An Architecture for Wide-Area Multicast Routing

Stephen Deering (Xerox), Deborah Estrin (USC), Dino Farinacci (cisco), Van Jacobson (LBL),

Ching-Gung Liu (USC), Liming Wei (USC)[0]

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 inter-networks.

1 Introduction

This paper 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 [1, 2] and implemented in MOSPF and DVMRP [3, 4]. 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.

1.1 Background

In the traditional IP multicast model, established by Deering [2], 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 [5] 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 [6]. 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

[0] 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 mean 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.

Figure 1: Example of Multicast Trees

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 [7], thus limiting its applicability on an internet wide basis.

Distance-vector multicast routing protocols construct multicast distribution trees using variants of Reverse Path Forwarding [8]. 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 [1]. A special mechanism 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 (RPF) [8]. 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 toward 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. 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 [9]. 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.

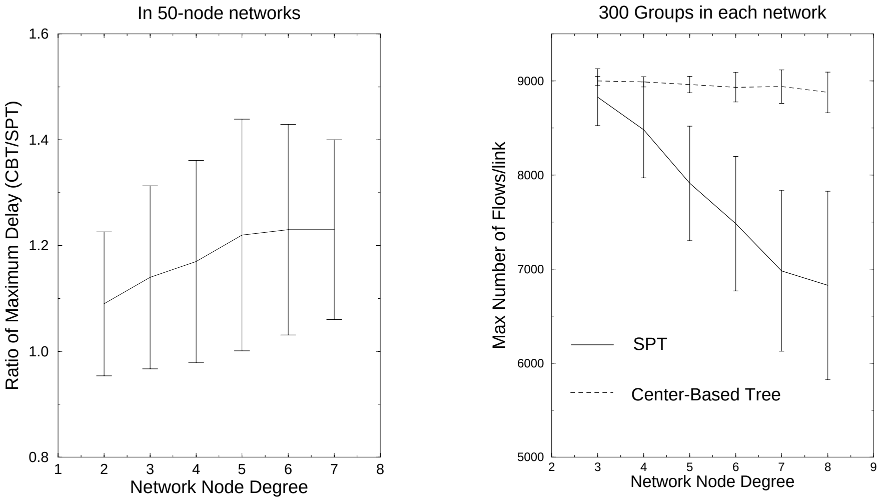
Existing link-state and distance-vector multicast routing schemes have good scaling properties only when multicast groups densely populate the network of interest. When most of the subnetworks or links in the (inter)network have group members, then the bandwidth, storage and processing overhead of broadcasting 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 inter-networks.

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 time out.

Thus far we have motivated our design by contrasting it to the traditional densely-distributed-membership IP multicast routing protocols. More recently, the Core Based Tree (CBT) protocol [10] was proposed to address similar scaling

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



(a) (b)

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

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 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 the 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 [11] 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 of the shortest-path trees [2].

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 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) [3]. 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 described 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. B

The capability to support different tree types is the fundamental difference between PIM and CBT. There are other significant protocol engineering differences as well [4].

[2] 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 [12].

[3] A more complete analysis of these tradeoffs can be found in [12].

[4] Two obvious engineering tradeoffs are:

1. **Soft state versus explicit reliability mechanism:** CBT uses explicit hop-by-hop mechanisms to achieve reliable delivery of control messages. As described in the next section, PIM

1.4 Paper organization

In the remainder of this paper we enumerate the specific design requirements for wide-area multicast routing (section 2), describe a specific protocol for realizing these requirements (sections 3), and discuss open issues (section 4).

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 the 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 inter-network 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. We accomplish this by letting the multicast protocol make use of the unicast routing tables, independent of how those tables are computed.

- **Interoperability:**

  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. 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.

3 PIM Protocol

In this section we start with an overview of the PIM protocol and then give a more detailed description of each phase.

As described, traditional multicast routing protocols which were designed for densely populated groups, rely on data driven actions in all the 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:

1. 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.

2. whereas dense mode IP multicast tree construction is data driven, PIM must use per-group *rendezvous point(s)* (RPs) for receivers to "meet" new sources. RPs are used by senders to announce their existence and by receivers to learn about new senders of a group [5].

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) [6]. 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 an (S,G) entry keeps, except that it saves the RP address in place of the source address. There is a wildcard flag 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

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.

2. **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.

[5] We will discuss how RPs are selected in section 4

[6] The oif's and iif's of (S,G) entries in all routers together form a shortest path tree rooted at S.

Figure 3: How senders rendezvous with receivers

PIM-speaking router (*A* in figure 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 the multicast group, the first-hop PIM-speaking router (D in figure 3) sends a PIM register message, piggybacked on 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 figure 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 toward the source. After data packets are received on the new path (router D to router B, then to router A), router B in figure 3 sends a PIM prune message toward the RP [7].

One or more rendezvous points (RPs) are used *initially* to propagate data packets from sources to receivers. An RP can be any 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 4. 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).

The remainder of this section describes the protocol design in more detail.

3.1 Local hosts joining a group

A host sends IGMP report message identifying a particular group, G, in response to a directly-connected router's IGMP query message, as shown in figure 4. From this point on we refer to such a host as a receiver, R, (or member) of the group G.

When a *designated router* (DR) [8] [5] receives a report for a new group G, it checks to see if it has RP address(es) associated with G [9]. A DR will identify a new group (i.e., one for which it has no existing multicast entries) as needing PIM sparse mode support by checking if there exists an RP mapping. If there is no RP mapping provided in IGMP report messages, and there is no mapping provided in the appropriate configuration file, then the router will assume that the group is *not* to be supported with PIM sparse mode. Even when a group has an associated RP, it may be that some outgoing and incoming interfaces do not require PIM sparse mode, but are handled using a dense mode scheme such as MOSPF, DVMRP or a dense mode variant of PIM [13]. In this case the router will flag individual interfaces as dense or sparse mode, to allow differential treatment of different interfaces. For the sake of clarity, we will ignore these added complexities throughout most of the protocol description. See section 4 for some further discussion of

[7] 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. This minimizes the chance of losing data packets during the transition.

[8] A designated router is the one that takes responsibility for serving the members on a multi-access LAN.

[9] The mechanism for learning this mapping of G to RP(s) is somewhat orthogonal to the specification of this protocol; however, we require some mechanism in order for the protocol to work. At the very least this information must be manually configurable. We propose the use of a new host message that would allow hosts to inform their directly-connected PIM-speaking routers of G,RP(s) mappings. This is important for dynamic groups where hosts participate in special applications to advertise and learn of multicast addresses and their associated RP(s).

Figure 4: Example: how a receiver joins, and sets up shared tree
Actions are numbered in the order they occur

these very practical issues.

For the remainder of this description we will also assume a single RP just for the sake of clarity. We discuss the direct extensibility to operation with multiple RPs later in the document in section 3.9.

The DR (e.g., router A in figure 4) creates a multicast forwarding cache for (*,G) . The RP address is included in a special record in the forwarding entry, so that it will be included in upstream join messages. The outgoing interface is set to that over which the IGMP report was received from the new member. The incoming interface is set to the interface used to send unicast packets to the RP. A wildcard (WC) bit associated with this entry is set, indicating that this is a (*,G) entry.

The DR sets an RP-timer for this entry. The timer is reset each time an RP reachability message is received for (*,G) (see section 3.2).

3.2 Establishing the RP-rooted shared tree

The DR router creates a PIM join message with the RP address in its join list with the RP and WC bits set; nothing is listed in its prune list. The WC bit flags an address as being the RP associated with that shared tree. The RP bit indicates that the receiver expects to receive packets from new sources via this (shared tree) path and therefore upstream routers should create or add to (*,G) forwarding entries [10]. The PIM join message payload contains Multicast-address=G, PIM-join={RP,RPbit,WCbit}, PIM-prune=NULL.

Each upstream router creates or updates its multicast forwarding entry for (*,G) when it receives a PIM join with the WC and RP bits set. The interface on which the PIM join message arrived is added to the list of outgoing interfaces for (*,G). Based on this entry each upstream router between the receiver and the RP sends a PIM join message in which the join list includes the RP. The packet payload contains Multicast-address=G, PIM-join={RP,RPbit,WCbit}, PIM-prune=NULL.

The RP recognizes its own address and does not attempt to send join messages for this entry upstream. The incoming interface in the RP's (*,G) entry is set to null. RP reachability messages are generated by RPs periodically and distributed down the (*,G) tree established for the group. This allows downstream routers to detect when their current RP has become unreachable and triggers joining toward an alternate RP.

3.3 Switching from shared tree (RP tree) to shortest path tree (SPT)

When a PIM-speaking router on a shared tree, which has directly-connected members, wants to join the group with shortest paths, the router notices data packets for G that are sourced by an address Sn for which it does not have a multicast forwarding entry (Sn,G). As shown in figure 5, router A initiates a new multicast forwarding entry for (Sn,G), with SPT bit cleared indicating that the shortest path tree branch from Sn has not been completely setup, and in the mean time it still uses the shared tree to get packets from Sn. A timer is set for the (Sn,G) entry.

A PIM join message will be sent upstream to the best next hop toward the new source, Sn, with Sn in the join list: Multicast-address=G, PIM-join={Sn}, PIM-prune=NULL.

When a router which has a (Sn,G) entry with SPT bit cleared, starts to receive packets from the new source Sn on the interface used to reach Sn, it sets the SPT bit, and sends

[10] A RP bit in a forwarding entry indicates that the incoming interface check for that entry should be the RPF interface to the RP, not to the source. PIM prune messages with the RP bit set cause this bit to be set in the associated forwarding entry. The RP bit in an (S,G) entry indicates that periodic PIM join/prune should be sent toward the RP.

Figure 5: Example: Switching from shared tree to shortest path tree
Actions are numbered in the order they occur

a PIM prune toward RP if its shared tree incoming interface differs from its shortest path tree incoming interface, indicating that it no longer wants to receive packets from Sn via RP. In the PIM message toward RP, it includes Sn in the prune list, with RP bit set indicating that a negative cache [11] should be set up on the way to RP.

When the (Sn,G) entry is created, the outgoing interface list is copied from (*,G), i.e. all local shared tree branches are replicated in the new shortest path tree. In this way when a data packet from Sn arrives and matches on this entry, all receivers will continue to receive source packets along this path unless and until the receivers choose to prune themselves.

Note that a DR may adopt a policy of not setting up an (S,G) entry (and therefore not sending a PIM join message toward the source) until it has received *m* data packets from the source within some interval of *n* seconds. This would eliminate the overhead of sending (S,G) state upstream when small numbers of packets are sent sporadically. However, data packets distributed in this manner may be delivered over the suboptimal paths of the shared RP tree.

The DR may also choose to remain on the RP-distribution tree indefinitely instead of moving to the shortest path tree.

3.4 Steady state maintenance of router state

In the steady state each router sends periodic refreshes of PIM messages upstream to each of the next hop routers that is en route to each source, S, for which it has a multicast forwarding entry (S,G); as well as for the RP listed in the (*,G) entry. These messages are sent periodically to capture state, topology, and membership changes. A PIM message is also sent on an event-triggered basis each time a new forwarding entry is established for some new (Sn,G) (note that some damping function may be applied, e.g., a merge time). Optionally the PIM message could contain only the incremental information about the new source. The delivery of PIM messages does not depend on positive acknowledgement; lost packets will be recovered from at the next periodic refresh time.

3.5 Multicast data packet processing

Data packets are processed in a manner similar to existing multicast schemes. An incoming interface check is performed and if it fails the packet is dropped, otherwise the packet is forwarded to all the interfaces listed in the outgoing interface list (whose timers have not expired). There are two exception actions that are introduced if packets are to be delivered continuously, even during the transition from a shared to shortest path tree. First, when a data packet matches on an (S,G) entry with a cleared SPT bit, if the packet does not match the incoming interface for that entry, then the packet is forwarded according to the (*,G) entry; i.e., it is sent to the outgoing interfaces listed in (*,G) if the incoming interface matches that of the (*,G). In addition, when a data packet matches on an (S,G) entry with a cleared SPT bit, *and* the incoming interface of the packet matches that of the (S,G) entry, then the packet is forwarded and the SPT bit is set for that entry.

Data packets never trigger prunes. Data packets may trigger actions which in turn trigger prunes. In particular data packets from a new source can trigger creation of a new (S,G) forwarding entry. This causes S to be included in the prune list in a triggered PIM messages toward the RP; just as it causes S to be included in the join list in a triggered PIM message toward the source.

[11] A negative cache entry is a (S,G) entry on the RP tree. The RP bit is set, indicating that the associated prune messages should be sent up the shared tree toward the RP. In addition, the outgoing interface from which it receives a PIM prune message with (S,G) and the RP bit in the prune list, is deleted from the outgoing interface list. Data packets matching the negative cache are not sent to that interface.

3.6 Timers

A timer is maintained for each outgoing interface listed in each (S,G) or (*,G) entry. The timer is set when the interface is added. The timer is reset each time a PIM join message is received on that interface for that forwarding entry (i.e., (S,G) or (*,G)) [12].

When a timer expires, the corresponding outgoing interface is deleted from the outgoing interface list. When the outgoing interface list is null a prune message is sent upstream and the entry is deleted after 3 times the refresh period [13].

3.7 PIM-speaking routers on multi-access subnetworks

Certain multi-access subnetwork configurations require special consideration. When a LAN-connected router receives a prune from the LAN, it must detect whether there remain other downstream routers with active downstream members. The following protocol is used: when a router whose incoming interface is the LAN has all of its outgoing interfaces go to null, the router multicasts a prune message for (S,G) onto the LAN. All other routers hear this prune and if there is any router that has the LAN as its incoming interface for the same (S,G) and has non-null outgoing interface list, then the router sends a join message onto the LAN to override the prune. The join and prune should go to single upstream router that is the right previous hop to the source or RP; however, at the same time we want others to hear the join and prune so that they suppress their own joins/prunes or override the prune. For this reason the join is sent to a special multicast address which all routers on the same LAN (and only those on the same LAN) are members [14], with the IP address of the previous hop in the IGMP header.

3.8 Unicast routing changes

When unicast routing changes an RPF check is done and all affected multicast forwarding entries are updated. In particular, if the new incoming interface appears in the outgoing interface list, it is deleted from the outgoing list.

The PIM-speaking router sends a PIM join message out its new interface to inform upstream routers that it expects multicast datagrams over the interface. It sends a PIM prune message out the old interface, if the link is operational, to inform upstream routers that this part of the distribution tree is going away.

3.9 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.

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.

When multiple RPs are used, each source registers and sends data packets toward each of the RPs, but receivers only join toward a single RP. If one of the RPs fails, receivers that joined 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.

3.10 Summary

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 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" to 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 distribution is used, most data packets will arrive at receivers over a shortest path tree.

4 Open Issues

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

- **Interoperation with dense mode networks / regions:**

  A network or collection of networks should be able to choose whether to use sparse mode PIM as described here, or dense mode multicast to join a distribution tree, depending on the density of the group memberships in that region or on the scarcity/availability of bandwidth [15]. Links should be configurable to operate in dense mode or in sparse mode. If the group membership density is high or bandwidth is plentiful then it is efficient to use reverse path multicasting (RPM) or flood membership reports, since in general most links will be on a path from some source to some destination. For example, an expensive WAN link or inter-domain link *may* be configured as default sparse-mode. Most intra-domain or intra-campus links will probably be configured as default dense-mode.

[12] When a timer is reset for an outgoing interface listed in (*,G) entry, we should also reset the interface timers for all (S,G) entries which contain that interface in their outgoing interface list. Because some of the outgoing interfaces in (S,G) entry are copied from (*,G) outgoing interface list, they may not have explicit join messages from the downstream routers.

[13] Negative cache entries on the RP tree must be kept alive by receipt of Prunes. We do not want to delete such entries if (*,G) entry exists; otherwise, data packets will travel down both RP tree and SPT. It may not result in periodic duplicates (because of the RPF check), but it does waste a lot of network bandwidth.

[14] see [14], this address (224.0.0.2) is also used by routers to send PIM query packets to neighbor routers on the same LAN

[15] For this reason we have also developed a dense mode multicast scheme that uses DVMRP-like RPF, but that is unicast routing protocol independent [13]

The primary issue in splicing dense mode regions onto a distribution tree comprised, in whole or part, of sparse mode regions, is the incompatability between the data driven nature of dense mode, and the explicit join nature of sparse mode. In other words, the first group member in a dense mode region needs to have some way of initially pulling down the data packets from (or through) an upstream sparse mode region. Normally, data packets emanating from or traveling through a sparse mode region would not be sent to the dense mode region without explicit joins. We are working on a mechanism to address this problem that relies on getting the group member existence information to the border routers, and having border routers send explicit joins.

A second issue in splicing these "IP clouds" onto PIM trees is identifying which border router for the IP cloud should be the entry point for data packets from a particular source, and therefore which sources individual border routers should put in their join and prune lists. This is analogous to the LAN case when there is more than one router serving it. The designated router is the one that takes responsibility for serving the members on the LAN.

- **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 issues are PIM message size and the amount of memory used for 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.

  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 and identifying RPs:**

  An RP for a particular multicast group can be any IP-addressable entity in the internet. However, it is most efficient and convenient for the RP to be the directly-connected PIM-speaking 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 also happens to be on the SPT. 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 QOS 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-speaking 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 table. 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 PIM join messages toward source *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.

5 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.

A protocol implementation of PIM using extensions to existing IGMP message types is in progress. Simulation and implementation efforts are underway to characterize configuration criteria and deployment issues.

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 inter-domain routing and resource reservation protocols.

References

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

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

[3] J. Moy. Multicast extension to OSPF. *Internet Draft*, September 1992.

[4] D. Waitzman, C. Partridge, and S. Deering. Distance vector multicast routing protocol, November 1988. RFC1075.

[5] S. Deering. Host extensions for IP multicasting, August 1989. RFC1112.

[6] J. Moy. OSPF version 2, October 1991. RFC1247.

[7] J. Moy. MOSPF: Analysis and experience. *Internet Draft*, July 1993.

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

[9] R. Frederick. IETF audio & videocast. *Internet Society News*, 1(4):19, 1993.

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

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

[12] 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.

[13] S. Deering, D. Estrin, D. Farinacci, and V. Jacobson. IGMP router extensions for routing to dense multicast groups. *Internet Draft*, October 1993.

[14] S. Deering, D. Estrin, D. Farinacci, and V. Jacobson. IGMP router extensions for routing to sparse multicast groups. *Internet Draft*, October 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.