths established by the PIM-Join messages sent upstream from receivers toward the RP. Multicast packets will arrive at some receivers before reaching the RP if the receivers and the source are both "upstream" of the RP. When the receivers initiate shortest-path distribution, additional outgoing interfaces will be added to the S,G entry and the data packets will be delivered via the shortest paths to receivers. Data packets will continue to travel from the source to the RP(s) in order to reach new receivers. Similarly, receivers 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.

4 Protocol Engineering Design Features

In this section we describe engineering features embodied in the PIM protocols: robustness.

4.1 Robustness Features

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

[6]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 finished. of data packets. This minimizes the change of losing data packets during the transition.

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 RP initializes an S,G entry and the associated PIM-Join messages propagate up to the source.

If a PIM-Join message is lost then for the remainder of the refresh period, packets will not be forwarded on the new path, or will continue to be forwarded until the refresh is sent.

For example, PIM messages may be transmitted at a rate of 60 seconds. Information that is cached should be timed out after 3 times the transmission period if no PIM message for the entries have been received. When a forwarding entry has no more outgoing interfaces it is deleted and a prune can be sent upstream (or the router can wait until the next period when the PIM list will no longer include the Source for the deleted entry and the state will eventually be timed out upstream).

4.1.2 Multiple Rendezvous Points and RP failure scenarios

If there is one RP then there is no concern about sources and receivers actually being able to rendezvous, but there is a reliability issue. If there are more than one RPs then each receiver still joins to a single RP, but each source must register to *each* and *every* RP. In other words there are multiple RP distribution trees, and so long as each source sends its packets to all of them, receivers need only join to one.

When the RP fails or becomes unreachable by receivers, members who have already joined will continue to receive packets from sources that had previously sent to the group and for which the receivers had already switched to the SPT (assuming the SPT is not affected by the same failure as makes the RP unreachable). However, new members will send joins toward the unreachable RP and will *not* be successfully joined to the group unless their join packets reach existing SPTs of the sources before they reach the RP. New sources will attempt to register and send to the failed RP. As a result, their packets will not be delivered to any receivers and the SPT from the source to receivers will never be set up even if the paths that make up the SPT are available. This leads to the motivation for employing multiple RPs.

Unreachable RPs are detected using the RP reachability message. When a *,G entry is established by a router with local members, a timer is set. The timer is reset each time an RP reachability message is received. If this timer expires, the router looks up an alternate RP for the group, sends a join toward the new RP. A new *,G entry is established with the incoming interface set to the interface used to reach the new RP. The outgoing interface list includes only those interfaces on which IGMP Reports for the group were received. (Other outgoing interfaces may no longer be valid since the router in question may not be on the shortest path between the downstream branch and the new RP. If the router is on this shortest path as well, it will eventually receive an explicit join from that downstream branch as the last hop routers take the same action).

When multiple RPs are used, each source registers and sends data packets towards each of the RPs, but receivers only join toward a single RP. If one of the RPs fails, receivers that joined to that RP will stop receiving RP-reachability messages and will start sending joins to one of the alternative RPs. Sources do not need to take special action. When an RP is unreachable it will not receive the source's register messages and therefore will not respond with joins and so the outgoing interfaces in *,G pointing toward the unreachable RP will time out; without any explicit action on the part of the source.

Because each receiver's directly connected router selects an RP independently, it is possible for routers on the same part of the distribution tree to specify different RPs while both are still available. This can lead to looping in some topologies. To avoid looping, RP address information carried in PIM-Join and RP-reachability messages is examined to converge to a common RP (the larger numbered RP dominates).

4.1.3 Unicast routing changes

When unicast routing changes an RPF check is done and all affected expected incoming interfaces are updated. If the new incoming interface appears in the outgoing interface list, it is deleted from the outgoing list. The previous incoming interface may be added to the outgoing interface list by a subsequent join from downstream. Joins received on the current incoming interface are ignored. Joins received on new interfaces or existing outgoing interfaces are not ignored. Other outgoing interfaces are left as is until they are explicitly pruned by downstream routers or are timed out due to lack of appropriate join messages.

4.2 Control Message Overhead Reduction

[7] One potential drawback of PIM-Sparse Mode is the unnecessary overhead associated with periodic join messages when a group is locally dense with respect to individual routers. PIM-SM provides the following mechanism to address this problem.

If a router finds that most of its outgoing interfaces are in the outgoing interface lists for a particular group, the router can tell its downstream routers to switch into *Implicit Join mode* and eliminate the periodic join traffic that is normally sent in Sparse Mode. The router can switch back into *Periodic Join mode* by sending mode messages with the appropriate flag.

4.3 Relationship between Dense Mode and Sparse Mode PIM

There are three differences between PIM SM and PIM DM:

1. Dense mode is data driven and uses explicit pruning. Sparse mode requires explicit joining; the default action is to not send data packets where they have not been requested.

2. Because it is not data driven, 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.

3. Because it is not data drive, Sparse Mode relies on periodic refreshing of explicit join 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 flooding behavior, whereas the cost of sparse mode is the need for RPs and periodic refreshes. We accommodate the use of both SM and DM within the same router, and even within the same group to avoid imposing a single tradeoff among these costs.

Some links will have cost functions that prohibit dense mode operation to avoid the default flooding behavior. Other links may be treated in dense mode if there is no RP information. For example, an expensive WAN link or inter-domain link *may* be configured as default-sparse-mode if it is never appropriate to default to dense mode behavior. In contrast, most intra-domain or intra-campus links will probably be configured as Dense Mode Acceptable. The dense-mode configuration of an interface does not affect the behavior of the PIM-SM protocol, per se. It does affect the treatment of data packets sent to groups for which the router has no state. If an interface is configured as Dense Mode Acceptable, then that interface will be included in the new S,G entry built when a data packet to G from S arrives and there is no local state. If the interface is not so configured, it will not be included in such entries unless and until an explicit join is received from downstream.

If all members of a group are located within a region whose router interfaces are all configured as Dense Mode Acceptable, then the group may be supported in a strictly dense mode. These groups require

[7] Editors Note: This section is rather new and better mechanisms might be found than some of those employed here; we have not had a chance to go through them thoroughly and consider all the alternatives. In other words this particular subsection is still only half baked !

no RP to be configured or used, and shortest path trees are built in a data driven manner. Efficient and simple multicast routing support for this type of group is important, and for that reason we have defined the Protocol Independent Multicast-Dense Mode protocol[2] to provide DVMRP style multicast support in a unicast-routing protocol independent manner. However, such groups that do not make use of RPs will *not* be able to include any receivers that must be reached via a sparse mode interface.

PIM-SM is designed to address the more general problem of groups that are not a priori limited to intra-domain membership and must therefore span Sparse Mode interfaces and boundaries. Any such group that is not strictly local to a dense mode configured domain must have at least one RP defined. All receivers join such inter-domain groups using periodic explicit join messages defined in the Sparse Mode protocol.

After the initial joining, if any router on the distribution tree observes that the group is densely represented locally and that it would be efficient to eliminate the periodic joins, it can switch to the implicit join mode of sparse mode. This allows receivers within a "densely populated region" to maintain their portion of the tree in a style more similar to dense mode. However, in exchange for being able to reach inter-domain participants, this approach does not insulate the receivers within these densely populated parts of the tree from having to access and use RP information.

4.4 Interoperation with non-PIM networks

We wish to interoperate with networks that do not have hosts and routers modified to generate and interpret PIM-Join messages.

In PIM-SM, receivers are not passive, they must take explicit join action to receive data packets. This creates problems when a non-PIM, dense mode region, wishes to interoperate with PIM-SM. In particular, when a receiver decides to join a group inside of a non-PIM cloud in which there are no other members then the Border Routers (BRs) of that cloud must be notified in order to trigger sending of an PIM-Join message toward the RP. In the case of MOSPF new group membership is advertised to backbone routers but not necessarily to all Area Border Routers (ABRs). In the case of DVMRP and most other distance vector IGPs membership is not advertised at all. Therefore in both cases, some additional mechanism is needed.

We can solve this problem in a manner similar to the multi-access LAN case. All BRs that speak PIM join a special multicast group. Members of this group conduct a Designated BR (DBR) election among themselves. The resulting DBR sends an IGMP-query to the a multicast group consisting of all multicast routers in the domain. Members of this group respond by sending IGMP-report messages to the group; members of group also listen to these reports and suppress sending reports for groups that have been reported by other routers.

As a result, all PIM-BRs hear of all groups for which internal members exist. Based on this information, and information obtained from IBGP or OSPF, the BRs can determine which of them should send an PIM-Join message to the RP for each group for which there is a local member. Note that DBRs are source, group specific.

If the Border routers are running IBGP then they have the information necessary to determine which BR should include a particular host in its join list. Similarly if the BRs are running OSPF then the information can be computed. However, if the domain is running DVMRP or some other scheme, there may need to be some additional mechanism employed in the BRs. This is an open issue still to be resolved in order to achieve maximum interoperation with existing networks.

4.5 Multicast Service Interface

In the general case, Sparse mode PIM requires that receivers obtain the address(es) of an RP, along with the address of the multicast group. Receivers then need to communicate these values to their leaf routers, just as receivers have to communicate multicast addresses currently. In PIM, sources will also have to provide the multicast and RP address information to their leaf routers.

For special cases routers can use configured, well-known-value, or default RP information to avoid the necessity for this additional information from hosts; however, these solutions are not general enough.

Although it is always better to avoid changes in the service model if possible, in this case, the change is quite minor in that it is not implicating an additional information distribution mechanism. In other words, the host does not need to interact with some new directory service or number distribution/advertisement mechanism. Rather, the host just needs to obtain more than one number from whomever the multicast address is currently obtained.

Note that strictly local (e.g. intra-domain) groups do not require RPs, even for PIM dense mode. Deering has suggested defining a separate part of the address space for multicast groups that do require RP (SM) joining to avoid the ambiguity that might occur if the binding between RP and multicast address information is lost.

5 Open issues

Before concluding we discuss several open issues that require further research, engineering, or experimental attention.

- **Aggregation of information in PIM:**

  There are several motivations for aggregating source information beyond the subnet level supported in the current specification; the most important are PIM message size and the amount of memory used for multicast routing forwarding entries.

  One might consider using the highest level aggregate available for an address when setting up the multicast forwarding entry. This is optimal with respect to forwarding 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 toward all members of the aggregate which would be inefficient in most/many cases.

  If PIM is being used for inter-domain routing, and routers are able to map from IP address to domain identifier, then one possibility is 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 will travel to the border router(s) (BR) of the domain and the BRs 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.

  Another possibility is to use proxies as suggested by V. Jacobson. In this case within PIM clouds, PIM messages need only refer to proxies for sources outside the cloud. In this scheme BRs would join a PIM tree externally and inject themselves as sources internally. When data packets arrived, the data packet would be forwarded into the cloud and routers would see a new source. They would then need to determine which is the entry BR for the particular source and forward the packet on the multicast tree associated with that BR. The router could cache a forwarding entry for the new source in order to avoid repeating this step on each data packet. This technique is currently being developed and would be deployable as an addition to the current protocol without affecting the protocol specification per se.

In the absence of aggregation or proxy techniques, when the number of sources get to some threshold value (to be determined), receivers could compromise the quality of the distribution tree in exchange for accommodating large numbers of unaggregatable sources.

As the number of groups grows very large, it may be necessary to allow aggregation of state across groups; whereas thus far we have only addressed aggregation across sources. Two of the authors (Deering and Jacobson) have proposed creating default S,* entries to address this problem.

- **Selecting RPs:**

  An RP for a particular multicast group can be any IP-addressable entity in the internet. However, it is efficient and convenient for the RP to be the directly-connected PIM router of one of the members of the group. If an RP has local members of the group then there is no wasted overhead associated with sources continually sending their data packets to the RP since it needed to be delivered there anyway for delivery to those members. Nevertheless, we need not be overly concerned with placement of the RPs when shortest path trees are used because the RP will not remain on the distribution path for most receivers, unless it happens to also be on the SPT.

  As described earlier, the RP address can be configured or can be dynamically discovered by mapping from the multicast address, query of a directory service, or from information obtained via some new PIM-RP-Report messages. The mapping of G to RP addresses should be cached.

- **Interaction with policy-based and TOS routing:**

  PIM messages and data packets may travel over policy-constrained routes to the same extent that unicast routing does, so long as the policy does not prohibit this traffic explicitly.

  To obtain policy-sensitive distribution of multicast packets we need to consider the paths chosen for forwarding PIM-Join and Register messages.

  If the path to reach the RP or some source is indicated as being the appropriate QOS and indicated as being symmetric then PIM routers can determine that if they forward joins upstream that the data packets will allowed to travel downstream. This implies that BGP/IDRP[15, 16] should carry two QOS flags: symmetry flag and multicast willing flag.

  If the generic route computed by hop-by-hop routing does not have the symmetry and multicast bits set, but there is an SDRP[17] route that does, then the PIM message should be sent with an embedded SDRP route. This option needs to be added to PIM join messages. Its absence will indicate forwarding according to the router's unicast routing tables. Its presence will indicate forwarding according to the SDRP route. This implies that SDRP should also carry symmetry and multicast QOS bits *and* that PIM should carry an optional SDRP route inside of it to cause the PIM message and the multicast forwarding state to occur on an alternative distribution tree branch.

- **Interaction with Receiver Initiated reservation setup such as RSVP[18]:**

  Once the shortest path distribution tree has been established RSVP reservation messages follow the reverse of senders path messages and the senders path messages will travel according to the state that PIM installs. However, one wants to avoid switching reservation-oriented routes so the receiver could initially receive all packets via the RP distribution tree and after some delay it could send PIM messages to establish the shortest path tree and then establish reservations over that tree. The source's path message would travel first via the RP path, then to avoid setting up a reservation on the RP path, the receiver would send its IGMP message *before* it sends out its reservation message and wait for another path message to travel over the new shortest path.

In summary we expect that this receiver initiated routing is well suited to receiver initiated reservations since if a reservation is blocked the previous router or the receiver can select an alternative reverse path to the particular source(s). This is also a subject for future work that will affect the use of the protocol, and not the protocol itself.

6 Conclusions

We have presented a solution to the problem of routing multicast packets in large, wide area internets. Our approach (1) uses constrained, receiver-initiated, membership advertisement for sparsely distributed multicast groups; (2) supports both shared and shortest path tree types in one protocol; (3) does not depend on the underlying unicast protocols; and (4) 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.

Due to the complexity of the environments PIM expects to operate in, there are still several issues not completely resolved. Solutions to some of the issues require coordination with efforts in other areas such as interdomain routing and resource reservation protocols.

7 Acknowledgments

Tony Ballardie, Scott Brim, Jon Crowcroft, Paul Francis, and Lixia Zhang 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] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. Protocol independent multicast (pim), sparse mode protocol : Specification. *Internet Draft*, March 1994.

[2] S. Deering, D. Estrin, D. Farinacci, and V. Jacobson. Protocol independent multicast (pim), dense mode protocol : Specification. *Internet Draft*, March 1994.

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

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

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

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

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

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

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

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

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

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

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

[14] L. Wei and D. Estrin. A comparison of multicast trees and algorithms. Technical Report USC-CS-93-560, Computer Science Department, University of Southern California, September 1993.

[15] Y. Rekhter and T. Li , editors. A border gateway protocol 4 (bgp-4). *Internet Draft*, January 1994.

[16] S. Hares and John Scudder. Idrp for ip. *Internet Draft*, September 1993.

[17] D. Estrin, T. Li, Y. Rekhter, and D. Zappala. Source demand routing protocol: Packet format and forwarding specification. *Internet-Draft*, March 1993.

[18] L. Zhang, R. Braden, D. Estrin, S. Herzog, and S Jamin. Resource reservation protocol (rsvp) – version 1 functional specification. *Internet-Draft*, October 1993.