Protocol Independent Multicast (PIM), Sparse Mode Protocol Specification

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

draft-ietf-idmr-pim-sparse-spec-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 describes the protocol details. For the motivation behind our design and a description of the architecture, see [1].

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 the protocols defined to support our 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.

This document describes the protocol details. For the motivation behind our design and a description of the architecture, see [1].

2 PIM Protocol Overview

In this section we provide an an overview of the PIM protocol.

Traditional multicast routing protocols were designed for densely populated groups and 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:

(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).[2] 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 an RP-flag indicating that this is a shared tree entry.

Figure 1 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 1) 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-router (D in fig 1) sends a PIM-Register message, piggybacked with the data packet, to the RP(s) for that group. The RP responds by sending a join and an PIM-Register-Stop message toward the source. Processing of the join messages by intermediate routers (there are no intermediate routers between the RP and the source in fig 1) sets up a packet delivery path from the source to the RP(s). The PIM-Register-Stop message is sent when this path is setup, which tells the source to stop sending register messages to the RP.

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.

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

Figure 1: How senders rendezvous with receivers

After data packets are received on the new path, router B in fig 1 sends a PIM-Prune message toward the RP[3].

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 has directly connected 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 sparse mode mechanism, 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. 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 behavior in more detail. Section 3 provides the detailed protocol specification.

2.1 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 2. 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, see section 3.4.1) receives a report for a new group G it checks to see if it has RP address(es) associated with G. [4] A DR will identify a new group (i.e., one for which it

[3]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.

[4]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

Figure 2: Example: how a receiver joins, and sets up shared tree. Actions are numbered in the order they occur

has no existing multicast entries) as needing PIM 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 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[2]. 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.

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 2.8.

The DR (e.g., router A in fig 2) 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 wildcard entry; if there is no more specific match for a particular source, it will be forwarded according to this entry. A RP bit associated with this entry is also set, indicating that this entry, (*,G), represents state on the shared, RP, tree. 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 2.2).

2.2 Establishing the RP-rooted shared tree

The DR router creates a PIM-Join/Prune message with the RP address in its join list with the WC and RP 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.

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

[5] The PIM-Join/Prune message payload contains Multicast-Address=G, PIM-Join={RP,WCbit,RPbit}, PIM-Prune=NULL.

Each upstream router creates or updates its multicast forwarding entry for (*,G) when it receives a PIM-Join with the RP and WC 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/Prune message in which the join list includes the RP. The packet payload contains Multicast-Address=G, PIM-Join={RP,WCbit,RPbit}, 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.

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

When a PIM-router on a shared tree, which has directly-connected members that want to join the group with shortest paths, the router notices data packets for G that are *not* sourced by an address Sn for which it has a multicast forwarding entry (Sn,G). As shown in figure 3, 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 and this timer is reset whenever a data packet for (S,G) is received.

A PIM-Join/Prune message will be sent upstream to the best next hop toward the new source, Sn, with Sn in the join list. The payload contains 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 a PIM-Prune toward RP if its shared tree incoming interface differs from its shortest path tree incoming interface, indicating

[5] 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.

that it no longer wants to receive packets from Sn via RP. In the PIM message sent toward the RP, it includes Sn in the prune list, with the RP-bit set indicating that a negative cache should be set up on the way to RP[6]. The PIM-Join/Prune message payload contains Multicast-Address=G, PIM-Join=NULL, PIM-Prune={Sn,RPbit}.

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 a (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 (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. Note that (S,G) state must be maintained in all last hop routers when an SPT is maintained (and this is suboptimal when (*,G) and (S,G) overlap because you need both pieces of state to keep the joins going to the right places upstream).

The DR may also choose to remain on the RP-distribution tree indefinitely instead of moving to the shortest path tree. Similarly if a DR decides to switch back to RP tree from SPT, it can trigger an explicit PIM-Join/Prune message with Multicast-Address=G, PIM-Join={S, RPbit}, PIM-Prune=NULL. The message will travel up the RP tree and override the pruned state.

2.4 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.

2.5 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 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. For example, router *B* in fig 3 decides to switch to SPT at step 3, it creates a (Sn, G) entry with SPTbit

[6]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-Join/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.

set to 0. When data packets from Sn arrive at interface #2 of B, B sets the SPTbit to 1, which in turn triggers the sending of prunes toward the RP.

2.6 Timers

An entry timer is set for each (S,G) entry (including each (*,G) entry). The timer of an (S,G) entry is reset whenever a data packet for (S,G) is received and the timer for (*,G) entry and negative cache entries regarding G are reset whenever an RP-Reachability message for G is received. The timer expires after 3 times the refresh period, typically it is 270 seconds (because the RP-reahability message is sent every 90 seconds).

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)). [7]

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 (180 seconds). [8]

2.7 Unicast routing changes

When unicast routing changes an RPF check is done and all affected expected incoming interfaces are updated. In particular, 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.

The PIM-router must send a PIM-Join message out its new interface to inform upstream routers that it expects multicast datagrams over the interface. It must send 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.

2.8 Multiple Rendezvous Points (RPs) 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 last hop router, i.e., the DR, 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 to that RP will stop

---

[7] 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.

[8] 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.

receiving RP-Reachability messages and will start sending joins to one of the alternative RPs. Sources do not need to take special action.

2.9 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 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.

3 Detailed Protocol Specification

3.1 Receiver/Upstream PIM messages

3.1.1 PIM-Join/Prune messages

The join list in a PIM-Join/Prune message sent to a neighboring router, N, includes an address for each source, S, for which:

1. There is a multicast forwarding entry (S,G) or S is listed as the RP-entry for (*,G); and
2. N is the next-hop router used to send unicast packets to S; and
3. the outgoing interface list in the forwarding entry is *not* null. [9]

The prune list in a PIM-Join/Prune message sent to a neighboring router N, includes an address for each source, S, for which:

1. There is a multicast forwarding entry (S,G) or S is listed as the RP-entry for (*,G); and
2. N is the next-hop router used to send unicast packets to S; and
3. the outgoing interface list in the forwarding entry is null; and
4. If N is the next hop used to reach the RP, the prune list also includes an address for each source S, with RP bit set, for which there is a (S,G) multicast forwarding entry, its SPT bit is set, and N is *not* the next hop used to reach S.

PIM-Join/Prune messages are sent periodically, typically every 60 seconds, and the join and prune lists are populated as specified above. In addition four events will trigger PIM-Join/Prune messages:

[9] If the router has an (*,G) entry and it is the RP for G, it still has to include S in the join list of PIM-Join/Prune message even the outgoing interface list of (S,G) entry is null, because a source must register with all RPs, all RPs must set up (S,G) paths toward all sources.

1. Receipt of an IGMP report message for a new group G (i.e., one for which the receiving router does not have a (*,G) entry) will trigger a PIM-Join message toward the RP with the RP address and RP and WC bits set in the join list.
2. Receipt of a PIM-Join message for an (S,G) pair (including (*,G)) for which there is no current forwarding entries or the outgoing interface list of (S,G) entry is null, will trigger a PIM-Join message toward S (or RP) with S (or RP with RP and WC bits set) in the join list.
3. Receipt of packet on the new created (S,G) entry, i.e., an entry with SPT bit cleared, over the appropriate incoming interface triggers
   (a) setting of the SPT bit on (S,G) entry, and
   (b) sending a PIM-Prune message up the RP tree, i.e., toward the RP, with S and the RP bit set in the prune list.
4. When the outgoing interface list of (S,G) entry becomes null, indicating no more downstream receivers, a PIM-Prune with S in the prune list is sent upstream.

We do not trigger prunes based on data packets. Data packets that arrive on the wrong incoming interface are silently dropped.

Each router that receives a PIM-Join/Prune message from a downstream neighbor processes it as follows:

1. Notes the interface on which the PIM message arrived, call it I. The router accepts this PIM-Join/Prune message if this PIM-Join/Prune message is addressed to router itself; [10] otherwise, the router drops this PIM-Join/Prune message on the floor.
2. If an address Si in the list has WC bit set, Si is an RP address. Add I to the outgoing interface list of the (*,G) forwarding entry and set the timer for that interface (if there is no (*,G) entry, the router initializes one first). furthermore,
   (a) If I is a new interface in the (*,G) forwarding entry, then add I to all other existing (Si,G) forwarding entries with the exception of those Si listed in the prune list; and set the timers for that interface in each affected entry; or
   (b) If I is an old interface in the (*,G) forwarding entry, then reset the interface timers for all (*,G) and (Si,G) entries.

   If the RP address Si is different from the RP address listed in the existing (*,G) forwarding entry then:

   (a) If Si is greater than the listed RP-entry value, set RP-entry to Si; or
   (b) If Si is less than the listed RP-entry value. leave the RP-entry as is.

   The incoming interface is set to the interface used to send unicast packets to the RP in the (*,G) forwarding entry, i.e., RPF interface to the RP.

[10] In a point-to-point link, the router address is specified in IP packet header(*destination address* field); On a multi-access LAN, because the PIM-Join/Prune, PIM-Assert PIM-Mode messages are multicasted onto the LAN (see section 3.4), the destinating router address is specified in the IGMP header(*group address* field).

3. For each address Si in the PIM message join list whose WC bit is *not* set, and for which there is no existing (Si,G) forwarding entry, the router initiates one. [11] The outgoing interface is set to I, and the incoming interface is set to the interface used to send unicast packets to Si. If the interface used to reach Si is the same as the outgoing interface being built, this represents an error and the join should not be processed.

4. For any Si included in the join list of PIM-Join/Prune message, for which there is an existing (Si,G) forwarding entry,

   (a) if the RP bit is not set for Si listed in the join message, but the RP bit is set on the existing (Si,G) entry, the router clears the RP bit on (Si,G) entry and re-compute the incoming interface toward Si for that (Si,G) entry and send join to the new incoming interface; and

   (b) the router adds I to the list of outgoing interfaces, If I is not the same as the existing incoming interface.

5. For each address Si in the prune list, if there is an existing (Si,G) forwarding entry, the router deletes I from the list of outgoing interfaces. If the router has a current (*,G) forwarding entry, and if an (Si,G) entry also exists then the (Si,G) forwarding entry is maintained even if its outgoing interface list is NULL.

6. For any Si with the RP bit set, that is included in the prune list, for which there is *no* (Si,G) entry but there is a (*,G) entry , then the (Si,G) entry with RP bit set is created with the outgoing interface list copied from (*,G) entry, and the interface on which the prune was received is deleted. This acts as a negative cache so that packets from Si are not forwarded to pruned receivers.

3.2 Source/Downstream messages

3.2.1 PIM-Register messages

When a leaf router (first hop router) receives a multicast data packet from a locally connected host for which it has no (S,G) or (*,G) entries, the router sends a PIM-Register message to the RP for that group. The message indicates the group for which the source is registering.

The original data packet is encapsulated inside the Register packet. The message is sent as a unicast packet to the RP; it is not processed by the intermediate routers. If there are multiple RPs associated with the multicast group, then the source sends a Register message to each of them.

Subsequent data packets sent to the same group will trigger the same action until each RP has built a (S,G) path to source and send PIM-Register-Stop to source to stop the sending of PIM-Register messages. [12] When a router (i.e., the RP) receives a Register message, the router

1. decapsulates the data packet, and forwards it according to its local (*,G) forwarding entry; and

2. sets up an (S,G) forwarding entry with the outgoing interface list copied from the (*,G) outgoing interface list, its SPT-bit set to 0. The (S,G) entry is set up using the mask information, if provided, in the Register message. A timer is set for the (S,G) entry and this timer is reset whenever a data packet for (S,G) is received; and

---

[11] The router creates a (S,G) entry and copies all outgoing interfaces from the (*,G) entry, if it exists. If a router does not copy all outgoing interfaces from the (*,G) entry, all receivers on RP tree, downstream from outgoing interfaces other than the one newly added to (S,G), will not receive packets from source S. Data packets of S arriving from the RP will match the (S,G) entry instead of (*,G) entry, and will be dropped because the incoming interface is incorrect.

[12] When the SPT bit is set for an (S,G) entry in the RP, this signals to the RP that the path has been built and the RP sends a PIM-Register-Stop message to source S.

3. the (S,G) entry causes the RP to send a PIM-Join message for the indicated group toward the source of the Register message. The PIM-Join message includes the source's address in the join list; and

4. sends a PIM-Register-Stop message to the source after receiving a data packet from the incoming interface of (S,G) entry (the SPT bit is set also).

3.2.2 RP-Reachability messages

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-Host-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 toward 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 a PIM-Register-Stop message. The source of the Register messages will declare the RP down after 10 seconds without receiving a PIM-Register-Stop message.

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).

A router starts sending periodic RP-Reachability messages downstream when

1. it receives a PIM-Join/Prune message with its own address and WC, RP bits set in the join list; and

2. the incoming interface on its (*,G) entry is null[13].

The first condition is to make sure that it is an RP. The second condition is to make sure that only the *dominant* RP will send RP-Reachability messages, so the traffic can be minimized.

This obviates the need to do any kind of special configuration of RPs; any router can be an RP since RP behavior is triggered by the protocol itself. A router is responsible for initiating RP-Reachability messages to downstream nodes if it has a (*,G) entry with a null incoming interface.

The router sends the periodic RP-Reachability messages out all outgoing interfaces in the (*,G) entry. The period for this message is 90 seconds. The messages are addressed to the 224.0.0.2 class D address and the message content includes the RP and G.

When a router receives an RP-Reachability message for a group G it must compare the RP address listed in the message to the RP address listed in the current (*,G) RP-entry. If the RP listed in the message is greater than the RP listed in the (*,G) RP-entry, and if the next hop used to reach the listed RP is the same as the next hop used to reach the RP-entry, then the router replaces its current RP-entry with the RP address from the RP-Reachability message In more detail, when a router receives an RP-Reachability message it does the following; assume that router X receives an RP-Reachability message of RP1 from incoming interface I:

[13]This rule is needed when the current node was specified as an RP, but yielded to another RP in the multiple RP address arbitration process. If the other RP's address is bigger, the current entry will have *none* null incoming interface

1. Perform RPF check. If I is not the best next hop to RP1, drop this RP-Reachability message.
2. If the incoming interface of (*,G) entry is *not* null and *not* I, drop the RP-Reachability message.
3. If the incoming interface of (*,G) is I, compare RP1 with the address in RP-entry, say RP2. If RP1 is larger than RP2, set RP-entry to RP1 and propagate the RP-Reachability message downstream. Otherwise, drop the RP-Reachability message.
4. If the incoming interface of (*,G) is null and WC bit is set then this router is currently acting as an RP for G. In this case, compare RP1 with X. If RP1 is larger than X, set RP-entry to RP1, set the incoming interface to the RPF interface used to reach RP1. Also, propagate the RP-Reachability message downstream. Otherwise, if RP1 is less than X, drop the RP-Reachability message.

3.3 Data packet forwarding

When a router receives a data packet from a source S, addressed to a multicast group G, the router checks to see if it has an (S,G) entry.

1. If there is an (S,G) entry, the packet is forwarded according to the outgoing interfaces list in that entry if the RPF check succeeds. The SPT bit in the (S,G) entry is also set indicating the shortest path is established. The entry may also indicate that the data packet should be encapsulated in a Register message to one or more RPs if it is the first hop router. An implementation may cache the RP information and set a pointer from the entry so that future data packets will be processed more quickly while waiting for the Register-Stop message to arrive from the RP(s).
2. If there is an (S,G) entry and the SPT bit is not set, if the RPF check fails, then the packet is forwarded according to the outgoing interfaces list in (*,G) entry if there is one and the RPF check on (*,G) entry succeeds.
3. If there is no (S,G) entry but there is a (*,G) entry, then the packet is forwarded according to the outgoing interface list in that entry if the RPF check succeeds.
4. If there is neither a (S,G) nor (*,G) entry, the packet is dropped. (In dense mode PIM, a data packet could trigger the creation of an (S,G) entry, see section 3.8 for more detail.)

3.4 Operation over multi-access subnetworks

The PIM-Query, RP-Reachability, PIM-Join/Prune, PIM-Assert(see section 3.6) and PIM-Mode(see section 3.7) messages are multicasted onto multi-access LANs. There are several multi-access subnetwork configurations that require special consideration.

3.4.1 Designated routers

When there are multiple PIM routers on a multi-access network, only a single router must be responsible for the following actions

1. Soliciting group membership from hosts and sending PIM-Join messages.
2. Sending Register messages on behalf of a connected source host when it sends a multicast packet.
3. Forwarding multicast packets onto the multi-access network.

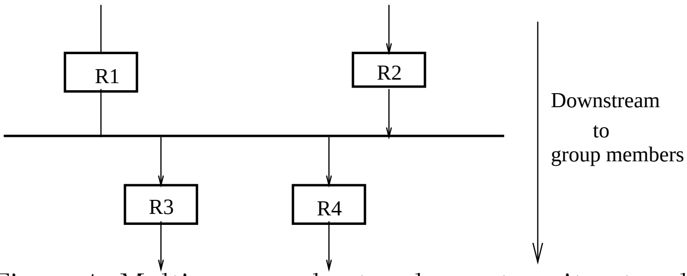

Figure 4: Multi-access subnetwork as a transit network

This is done with a simple Designated Router (DR) election. Neighboring routers send PIM-Query packets to each other. The packets are sent to 224.0.0.2. The largest IP addressed system will assume role as DR. The default transmission interval is 30 seconds. A router should detect the DR as unreachable when it does not receive a Query in 3 times the transmission interval (90 seconds). There will be one DR that supports all groups per multi-access network. The DR sends periodic IGMP-Host-Query packets to 224.0.0.1 soliciting hosts to respond. The DR sends multicast packets using the data link address that is mapped from the IP Class D multicast address.

3.4.2 Multi-access subnetwork as a transit network

When a LAN is used as a transit network among routers, it is required that a single router forward multicast packets to downstream routers. A single forwarder is chosen by the receipt of PIM messages sent by each of the downstream routers with the address of the target router embedded in the join message. All routers that use the LAN as their incoming interface for multicast packets from a particular source, will expect it from a single forwarder. This router is the one they use for sending unicast packets to the source. All routers will select the same router. In the case of equal-cost unicast paths, the largest IP addressed next-hop is used. The forwarder forwards multicast packets using data link address that is mapped from the IP Class D multicast address. The multicasted data packets will be seen by all other routers connected to the LAN. For each router, if it has an entry for (S,G) or (*,G) and the LAN is the indicated incoming interface then the router will forward the packet. If there is no such entry or if the incoming interface is not the LAN then the packet will be silently dropped.

Figure 4 shows the case where a multi-access network is used as a transit network.

In figure 4, both R3 and R4 have downstream group members. They will send their PIM-Join messages toward the RP, and therefore select either R1 or R2 to send the message. Assume R2 is the shorter path to the RP. Later, when multicast datagrams travel from the RP, they will come through R2 only, avoiding duplicates on the LAN. The same procedure takes place when the source-based distribution tree is built. Whatever router on the LAN is chosen is also responsible for delivering multicast packets to host members, if they were present.

3.4.3 Parallel routers

Figure 5 illustrates the behavior of routers that are connected in parallel to a common LAN.

Assume R1 is the RP for group G, R3 is DR for LAN1, R2 the DR for LAN2, and that LAN1 is the preferred path among routers to reach the RP. If the receivers of group G (H1 and H2) join first, R2 will send a PIM-Join to R1, the RP, out LAN1. R2 builds a multicast entry for (*,G) with incoming interface LAN1 and outgoing interface LAN2. R1 receives the PIM-Join message and builds a multicast entry for (*,G) with incoming interface set to null (since it is the RP) and outgoing interface set to LAN1.

When S sends a multicast datagram, the DR for LAN1, R3, will encapsulate the data packet in a Register message and send it to R1, the RP, out LAN1.

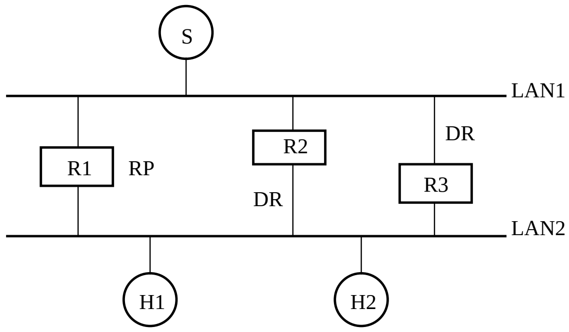

Figure 5: Parallel routers

The RP, R1, decapsulates the data packet and forwards it onto LAN1, as indicated in the outgoing interface list of R1's (*,G) entry.

The RP, R1, then processes the Register part of the message and sets up an (S,G) entry with LAN1 as the incoming interface and a null outgoing interface list; the outgoing interface list is copied from (*,G) but LAN1 is *not* included in (S,G) outgoing interface list because LAN1 is the incoming interface.

R1 triggers a join toward S. When R1 does a lookup on S it finds that S is on a directly connected LAN and sends the join to the DR for that LAN, i.e., R3.

When the DR, R3, receives the join it builds an (S,G) entry with LAN1 as the incoming interface. Since there is no (*,G) entry, the outgoing interface list is set to null.

Subsequent data packets from S for G that arrive from LAN1 will be silently dropped by R1 and R3 since the (S,G) entries have null outgoing interface lists. But R2 will accept those datagrams since the incoming interface of the (*,G) entry is LAN1 and the outgoing interface list is not null (i.e., LAN2); multicast datagrams for (S,G) will be forwarded to LAN2 and H1 and H2 will receive them.

If shortest path trees are desired, when R2 receives the multicast datagram from S and finds that the longest match is on (*,G) (i.e., there is no (S,G) entry in R2), R2 creates an (S,G) entry, sets the incoming interface to LAN1 (since that is the interface used to send packets to S), and copies the outgoing interface list from the existing (*,G) entry (in this case LAN2). R2 forwards subsequent multicast datagrams for (S,G) onto LAN2.

When R2 creates (S,G), it also triggers a PIM-Join message to the next hop to S; in this case S is directly connected so the PIM-Join is sent to the DR for that LAN, R3. R3 receives the join but does not add LAN1 to its outgoing interface list for (S,G) because LAN1 is the incoming interface for (S,G).

3.4.4 Prunes on multi-access subnetworks

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

[14] see [3], this address (224.0.0.2) is also used by routers to send PIM-Query packets to neighbor routers on the same LAN.

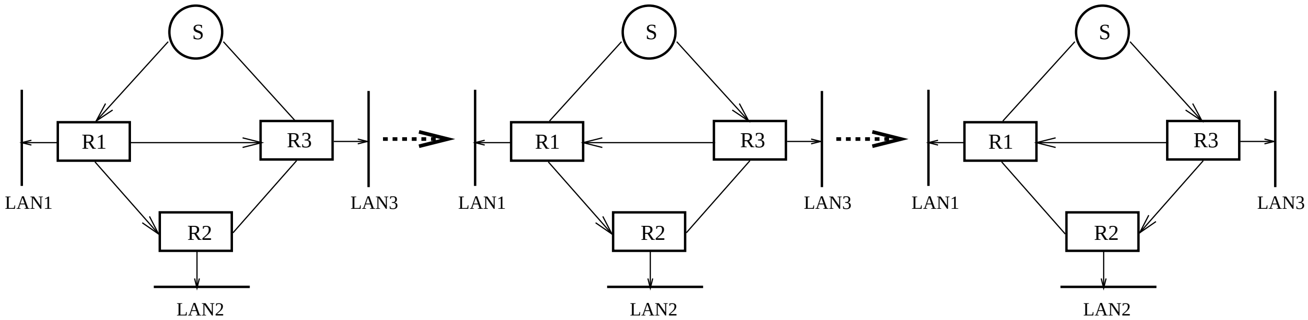

Figure 6: Responding to unicast routing changes

header. [15]

3.5 Responding to unicast routing changes

If the unicast route goes unreachable, all multicast entries for (S,Gi) should be modified to have null outgoing interface lists, however the entries should not be deleted immediately; this causes periodic prunes to be sent and multicast packets to be discarded. The entry should be kept alive for the remainder of the timeout lifetime. This helps to eliminate transient multicast routing forwarding loops. If the unicast route has a new next-hop interface, the (S,Gi) entries must be updated.

Figure 6 shows how a multicast forwarding loop can be avoided. Assume all LANs have members for a given group. The arrows represent the expected interface each router will receive multicast datagrams on, as well as the outgoing interfaces with respect to Source.

If the path to Source changes, R1 and R3 may converge on the new path before R2, e.g., R1 uses its link to R3 as its expected incoming interface and R3 uses its new shortest path link to Source. In this state, R1 and R3 were informed about a topology change for Source and changed their incoming interfaces. Both R1 and R3 send joins up their new incoming interfaces. R1 also deleted its outgoing interface to R3 because this interface is used as an incoming interface. R2, however has not been informed about the topology change.

If R1 received a multicast datagram on its old expected interface, it would silently drop it. This would happen if upstream routers from R1 to the Source had old routing information. If upstream routers have converged on a new path all datagrams will enter this part of the network through R3, and it would forward appropriately to its LAN and to R1 which expects it. R1 would forward to its LAN as well as R2. R2, using out of date expected incoming interface, would also forward the packet. Once R2 is informed of the topology change, it will change its expected incoming interface to R3 and will send a prune to R1 and a join to R3.

More generally, if unicast routing changes and the router in question has not converged then one of two situations exists. In the first, one or more of the existing outgoing interfaces may no longer reach any receivers. In this case data packets are forwarded until they reach a router that has converged and finds that the incoming interface for the packet is not right; in which case the packet will be dropped. The cost of this transient condition is the continued sending of data packets down links that do not lead to receivers; this can occur for the duration of a refresh period.

The second situation occurs when data packets begin to arrive over an incoming interface other than the one listed in the corresponding (S,G) entry. This occurs when upstream has converged and the router

[15] The IP address goes in the group address field which is the bottom 4 bytes of the 8 byte IGMP header. Upstream routers delay for $5 + e$ seconds after receiving a prune before deleting the interface from (S,G) entry's outgoing list. The join sent by some other router who has a non-null outgoing interface list for (S,G) entry is sent with some randomness, between $1 - 5$ seconds. e should be a function of average delay through an ethernet taking into account contention.

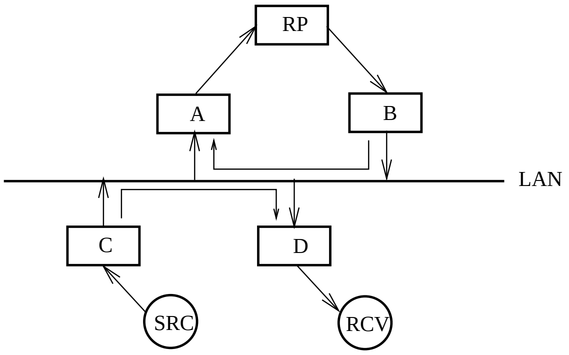

Figure 7: Looping problem on multi-access subnetworks

in question has not. In this case, data packets will be dropped instead of delivered for a time less than or equal to the convergence time + refresh period.

3.6 Looping problem on multi-access subnetworks

In the protocol described thus far, a looping problem can result when the RP tree and the shortest path tree overlap on a multi-access link, and the paths were asymmetric. Below we describe the problem and the solution.

In figure 7, assume that the RP thinks A is the best path to SRC, and D thinks that B is the best path to the RP, the loop is created by the following scenario

1. SRC registers to the RP. The RP sends a join via A to the SRC. This causes A and C to install (S,G) state pointing to the RP.

2. RCV joins and D sends a join to the RP. This causes B and D to install (*,G) state pointing to RCV.

3. When SRC sends a data packet, C forwards it onto the LAN, A and D will pick it up, A will send it to the RP and D will send it to RCV. The RP will send the packet to B which will in turn send it onto the LAN, and A and D will accept it again so a loop is created among A, RP and B and RCV will keep receiving duplicates.

We fix this bug with two simple mechanisms; the first avoids the situation from occurring, and the second detects and recovers from the situation once it does occur. The first approach takes advantage of the fact that all PIM-Join messages are multicasted onto the LAN, and each router can take proper action by monitoring the PIM-Join messages on the LAN. The second approach uses a new PIM-Assert message to stop all but one router from sending the packets in question onto the LAN. We cannot rely on the first approach alone to kill the loop in all contexts because PIM-Join messages are not sent reliably. We nevertheless implement the first mechanism as well as the second because the first will reduce the incidence of duplicate packets in the common case when PIM-Join messages are not lost.

The two rule are specified below in more detail. Following is rule 1.

1. If router C with (S,G) entry sees a join message for (*,G) addressed to router B from the LAN, and the LAN is an outgoing interface listed in (S,G) entry, then it sends a PIM-Join/Prune message unicasted to B with Multicast-Address=G, PIM-Join=NULL, PIM-Prune={S,RPbit}. [16] When Router B receives this message it will create a negative cache entry for (S,G) with the outgoing

[16] This negative cache pruning message should be unicasted to the upstream router instead of multicasted to the LAN Because this message should not be overrided by other routers on the LAN.

interface list copied from the (*,G) outgoing interface list except LAN, so later packets for (S,G) will not be forwarded onto LAN.

2. If router B with (*,G) entry sees a join message for (S,G) addressed to router C from the LAN, and the LAN is an outgoing interface listed in (*,G) entry, then it creates a negative cache entry for (S,G) with the outgoing interface list copied from the (*,G) outgoing interface list except LAN, so later packets for (S,G) will not be forwarded onto LAN.

The second approach uses a PIM-Assert message, see [4]. When a packet is received on an outgoing interface for (S,G), a PIM-Assert message is sent on that interface (i.e., it is addressed to 224.0.0.2). Routers on the LAN will process the PIM-Assert according to the following rules; assume that router X receives a PIM-Assert message for (S,G) from incoming interface I generated by route Y.

1. If X has no (S,G) entry or I is not one of the outgoing interfaces for (S,G) entry, the PIM-Assert is ignored.
2. Otherwise, X deletes I from (S,G)'s outgoing interface list if one of the following conditions stands
   (a) X's (S,G) entry is a negative cache or a (*,G) entry but Y's (S,G) entry is not, or
   (b) They both are negative cache entries or (*,G) entries but Y is greater than X, or
   (c) They both are not negative cache and (*,G) entries but Y is greater than X.

If none of the above conditions stands, X sends a PIM-Assert message for (S,G) onto the LAN, Y receives this Assert and will process it according the above rules.

3.7 Optimizing control message overhead

[17] If a router finds that a large number of its outgoing interfaces have joined a particular group, the router can switch all interfaces into *implicit join mode* to reduce the overhead of periodic traffic. The router does so by sending a PIM-Mode message out each of the interfaces for that entry. The message indicates the group address and a wildcard source address (indicating it is for all sources for which the generator of the Mode message is the upstream router), and the desired mode, *periodic join mode* or *implicit join mode*. The message only travels a single hop downstream to the directly connected neighbors. The downstream routers will mark the incoming interface for the specific entries as implicit or explicit periodic join mode, according to the flag set in the mode message. Thereafter these neighbors will *not* need to send periodic PIM-Join/Prune messages upstream; instead they will send explicit PIM-Prune if and when the outgoing interface list becomes null (i.e., no local or downstream members).

If a router finds later that the majority of outgoing interfaces are pruned, instead of joined, the router can switch its neighbors back into periodic join mode. It does so by sending a PIM-Mode message indicating the (*,G) entry and periodic join mode. [18] The downstream routers set the incoming interface bit to periodic join mode for each S,G entry that indicates the Mode message sender as its upstream router. This causes the recipient of the Mode message to return to periodically sending join messages upstream for those entries.

When a new entry is initialized, the router treats all outgoing interfaces in periodic join mode and its incoming interface is also be marked as periodic join mode. A timer is set for each outgoing interface and

[17] 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 !

[18] An upstream router could list individual (S,G) entries either to facilitate processing by the downstream neighbor or to handle different (S,G) entries differently.

the interface is timed out and deleted from the outgoing interface list if the periodic join is not received on time. If the router later switches all outgoing interfaces for G into implicit join mode, the timers on those interfaces are cleared, which means those interfaces will never be timed out and they will be deleted from the outgoing interface list only by receiving an periodic prune message from downstream.

Other interfaces that are *not* in the outgoing interface list still need periodic join messages in order to join, they will not be put into the outgoing interface list automatically. If the router switches from implicit join mode to periodic join mode, the timer on each outgoing interface is set again and the outgoing interface management mechanism returns back to the normal periodic refresh behavior.

It will not cause any problem if a router marks an incoming interface as periodic join mode and its upstream treats it as implicit join mode. When the upstream router receives the unnecessary (i.e., the interface is already in the outgoing interface list and the interface is in implicit join mode) explicit join message, the router will send back a PIM-Mode message. The above scenario happens when a newly joined (or rebooted) router starts sending periodic join messages upstream and the mode message has not yet been received. However, if the upstream router is in periodic join mode, and a downstream router is in implicit join mode, the interface will be timed out and packets will not be forwarded as they should be. Upon startup this scenario will not occur because all routers start off joining in periodic explicit join mode. However, the above scenario can occur in one of the two following cases :

1. If an upstream router switches to implicit join mode, and then back to periodic join mode, but the second message is not received by a downstream router. We do not want to send Mode messages reliably because of the overhead of sending reliable messages to a group of downstream routers connected to a LAN. To prevent having the affected downstream router(s) timed out from the outgoing interface list indefinitely, Mode messages that indicate periodic join mode are always sent periodically, every 60 seconds. Apparently, this scheme adds extra overhead in periodic join mode. However, we argue that the number of outgoing interfaces is small when an entry is treated as periodic join mode and the number of outgoing interfaces is large when it is treated as implicit join mode. Therefore, the savings by switching to implicit join mode is much greater than the overhead added to the periodic join mode.

2. If an upstream router switches to implicit join mode, and then it crashed. It lost all states(i.e., multicast forwarding entries) and it takes periodic join mode as the default after recovery. It has to inform all its downstream routers to switch back to periodic join mode. We solve this problem by add a reset flag in PIM-Query message. (See section 4) When a new router comes up, it includes the reset flag in its first 3 PIM-Query messages. All the neighboring routers with entries whose upstream is the new router will switch back to periodic join mode. During these 3 PIM-Query refresh time, the new router remains in periodic join mode. [19]

An incoming interface should be marked as periodic join mode whenever the incoming interface is changed, and an explicit join is sent to the new interface and an explicit prune is sent to the old interface.

On a multi-access LAN, all downstream routers with respect to (S,G) will converge to treating the incoming LAN as either in periodic join mode or in implicit join mode since they are all receiving the same mode messages and the upstream router is in control.

The (Si,G) entries in a single downstream router may have different modes because they may have different upstream routers, which independently determine the mode. We expect the common form of usage will be to treat all entries for a particular group in the same mode. For this reason, mode messages allow the use of wildcarding.

[19] The reset flag is only included in the first 3 Query because it is equal to the neighbor timeout period. If all the 3 messages are lost, the downstream router will consider its upstream router is down, it then changes the incoming interface and switches to periodic join mode automatically.

3.8 Hybrid dense mode – sparse mode behavior

We define the boundary of dense mode PIM and sparse mode PIM by interfaces connected to each router. That is, an interface can be either configured as *dense mode acceptable* (DM) or *sparse mode only* (SM). A DM interface operates in dense mode PIM, and a SM interface operates in sparse mode PIM.

Each router decides its own operation mode for each interface, if it is DM, the router interacts with all its downstream routers on that interface in dense mode; if it is SM, the router interacts with all its downstream routers on that interface in sparse mode. The downstream router knows its upstream router's decision by receiving PIM-Query messages. An interface mode flag is included in PIM-Query indicating the operation mode of that interface. (See section 4) It the flag is set, the upstream router is SM; if it is not set, the upstream router is DM. On a multi-access LAN, all downstream routers will converge to treating the incoming LAN as either SM or DM. since they are all receiving the same PIM-Query messages and the upstream router is in control.

When an event-triggered join message is sent out to a DM interface, it is converted into a PIM-Graft message. Periodic join messages are never sent out to a DM interface. When a graft message is sent out to a SM interface, it is converted into a PIM-Join message. A router will send periodic join messages if the incoming interface is SM.

If a router receives a join (or graft) for (S,G) and there is no (S,G) forwarding entry, the router creates an (S,G) entry. Its outgoing interface list is populated with all interfaces that are configured as DM plus the incoming interface of the join (or graft) message. That is, data packets will be flooded to all DM interfaces. [20] A SM interface is added to the outgoing interface list only by explicit join and a DM interface could be added to the outgoing interface list automatically or by explicit graft. However, a DM interface is deleted from the outgoing interface list only by explicit prune and a SM interface could be deleted from the outgoing interface list automatically or by explicit prune. A timer is associated with each outgoing interface. A pruned DM interface is added back to an outgoing interface list after the interface timed out; a joined SM interface is deleted from an outgoing interface list after the interface timed out.

If a data packet is received from a DM interface and there is neither a (S,G) nor (*,G) entry, the router creates an (S,G) entry. The outgoing interface list is populated with all outgoing interfaces that are configured as DM. If there are no such interfaces, the outgoing interface list is null. In addition,

1. If there is one or more interface that is not configured as DM (i.e., configured as SM),

2. If this is the first hop router for S, and

3. If RP information is available for G, then

the data packet is encapsulated in a Register message to the RP(s).

4 Packet Types

RFC-1112, see [5], specifies two types of IGMP packets for hosts and routers to convey multicast group membership and reachability information. An IGMP-Host-Query packet is transmitted periodically by routers to ask hosts to report which multicast groups they are members of. An IGMP-Host-Report packet is transmitted by hosts in response to received queries advertising group membership.

This document introduces new types of IGMP packets that are used by PIM routers. The packet format is shown in figure 8.

[20] A (*,G) entry is strictly sparse mode, its outgoing interface list is not populated with DM interfaces. In other words, a (*,G) entry treats all interfaces as SM and all operations on interfaces are sparse mode.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Version	Type	Code	Checksum
Group Address			

Figure 8: IGMP packet format

Version This memo specifies version 1 of IGMP. Version 0 is specified in RFC-988 and is now obsolete.

Type There are five types of IGMP messages:

1 = Host Membership Query
2 = Host Membership Report
3 = Router DVMRP Messages
4 = Router PIM Messages
5 = Trace Messages

Code Codes for specific message types. Used only by DVMRP and PIM. PIM codes are:

0 = Router-Query	
1 = Register	
2 = Register-Stop	
3 = Join/Prune	
4 = RP-Reachability	
5 = Assert	
6 = Graft	dense-mode PIM only
7 = Graft-Ack	dense-mode PIM only
8 = Mode	

Checksum The checksum is the 16-bit one's complement of the one's complement sum of the entire IGMP message. For computing the checksum, the checksum field is zeroed.

Group Address In a IGMP-Host-Query message, the group address field is zeroed when sent, ignored when received. In a IGMP-Host-Report message, the group address field holds the IP host group address of the group being reported. PIM-Join/Prune, PIM-Assert and PIM-Mode messages have the group address field zeroed when sent to a point-to-point link, and have the next hop router address in it when sent to a multi-access LAN. PIM-Register and Register-Stop messages have the group address field zeroed when sent. (See section 4.2 for RP-Reachability format.) PIM-Query message has the first 30 bits in the group address zeroed. The 30^{th} bit carries the sparse mode reset flag, if 1, the modes of entries in the downstream routers are reset to periodic join mode. The 31^{st} bit carries the mode of the interface where the PIM-Query is sent, if 1, the mode on the interface is configured as sparse mode only; if 0, the mode on the interface is configured as dense mode acceptable. (See section 3.8.)

4.1 PIM-Join/Prune, PIM-Assert and PIM-Mode messages

PIM-Join/Prune, PIM-Assert and PIM-Mode messages have the following additional information appended to the fixed header (RP-Reachability message type will be described in the next section, see section 4.2). The format of the additional information is shown in figure 9.

<table>
<tr><td colspan="4">0   1   2   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td></tr>
<tr><td>Reserved</td><td>Maddr Length</td><td>Addr Length</td><td>Number of Groups</td></tr>
<tr><td colspan="4">Multicast Group Address-1</td></tr>
<tr><td colspan="4">Multicast Group Address-1 Mask</td></tr>
<tr><td colspan="2">Number of Join Source</td><td colspan="2">Number of Prune Source</td></tr>
<tr><td colspan="4">Join Source Address-1</td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td colspan="4">Join Source Address-n</td></tr>
<tr><td colspan="4">Prune Source Address-1</td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td colspan="4">Prune Source Address-n</td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td colspan="4">Multicast Group Address-n</td></tr>
<tr><td colspan="4">Multicast Group Address-1 Mask</td></tr>
<tr><td colspan="2">Number of Join Source</td><td colspan="2">Number of Prune Source</td></tr>
<tr><td colspan="4">Join Source Address-1</td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td colspan="4">Join Source Address-n</td></tr>
<tr><td colspan="4">Prune Source Address-1</td></tr>
<tr><td colspan="4">⋮</td></tr>
<tr><td colspan="4">Prune Source Address-n</td></tr>
</table>

Figure 9: Additional packet format for PIM-Join/Prune, PIM-Assert and PIM-Mode messages

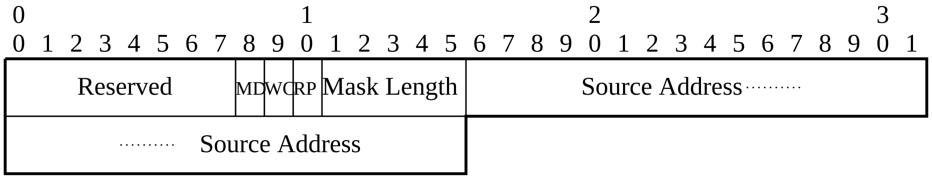

Figure 10: Source address format

Reserved Unused field, zeroed when sent, ignored when received.

Addr Length The length in bytes of the encoded source addresses in the Join and Prune lists.

Maddr Length The length in bytes of the encoded multicast addresses.

Number of Groups The number of multicast group sets contained in the message.

Multicast group address For IP, it is a 4-byte Class D address.

Multicast group address mask A bit mask used against the multicast group address. This is the method to describe a range of multicast addresses. If the multicast group address field describes a single group address, the value must be 255.255.255.255.

Number of Join Sources Number of join source addresses listed for a given group. For PIM-Assert message, this is the number of assert source addresses listed for a group. For PIM-Mode message, this is the number of source addresses listed for a group whose modes are changed.

Join Source Address-1 ∼ n This list contains the sources that the sending router will forward multicast datagrams for if received on the interface this message is sent on. The address 0.0.0.0 indicates a join for all sources.

Number of Prune Sources Number of prune source addresses listed for a given group. PIM-Assert and PIM-Mode messages only use the join list, so the number of prune sources should always be 0.

Prune Source Address-1 ∼ n This list contains the sources that the sending router does not want to forward multicast datagrams for when received on the interface this message is sent on. The address 0.0.0.0 indicates a prune for all sources.

4.1.1 Source address format

In PIM messages, all source addresses will have the format shown in figure 10.

Reserved Unused field, zeroed when sent, ignored when received.

MD Used when sending Mode messages. An upstream router indicates to downstream neighbor(s) if the specific group should operate in periodic join mode. If the bit is clear, the routers should operate in implicit join mode which means they do not send periodic join messages to the upstream router. The *Source Address* should be set to 0.0.0.0 in Mode messages with the WC bit and RP bit set to 0 if wildcarding is used.

WC The WC bit is a 1 bit value. If 1, the Join or Prune applies to the (*,G) entry. If 0, the Join or Prune applies to the (S,G) entry where S is Source Address. Joins and Prunes sent toward the RP should have this bit set.

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Version	Type	Code	Checksum
Group Address			
Group Address Mask			
RP Address			

Figure 11: RP-Reachability message packet format

RP The RP bit is a 1 bit value. If 1, the information about (S,G) is sent toward the RP. If 0, the information should be sent about (S,G) toward S, where S is Source Address.

Mask Length Mask length is 6 bits. The value is the number of contiguous bits left justified used as a mask which describes the address. The mask length must be less than or equal to $Addr\ Length * 8$.

Source Address The address length is indicated from the *Addr Length* field at the beginning of the header. For IP, the value is 4 octets. This address is either an RP address (WCbit = 1) or a source address (WCbit = 0). When it is a source address, it is coupled with the group address to make (S,G).

Represented in the form of $< WCbit >< RPbit >< Mask\ length >< Source\ address >$:
A source address could be a host IP address :
$< 0 >< 0 >< 32 >< 192.1.1.17 >$
A source address could be the RP's IP address :
$< 1 >< 1 >< 32 >< 131.108.13.111 >$
A source address could be a subnet address to prune from the RP-tree :
$< 0 >< 1 >< 28 >< 192.1.1.16 >$
A source address could be a general aggregate :
$< 0 >< 0 >< 16 >< 192.1.0.0 >$

4.2 RP-Reachability message

RP-Reachability messages have the packet format shown in figure 11.

Each RP will send RP-Reachability messages to all routers on its distribution tree for a particular group. These messages are sent so routers can detect that an RP is reachable. Routers that have attached host members for a group will process the message.

The RPs will address the RP-Reachability messages to 224.0.0.2. Routers that have state for the group with respect to the RP distribution tree will propagate the message. Otherwise, the message is discarded. If an RP address timer expires, the router should attempt to send an PIM join message toward an alternate RP provided for that group if one is available.

Group Address Group address associated with RP.

Group Address Mask A bit mask that allows to describe group ranges. Must be set to 255.255.255.255 when *Group Address* describes a single group address.

RP Address The rendezvous point IP address of the sender.

Finally, in a future version of this document we will specify a new IGMP message type that will allows hosts to advertise a list of 1 to n RP addresses associated with a particular group address.

5 Pseudocode

- Terminology :

all-host-group	224.0.0.1
all-router-group	224.0.0.2
router-itself	local router address
RPbit	indicating negative cache entry
PJ	periodic join mode
IJ	implicit join mode
I.timer	timer for subnetwork I of outgoing interface list
I.DR	DR address on subnetwork I
N.timer	timer for neighbor N
G.timer	timer for group G
G.RP	RP entry for group G
(S,G).timer	timer for (S,G) entry, including (∗,G)
(S,G).WCbit	WC bit for (S,G) entry, including (∗,G)
(S,G).RPbit	RP bit for (S,G) entry, including (∗,G)
(S,G).SPTbit	SPT bit for (S,G) entry, including (∗,G)
(S,G).iif	incoming interface for (S,G) entry, including (∗,G)
(S,G).oif	outgoing interface list for (S,G) entry, including (∗,G)
(S,G).imode	mode of (S,G).iif, PJ or IJ, including (∗,G)
(S,G).omode	mode of (S,G).oif, PJ or IJ, including (∗,G)

- Timer Expiration :
  1. On expiration of (S,G).timer on (S,G) entry, including (∗,G)
     if (S,G).RPbit is clear $\vee \neg\exists$ (∗,G) entry, remove (S,G) entry;
  2. On expiration of I.timer on outgoing interface I of (S, G).oif
     delete I from (S,G).oif;
  3. On expiration of G.timer on group G of subnetwork I
     $\forall$ source S, including RP : delete I from (S,G).oif;
  4. On expiration of N.timer on neighbor N of subnetwork I
     if I.DR = N, I.DR $\leftarrow$ largest neighbor, including *router-itself*;

- Sending Periodic Messages :
  1. On each 30-second interval
     $\forall$ subnetwork I : send PIM-Query to *all-router-group* on I;
  2. On each 60-second interval
     $\forall$ (S,G) entry, including (∗,G) :
        if (S,G).RPbit is set, recompute (S,G).iif toward G.RP;
        else, recompute (S,G).iif toward S;
        if (S,G).iif is changed, (S,G).imode $\leftarrow$ *PJ*;
     $\forall$ subnetwork I :
        send PIM-Join/Prune to *all-router-group* on I
           with PIM-Join $\leftarrow$ {(S,G) | (S,G).iif = I $\wedge$ (S,G).oif $\neq$ *null* $\wedge$ (S,G).imode = *PJ*},
                 PIM-Prune $\leftarrow$ {(S,G) | (S,G).iif = I $\wedge$ (S,G).oif = *null* $\wedge$ (S,G).imode = *PJ*} $\cup$
                  {(*RPbit*,S,G) | (∗,G).iif = I $\wedge$ (S,G).iif $\neq$ I $\wedge$ (S,G).SPTbit is set $\wedge$ (S,G).imode = *PJ*};
        send PIM-Mode to *all-router-group* on I
           with PIM-Mode $\leftarrow$ {(*PJ*,S,G) | (S,G).oif $\in$ I $\wedge$ (S,G).omode = *PJ*},

if I.DR = *router-itself*, send IGMP-Host-Query to *all-host-group* on I;

3. On each 90-second interval
   $\forall$ (∗,G) entry :
   if G.RP = *router-itself*,
   $\forall$ subnetwork I $\in$ (∗,G).oif : send RP-Reachability to *all-router-group* on I;

- Receiving/Accepting IGMP/PIM Messages :
  1. On receiving IGMP-Host-Report of group G on subnetwork I
     if I.DR = *router-itself*,
     reset G.timer;
     if $\neg\exists$ (∗,G) entry, initialize (∗,G) entry;
     if I $\notin$ (∗,G).oif $\wedge$ I $\neq$ (∗,G).iif, add I to (∗,G).oif;
     $\forall$ (S,G) entry, including (∗,G) : reset I.timer on (S,G).oif;
  2. On receiving PIM-Query from neighbor N on subnetwork I
     reset N.timer;
     if N > I.DR, I.DR $\leftarrow$ N;
  3. On accepting PIM-Join/Prune on subnetwork I
     $\forall$ (flag,S,G) $\in$ PIM-Join :
     if $\neg\exists$ (S,G) entry, initialize (S,G) entry;
     if I $\notin$ (S,G).oif $\wedge$ I $\neq$ (S,G).iif, add I to (S,G).oif;
     if flag = *WCbit* $\wedge$ S > G.RP,
     G.RP $\leftarrow$ S;
     recompute (∗,G).iif toward G.RP;
     if (∗,G).iif is changed, (∗,G).imode $\leftarrow$ *PJ*;
     send PIM-Join/Prune to *all-router-group* on (∗,G).iif
     with PIM-Join $\leftarrow$ {*WCbit*,*RPbit*,(∗,G)}, PIM-Prune $\leftarrow$ *null*;
     if flag $\neq$ *WCbit* $\wedge$ flag $\neq$ *RPbit* $\wedge$ (S,G).RPbit is set,
     clear (S,G).RPbit;
     recompute (S,G).iif toward S;
     if (S,G).iif is changed, (S,G).imode $\leftarrow$ *PJ*;
     send PIM-Join/Prune to *all-router-group* on (S,G).iif
     with PIM-Join $\leftarrow$ {(S,G)}, PIM-Prune $\leftarrow$ *null*;
     reset I.timer on (S,G).oif;
     if S = G.RP, $\forall$ (S,G) entry : reset I.timer on (S,G).oif;
     $\forall$ (flag,S,G) $\in$ PIM-Prune :
     if $\neg\exists$ (S,G) entry $\wedge$ flag = *RPbit*, initialize (S,G) negative cache entry;
     if $\exists$ (S,G) entry $\wedge$ I $\in$ (S,G).oif, delete I from (S,G).oif;
  4. On accepting PIM-Assert from neighbor N on subnetwork I
     $\forall$ (flag,S,G) $\in$ PIM-Assert :
     if $\exists$ (S,G) entry $\wedge$ I $\in$ (S,G).oif,
     if (S,G).RPbit is clear $\wedge$ (flag = *RPbit* $\vee$ N < *router-itself*), send PIM-Assert to N on I;
     else, delete I from (S,G).oif;
  5. On receiving PIM-Register of (S,G) piggybacked with data packet P
     G.RP $\leftarrow$ *router-itself*;
     if $\neg\exists$ (S,G) entry, initialize (S,G) entry;
     send PIM-Join/Prune to *all-router-group* on (S,G).iif
     with PIM-Join $\leftarrow$ {(S,G)}, PIM-Prune $\leftarrow$ *null*;
     $\forall$ subnetwork I $\in$ (S,G).oif : send P on I;

6. On receiving PIM-Register-Stop of (S,G) from rendezvous point RP
   stop sending PIM-Register to RP;
7. On receiving RP-Reachability of rendezvous point RP on subnetwork I
   if RP $>$ G.RP, G.RP $\leftarrow$ RP;
   recompute ($*$,G).iif toward G.RP;
   if ($*$,G).iif is changed, ($*$,G).imode $\leftarrow$ *PJ*;
   if I = ($*$,G).iif, reset ($*$,G).timer;
   $\forall$ subnetwork I $\in$ ($*$,G).oif : send RP-Reachability to *all-router-group* on I;
8. On receiving PIM-Mode on subnetwork I
   $\forall$ (mode,S,G) $\in$ PIM-Mode : if $\exists$ (S,G) entry, (S,G).imode $\leftarrow$ mode;

- Receiving Data Packets :
  1. On receiving data packet P of (S,G) from Neighbor N on subnetwork I
     if $\exists$ (S,G) entry $\wedge$ I = (S,G).iif,
     if (S,G).SPTbit is clear,
     set (S,G).SPTbit;
     if G.RP = *router-itself*, send PIM-Register-Stop to S;
     if $\exists$ ($*$,G) entry $\wedge$ (S,G).iif $\neq$ ($*$,G).iif,
     send PIM-Join/Prune to *all-router-group* on ($*$,G).iif
     with PIM-Join $\leftarrow$ *null*, PIM-Prune $\leftarrow$ {(*RPbit*,S,G)};
     reset (S,G).timer;
     $\forall$ subnetwork I $\in$ (S,G).oif : send P on I;
     else if ($\neg\exists$ (S,G) entry $\vee$ (S,G).SPTbit is clear) $\wedge$ $\exists$ ($*$,G) entry $\wedge$ I = ($*$,G).iif,
     $\forall$ subnetwork I $\in$ ($*$,G).oif : send P on I;
     else if ($\exists$ (S,G) entry $\wedge$ I $\in$ (S,G).oif) $\vee$ ($\exists$ ($*$,G) entry $\wedge$ I $\in$ ($*$,G).oif),
     send PIM-Assert to *all-router-group* on I;
  2. On receiving data packet P of (S,G) from host S on subnetwork I
     if $\exists$ (S,G) entry, reset (S,G).timer;
     $\forall$ subnetwork I $\in$ (S,G).oif : send P on I;
     else, encapsulate P in PIM-Register;
     $\forall$ rendezvous point RP of group G : send PIM-Register to RP;

- Switching between RP Tree and Shortest Path Tree :
  1. On switching from RP tree to shortest path tree of (S,G)
     initialize (S,G) entry;
     send PIM-Join/Prune to *all-router-group* on (S,G).iif
     with PIM-Join $\leftarrow$ {(S,G)}, PIM-Prune $\leftarrow$ *null*;
  2. On switching from shortest path tree of (S,G) to RP tree
     send PIM-Join/Prune to *all-router-group* on ($*$,G).iif
     with PIM-Join $\leftarrow$ {(*RPbit*,S,G)}, PIM-Prune $\leftarrow$ *null*;
     send PIM-Join/Prune to *all-router-group* on (S,G).iif
     with PIM-Join $\leftarrow$ *null*, PIM-Prune $\leftarrow$ {(S,G)};

- Switching between periodic join mode and implicit join mode :
  1. On switching from periodic join mode to implicit join mode of (S,G)
     (S,G).omode $\leftarrow$ *IJ*;
     $\forall$ subnetwork I : send PIM-Mode to *all-router-group* on I;
     if I $\in$ (S,G).oif, stop I.timer;
  2. On switching from implicit join mode to periodic join mode of (S,G)

(S,G).omode $\leftarrow$ *PJ*;
$\forall$ subnetwork I : send PIM-Mode to *all-router-group* on I;
if I $\in$ (S,G).oif, start I.timer;

- Detail Operations on (S,G) Entry, including ($*$,G) :
  1. On initializing (S,G) entry
     start (S,G).timer;
     if (S,G).RPbit is set, compute (S,G).iif toward G.RP;
     else, compute (S,G).iif toward S;
     if $\exists$ ($*$,G) entry, $\forall$ subnetwork I $\in$ ($*$,G).oif : add I to (S,G).oif;
     (S,G).imode $\leftarrow$ *PJ*;
     (S,G).omode $\leftarrow$ *PJ*;
  2. On removing (S,G) entry
     stop (S,G).timer;
  3. On adding subnetwork I to (S,G).oif
     if (S,G).oif = *null*, send PIM-Join/Prune to *all-router-group* on (S,G).iif
     with PIM-Join $\leftarrow$ {(S,G)}, PIM-Prune $\leftarrow$ *null*;
     start I.timer on (S,G).oif;
     if (S,G).WCbit is set, $\forall$ (S,G) entry : add I to (S,G).oif;
  4. On deleting subnetwork I from (S,G).oif
     stop I.timer on (S,G).oif;
     if (S,G).oif = *null*, send PIM-Join/Prune to *all-router-group* on (S,G).iif
     with PIM-Join $\leftarrow$ *null*, PIM-Prune $\leftarrow$ {(S,G)};

6 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.

This work was supported by the National Science Foundation, ARPA, cisco Systems and Sun Microsystems.

References

[1] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. Protocol independent multicast (pim) : Motivation and architecture. *Internet Draft*, March 1994.

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

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

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

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