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

Stephen Deering

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

Deborah Estrin

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

Dino Farinacci

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

Mark Handley

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

Ahmed Helmy

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

Van Jacobson

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

Ching-gung Liu

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

Puneet Sharma

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

David Thaler

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

Liming Wei

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

draft-ietf-idmr-PIM-SM-spec-02.ps

May 6, 1996

Status of This Memo

1 Introduction

This document describes a protocol for efficiently routing to multicast groups that may span wide-area (and inter-domain) internets. We refer to the approach as Protocol Independent Multicast–Sparse Mode (PIM-SM) because it is not dependent on any particular unicast routing protocol, and because it is designed to support sparse groups as defined in [1, 2]. This document describes the protocol details. For the motivation behind the design and a description of the architecture, see [1, 2].

Section 2 summarizes PIM-SM operation. It describes the protocol from a network perspective, in particular, how the participating routers interact to create and maintain the multicast distribution tree. Section 3 describes PIM-SM operations from the perspective of a single router implementing the protocol; this section constitutes the main body of the protocol specification. It is organized according to PIM-SM message type; for each message type we describe its contents, its generation, and its processing. Interoperability with other protocols will be further discussed in an appendix to this document.

Section 4 provides packet format details.

The most significant functional changes since the January '95 version, are the Rendezvous Point-related mechanisms and the removal of the PIM-DM protocol details to a separate document[3] (for clarity).

2 PIM-SM Protocol Overview

In this section we provide an overview of the architectural components of PIM-SM.

A router[1] receives explicit Join/Prune messages from those neighboring routers that have downstream group members. The router then forwards data packets addressed to a multicast group, G, only onto those interfaces on which explicit joins have been received.

A Designated Router (DR) sends periodic Join/Prune messages toward a group-specific Rendezvous Point (RP) for each group for which it has active members. Each router along the path toward the RP builds wildcard (any-source) forwarding state for the group and sends messages on toward the RP. The wildcard forwarding entry's incoming interface points toward the RP; the outgoing interfaces point to the neighboring downstream routers that have sent Join/Prune messages toward the RP. This forwarding state creates a shared, RP-centered, distribution tree that reaches all group members. When a data source first sends to a group, its DR unicasts Register messages to the RP with the source's data packets encapsulated within. If the data rate is high, the RP can send source-specific Join/Prune messages back towards the source and the source's data packets will follow the resulting forwarding state and travel unencapsulated to the RP. Whether they arrive encapsulated or natively, the RP forwards the source's decapsulated data packets down the RP-centered distribution tree toward group members. If the data rate warrants it, routers with local receivers can join a source-specific, shortest path, distribution tree, and prune these source's packets off of the shared RP-centered tree. Even if all receivers switch to the shortest path tree, state for that source will be kept at the RP, so that new members that join the RP-centered tree will receive data packets from the source. For low data rate sources, neither the RP, nor last hop routers need join a source-specific shortest path tree and data packets can be delivered via the shared, RP-tree.

The following subsections describe SM operation in more detail, in particular, the control messages, and the actions they trigger. Section 3 describes protocol operation from an implementors perspective, i.e., the actions performed by a single router.

[1] All routers mentioned in this document are assumed to be PIM-SM capable, unless otherwise specified.

Figure 1: Example: how a receiver joins, and sets up shared tree
Actions are numbered in the order they occur

2.1 Local hosts joining a group

In order to join a multicast group, G, a host sends an IGMP Host-Membership-Report message identifying the particular group. As specified in [4], IGMP Host-Membership-Report messages are sent in response to a directly-connected router's IGMP Host-Membership-Query message (see figure 1). [2] From this point on we refer to such a host as a receiver, R, (or member) of the group G.

When a DR receives an IGMP Host-Membership-Report for a new group, G, the DR looks up the associated RP. The DR (e.g., router A in figure 1) creates a wildcard multicast forwarding entry for the group, referred to here as a (*,G) entry; if there is no more specific match for a particular source, the packet will be forwarded according to this entry.

The RP address is included in a special field in the forwarding entry and is included in periodic upstream Join/Prune messages. The outgoing interface is set to that over which the IGMP Host-Membership-Report was received from the new member. The incoming interface is set to the interface used to send unicast packets to the RP. An RP-bit associated with this entry is also set, indicating that this entry, (*,G), represents state on the shared RP-tree. Each DR on the RP-tree with directly connected members sets a timer for this entry. If the timer expires and the DR has neither local members nor downstream receivers, the (*,G) state is deleted. If the DR does have local members, it refreshes the (*,G) entry timer each time it gets an IGMP Host-Membership-Report.

2.2 Establishing the RP-rooted shared tree

Triggered by the (*,G) state, the DR creates a Join/Prune message with the RP address in its join list and the WC-bit and RP-bit set; nothing is listed in its prune list. The RP-bit flags the join as being associated with the shared tree and therefore the Join/Prune message is propagated along the RP-tree. The WC-bit indicates that the address is an RP and the receiver expects to receive packets from all sources via this (shared tree) path.

Each upstream router creates or updates its multicast forwarding entry for (*,G) when it receives a

[2] All figures used in this section are for illustration and are not intended to be complete. For complete and detailed protocol action see Section 3.

Figure 2: Example: a host sending to a group
Actions are numbered in the order they occur

Join/Prune with the RP-bit and WC-bit set. The interface on which the Join/Prune message arrived is added to the list of outgoing interfaces (oifs) for (*,G). Based on this entry each upstream router between the receiver and the RP sends a Join/Prune message in which the join list includes the RP. The packet payload contains Multicast-Address=G, Join={RP,WCbit,RPbit}, Prune=NULL.

2.3 Hosts sending to a group

When a host first sends a multicast data packet to a group, its DR must deliver the packet to the RP for distribution down the RP-tree (see figure 2). This is done by the sender's DR unicasting a Register packet to the RP for the group. The data packet is encapsulated in the Register packet so that the RP can decapsulate it and deliver it to downstream members.

If the data rate of the source warrants[3] the use of a source-specific shortest path tree (SPT), the RP may construct a new multicast forwarding entry that is specific to the source, hereafter referred to as (S,G) state, and send periodic Join/Prune messages toward the source. The routers between the source and the RP build and maintain (S,G) state in response to these messages and send (S,G) messages upstream toward the source.

The source's DR must stop encapsulating data packets in Registers when (and so long as) it receives Register-Stop messages from the RP. The RP triggers Register-Stop messages in response to Registers, if the RP has no downstream receivers for the group (or for that particular source), or if the RP has already joined the (S,G) tree and is receiving the data packets natively.

2.4 Switching from shared tree (RP-tree) to shortest path tree (SP-tree)

When a router has directly-connected members, it first joins the shared RP-tree. The router can switch to a source's shortest path tree (SP-tree) after receiving packets from that source over the shared RP-tree. The recommended policy is to initiate the switch to the SP-tree after receiving a significant number of data packets during a specified time interval from a particular source. To realize this policy the router

[3] This decision is a local policy established at the RP. For example, when the Register rate exceeds a configured threshold at the RP, this may warrant the use of the SPT.

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

can monitor data packets from sources for which it has *no* source-specific multicast forwarding entry and initiate such an entry when the data rate exceeds the configured threshold. As shown in figure 3, router 'A' initiates a (S,G) state.

When a (S,G) entry is activated (and periodically so long as the state exists), a Join/Prune message is sent upstream towards the source, S, with S in the join list. The payload contains Multicast-Address=G, Join={S}, Prune=NULL. When the (S,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[4]. In this way when a data packet from S arrives and matches on this entry, all receivers will continue to receive the source's packets along this path. Note that (S,G) state must be maintained in all last-hop routers where an SP-tree is maintained. Even when (*,G) and (S,G) overlap, both states are needed to trigger the source-specific Join/Prune messages. (S,G) state is kept alive by data packets arriving from that source. A timer, S-timer, is set for the (S,G) entry and this timer is restarted whenever a data packet for (S,G) is forwarded out at least one oif. When the S-timer expires the state is deleted.

Only the RP and routers with local members can initiate switching to the SP-tree; intermediate routers do not. Consequently, last hop routers create (S,G) state in response to data packets from the source, S; whereas intermediate routers only create (S,G) state in response to Join/Prune messages from downstream that have S in the Join list[5].

The (S,G) entry is initialized with the SPT-bit cleared, indicating that the shortest path tree branch from S has not yet been setup completely, and the router can still accept packets from S that arrive on the (*,G) entry's iif.

When a router with a (S,G) entry and a cleared SPT-bit starts to receive packets from the new source S on the iif for the (S,G) entry, and that iif differs from the (*,G) entry's iif, the router sets the SPT-bit, and sends a Join/Prune message towards the RP, indicating that the router no longer wants to receive packets from S via the shared RP-tree. The Join/Prune message sent towards the RP includes S in the prune list, with the RP-bit set indicating that S's packets should not be forwarded down this branch of the shared tree. If the router receiving the Join/Prune message has (S,G) state (with or without the RPbit

[4]In more complicated scenarios, other entries in the router have to be considered. For details see Section 3.

[5]For example, to implement the policy that source-specific trees are only setup for high-data rate source, a last-hop router might not create a (S,G) entry until it has received m data packets from the source within some interval of n seconds.

set), it deletes the arriving interface from the (S,G) oif list. If the router has only (*,G) state, it creates an (S,G)RP-bit entry. The Join/Prune message payload contains Multicast-Address=G, Join=NULL, Prune={S,RPbit}.

If at a later time a new receiver joins the RP-tree, the negative cache state on the RP-tree must be eradicated to bring all sources' data packets down to the new receiver. Therefore, when a (*,G) Join arrives with a null prune list at a router that has any (S,G)RP-bit entries (which is causing it to send source-specific prunes toward the RP), all RP-bit state for that group has to be updated upstream of the router; so as to bring all sources' packets down to the new member. To accomplish this the router updates all existing (S,G)RP-bit entries; it adds to each (S,G)RP-bit entry's oif list the interface on which the (*,G) join arrived. The router also triggers a (*,G) join upstream to cause the same updating of RP-bit settings upstream and pull down all active sources' packets. If the arriving (*,G) join has some sources included in its prune list, then the corresponding (S,G)RP-bit entries are left unchanged (i.e., the RPbit remains set and no oif is added).

2.5 Steady state maintenance of distribution tree (i.e., router state)

In the steady state each router sends periodic Join/Prune messages for each active (S,G), (*,G) or (*,*,RP)[6] entry; the Join/Prune messages are sent to the RPF neighbor on the iif of the corresponding entry. These messages are sent periodically to capture state, topology, and membership changes. A Join/Prune message is also sent on an event-triggered basis each time a new forwarding entry is established for some new source (note that some damping function may be applied, e.g., a merge time). Join/Prune messages do not elicit any form of explicit acknowledgment; routers recover from lost packets using the periodic refresh mechanism.

2.6 Obtaining RP information

To obtain the RP information, all routers collect RP-Set messages. RP-Set messages are sent hop-by-hop within the domain; originating at the domain's bootstrap router (BSR). The BSR is elected dynamically within each domain. [7]

Routers then use the set of RPs to get the proper Group to RP mapping. Details are as follows:

2.6.1 Bootstrap Router

A (small) set of routers, within a domain, are configured as candidate bootstrap routers. Initially, each of these candidates includes its address in 'RP-set' messages. Through a simple election mechanism, a single bootstrap router (BSR) is elected for that domain (see Section 3.6).

2.6.2 Candidate RPs

A set of routers within a domain are configured as candidate RPs (C-RPs); typically these will be the same routers that are configured as C-BSRs. Candidate RPs periodically unicast Candidate-RP-Advertisement messages (C-RP-Advs) to the BSR of that domain. C-RP-Advs include the address of the advertising C-RP, as well as an optional group address and a mask length field, indicating the group prefix(es) for which the candidacy is advertised. The BSR then includes a set of these Candidate-RPs in the RP-Set messages, along with the corresponding group prefixes (see Section 3.6.2). RP-Set messages are periodically sent hop-by-hop throughout the domain.

---

[6] (*,*,RP) entry is introduced for interoperability, see Sections 2.10 and 6.

[7] A domain in this context is a multicast region in which routers implement PIM-SM. PIM-SM border routers are assumed to connect a domain to the rest of the internet.

2.6.3 Group to RP mapping

Routers receive and store RP-Set messages originated by the BSR. When a DR receives IGMP Host-Membership-Report (or a data packet) from a directly connected host, for a group for which it has no entry, the DR uses a hash function to map the pertinent group to one of the C-RPs whose Group-prefix includes the group (see Section 3.7). The DR then sends a Join/Prune message towards (or unicasts Registers to) that RP. [8]

2.6.4 Providing RP liveness

The RP-Set message indicates liveness of the RPs included therein; if an RP is included in the message, then it is tagged as 'up' at the routers, while RPs not included in the message are tagged as 'down' and removed from the list of RPs over which the hash algorithm acts. Each router continues to use the contents of the most recently received RP-set message until it receives a new RP-set message.

2.7 Multicast data packet processing

Data packets are processed in a manner similar to existing multicast schemes. A router first performs a longest match on the source and group address in the data packet. A (S,G) entry is matched first if one exists; a (*,G) entry is matched otherwise. If neither state exists, then a (*,*,RP) entry match is attempted as follows: the router hashes on G to identify the RP for group G, and looks for a (*,*,RP) entry that has this RP address associated with it. If none of the above exists, then the packet is dropped. If a state is matched, an incoming interface check (RPF check) is performed on the matching state and if it fails the packet is dropped, otherwise the packet is forwarded to all interfaces listed in the outgoing interface list.

Some special actions are needed to deliver packets continuously while switching from the shared to shortest-path tree. In particular, when a (S,G) entry is matched, incoming packets are forwarded as follows:

1. If the SPT-bit is set, then:
   (a) if the incoming interface is the same as a matching (S,G) iif, the packet is forwarded to the oif-list of (S,G).
   (b) if the incoming interface is different than a matching (S,G) iif , the packet is discarded.
2. If the SPT-bit is cleared, then:
   (a) if the incoming interface is the same as a matching (S,G) iif, the packet is forwarded to the oif-list of (S,G). In addition, the SPT bit is set for that entry if the incoming interface differs from the incoming interface of the (*,G) or (*,*,RP) entry.
   (b) if the incoming interface is different than a matching (S,G) iif, the incoming interface is tested against a matching (*,G) or (*,*,RP) entry. IF the iif is the same as one of those, the packet is forwarded to the oif-list of the matching entry.
   (c) Otherwise the iif does not match any entry for G and the packet is discarded.

Data packets never trigger prunes. However, data packets may trigger actions that in turn trigger prunes. For example, when router *B* in figure 3 decides to switch to SP-tree at step 3, it creates a (S,G) entry with SPT-bit set to 0. When data packets from S arrive at interface 2 of *B*, *B* sets the SPT-bit to 1 since the iif for (*,G) is different than that for (S,G). This triggers the sending of prunes towards the RP.

[8] Each intermediate router also uses this same hash function to determine the (*,*,RP) match for incoming data packets.

2.8 Operation over Multi-access Networks

This section describes a few additional protocol mechanisms needed to operate PIM over multi-access networks: Designated Router election, Assert messages to resolve parallel paths, and the Joiner bit to suppress redundant Joins on multi-access networks.

2.8.1 Designated router election

When there are multiple routers connected to a multi-access network, one of them should be chosen to operate as the designated router (DR) at any point in time. The DR is responsible for sending triggered Join/Prune and Register messages toward the RP[9].

A simple designated router (DR) election mechanism is used for both SM and traditional IP multicast routing.

Neighboring routers send Query messages to each other. The sender with the largest IP address assumes the role of DR. Each router connected to the multi-access LAN sends the Queries periodically in order to adapt to changes in router status.

2.8.2 Parallel paths to a source or the RP

If a router receives a multicast datagram on a multi-access LAN from a source whose corresponding (S,G) outgoing interface list includes the interface to that LAN, the packet must be a duplicate. In this case a single forwarder must be elected. Using Assert messages addressed to '224.0.0.13' (ALL-PIM-ROUTERS group) on the LAN, upstream routers can resolve which one will act as the forwarder. Downstream routers listen to the Asserts so they know which one was elected, and therefore where to send subsequent Joins. Typically this is the same as the downstream router's RPF neighbor but there are circumstances where this might not be the case, e.g., when using different unicast protocols.

The upstream router elected is the one that has the shortest distance to the source. Therefore, when a packet is received on an outgoing interface a router sends an Assert message on the multi-access LAN indicating what metric it uses to reach the source of the data packet. The router with the smallest numerical metric (with ties broken by highest address) will become the forwarder. All other upstream routers will delete the interface from their outgoing interface list. The downstream routers also do the comparison in case the forwarder is different than the RPF neighbor.

Associated with the metric is a metric preference value. This is provided to deal with the case where the upstream routers may run different unicast routing protocols. The numerically smaller metric preference is always preferred. The metric preference should be treated as the high-order part of an assert metric comparison. Therefore, a metric value can be compared with another metric value provided both metric preferences are the same. A metric preference can be assigned per unicast routing protocol and needs to be consistent for all routers on the multi-access network.

Asserts are also needed for (*,G) entries since there may be parallel paths from the RP and sources to a multi-access network. When an assert is sent for a (*,G) entry, the first bit in the metric preference (RP-bit) is always set to 1 to indicate that this path corresponds to the RP tree, and that the match should be done on (*,G) if exits. Furthermore, the RP-bit is always cleared for SP-tree entries' metric preference; this causes an SP-tree path to always look better than an RP-tree path. When the SP-tree and RPtree cross the same LAN, this mechanism eliminates the duplicates that would otherwise be carried over the LAN.

In case the packet, or the Assert message, matches on oif for (*,*,RP) entry, a (*,G) entry is created, and asserts take place as if the matching state were (*,G).

[9] IGMP Queries are sent by a PIMv2 DR if it supports IGMPv1. If a PIMv2 router is using IGMPv2 then Host queries are not sent by the PIMv2 DR but by the IGMP querier.

The DR may lose to another router on the LAN by the Assert process if there are multiple paths to the RP through the LAN. From then on, the DR is no longer the last-hop router for local receivers. The winning router becomes the last-hop router and is responsible for sending (*,G) join messages to the RP. Asserts are rate limited.

2.8.3 Join/Prune suppression

If a Join/Prune message arrives on the incoming interface for an existing (S,G) entry, and the sender of the Join/Prune has a higher IP address than the recipient of the message, a Joiner-bit in the multicast routing table entry is cleared to suppress further Join/Prune messages. A timer is set for the Joiner-bit; after it expires the Joiner-bit is set indicating further periodic Join/Prunes should be sent for this entry. The Joiner-bit timer is restarted each time a Join/Prune message is received from a higher-IP-addressed PIM neighbor.

2.9 Unicast Routing Changes

When unicast routing changes, an RPF check is done on all active (S,G), (*,G) and (*,*,RP) entries, 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 interface list. The previous incoming interface may be added to the outgoing interface list by a subsequent Join/Prune from downstream. Join/Prune messages received on the current incoming interface are ignored. Join/Prune messages 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/Prune messages. If the router has a (S,G) entry with the SPT-bit set, and the updated iif(S,G) does *not* differ from iif(*,G) or iif(*,*,RP), then the router resets the SPT-bit.

The router must send a Join/Prune message with S in the Join list out its new incoming interface to inform upstream routers that it expects multicast datagrams over the interface. It may also send a Join/Prune message with S in the Prune list out the old incoming interface, if the link is operational, to inform upstream routers that this part of the distribution tree is going away.

2.10 Interaction with dense mode protocols such as DVMRP

The essential problem in connecting to dense mode protocols is to pull all packets generated within the PIM-SM region down to the dense mode routers. To do this, a special entry type, referred to as (*,*,RP), is introduced. Every (*,*,RP) entry is associated with a particular RP in the domain; that RP is used to conduct RPF checks. Border routers initiate the building of (*,*,RP) towards all internal Candidate RPs. (*,*,RP) entries represent an aggregation of all the groups supported by the RP.

Most of the mechanisms needed to support interoperability with dense mode protocols such as DVMRP are implemented in BRs, i.e., special routers that sit at the boundary between a PIM-SM regions and the DVMRP regions and which speak both protocols. However, all PIM-SM routers must be capable of supporting (*,*,RP) state and interpreting associated Join messages. Interaction with non-PIM-SM networks will be discussed in a separate interoperability appendix.

2.11 PIM-SM for Inter-Domain Multicast

Future documents will address the use of PIM-SM as a backbone inter-domain multicast routing protocol. Design choices center primarily around the distribution and usage of RP information for wide area, inter-domain groups.

2.12 Security

Editors Note: This section requires further work.

All PIM control messages may use IPsec[5] to address security concerns.

3 Detailed Protocol Description

This section describes the protocol operations from the perspective of an individual router implementation. In particular, for each message type we describe how it is generated and processed.

3.1 Query

Query messages are sent so neighboring routers can discover each other.

3.1.1 Sending Queries

Query messages are sent periodically between PIM neighbors. By default they are transmitted every 30 seconds. This informs routers what interfaces have PIM neighbors. Query messages are multicast using address 224.0.0.13 (ALL-PIM-ROUTERS group). The packet includes the holdtime for neighbors to keep the information valid. The recommended holdtime is 3 times the query transmission interval. By default the holdtime is 90 seconds. Queries are sent on all types of communication links.

3.1.2 Receiving queries

When a router receives a Query packet, it stores the IP address for that neighbor, sets the PIM neighbor timer based on the Query holdtime, and determines the Designated Router (DR) for that interface. The highest IP addressed system is elected DR. Each query received causes the DR's address to be updated.

When a router that is the active DR receives a query from a new neighbor (i.e., from an IP address that is not yet in the DRs neighbor table), the DR unicasts its most recent RP-set information to the new neighbor.

3.1.3 Timing out neighbor entries

A periodic process is run to time out PIM neighbors that have not sent queries. If the DR has gone down, a new DR is chosen by scanning all neighbors on the interface and selecting the new DR to be the one with the highest IP address. If an interface has gone down, the router may optionally time out all PIM neighbors associated with the interface.

3.2 Join/Prune

Join/Prune messages are sent to join or prune a branch off of the multicast distribution tree. A single message contains both a join and prune list, either one of which may be null. Each list contains a set of source addresses, indicating the source-specific trees or shared tree that the router wants to join or prune.

3.2.1 Sending Join/Prune Messages

Join/Prune messages are merged such that a message sent to a particular upstream neighbor, N, includes all of the current joined and pruned sources that are reached via N; according to unicast routing Join/Prune messages are multicast to all routers on multi-access networks with the target address set to the next hop router towards S or RP. Join/Prune messages are sent periodically. Currently the period is set to 60 seconds.[10]

A router sends a periodic Join/Prune message to each distinct RPF neighbor associated with each (S,G), (*,G) and (*,*,RP) entry. Join/Prune messages are only sent if the RPF neighbor is a PIM

[10] In the future we will introduce mechanisms to rate-limit this control traffic on a hop by hop basis, in order to avoid excessive overhead on small links.

neighbor. A periodic Join/Prune message sent towards a particular RPF neighbor is constructed as follows:

1. Each router determines the RP for a (*,G) entry by using the hash function described. The RP address (with RP and WC bits set) is included in the join list of a periodic Join/Prune message under the following conditions:
   (a) The Join/Prune message is being sent to the RPF neighbor to the RP for an active (*,G) or (*,*,RP) entry, and
   (b) The outgoing interface list in the (*,G) or (*,*,RP) entry is non-NULL, or the router is the DR on the same interface as the RPF neighbor.
2. A particular source address, S, is included in the join list with the RP and WC bits cleared under the following conditions:
   (a) The Join/Prune message is being sent to the RPF neighbor to S, and
   (b) There exists an active (S,G) entry with the RPbit cleared, and
   (c) The *oif* list in the (S,G) entry is not null.
3. A particular source address, S, is included in the prune list with the RP and WC bits cleared under the following conditions:
   (a) The Join/Prune message is being sent to the RPF neighbor to S, and
   (b) There exists an active (S,G) entry with the RPbit cleared, and
   (c) The *oif* list in the (S,G) entry is null.
4. A particular source address, S, is included in the prune list with the RP bit *set* and the WC bit cleared under the following conditions:
   (a) The Join/Prune message is being sent to the RPF neighbor toward the RP and there exists a (S,G) entry with the RPbit set and null *oif* list, or
   (b) The Join/Prune message is being sent to the RPF neighbor toward the RP, there exists a (S,G) entry with the RPbit cleared and SPT-bit set, and the incoming interface toward S is different than the incoming interface toward the RP, or
   (c) The Join/Prune message is being sent to the RPF neighbor toward the RP, and there exists a (*,G) entry and (S,G) entry for a directly connected source.
5. The RP address (with RP and WC bits set) is included in the prune list if:
   (a) The Join/Prune message is being sent to the RPF neighbor toward the RP and there exists a (*,G) entry with a null *oif* list (see Section 3.5.2).

In addition to these periodic messages, the following events will trigger Join/Prune messages (the contents of triggered messages are the same as the periodic, described above)

1. Receipt of an IGMP Host-Membership-Report message for a group G will cause building or modifying corresponding (*,G) state, and subsequent triggering of upstream Join/Prune messages as follows:

   (a) If the receiving router does not have a forwarding entry for G the router creates a (*,G) entry, with the interface upon which the IGMP Host-Membership-Report was received included in the oif list. The router sends a Join/Prune message towards the RP with the RP address and RP-bit and WC-bits set in the join list. A timer is initiated for each interface in the oif list. Or,

   (b) If the (*,G) already exists, the interface upon which the IGMP Host-Membership-Report was received is added to the oif list (if it was not included already) and the timer for that interface is restarted.

2. Receipt of a Join/Prune message for (S,G), (*,G) or (*,*,RP) will cause building or modifying corresponding state, and subsequent triggering of upstream Join/Prune messages, in the following cases:

   (a) When there is no current forwarding entry, the RP address included in the Join/Prune message is checked against the local RP-Set information. If it matches, an entry will be created. If the router has no RP-Set information it may discard the message, or optionally use the RP address included in the message.

   The new entry will in turn trigger an upstream Join/Prune message.

   (b) When the outgoing interface list of (S,G) RPbit entry is null, the triggered Join/Prune message will contain S in the prune list.

3. Receipt of a packet on a (S,G) entry whose SPT-bit is cleared triggers the following if the packet arrived on the correct incoming interface and there is a (*,G) or (*,*,RP) entry with a different incoming RPF neighbor: a) setting of the SPT-bit on (S,G) entry, and b) sending a Join/Prune message towards the RP with S,RP-bit in the prune list if the iif of (S,G) is different from the iif of (*,G) or (*,*,RP).

4. When a Join/Prune message is received for a group G, the prune list is checked. If it contains a source for which the receiving router has a corresponding active (S,G), (*,G) or (*,*,RP) entry, and whose *iif* is that on which the Join/Prune was received, then a join for (S,G), (*,G) or (*,*,RP) is triggered to override the prune, respectively. (This is necessary in the case of parallel downstream routers connected to a multi-access network.)

5. When the RP fails, the RP will not be included in the RP-Set messages sent to the receivers' last-hop routers. This triggers the last-hop routers to send (*,G) joins towards the new RP for the group, as determined by the RP-Set and the hash function[11].

We do not trigger prunes onto interfaces for SM groups based on data packets. Data packets that arrive on the wrong incoming interface for an SM group are silently dropped.

It is possible that a Join/Prune message constructed according to the preceeding rules could exceed the MTU of a network. In this case, the message can undergo semantic fragmentation whereby information corresponding to different groups can be sent in different messages. However, if a Join/Prune message must be fragmented the following rule must be followed:

1. The complete prune list corresponding to a group G must be included in the same Join/Prune message as the associated RP-tree Join for G.

[11] PIM Multicast Border Routers (PMBRs), handling interoperability functionality, trigger (*,*,RP) joins towards each RP in the RP-Set.

3.2.2 Receiving Join/Prune Messages

When a router receives a Join/Prune message, it processes it as follows:

1. The receiver of the Join/Prune notes the interface on which the PIM message arrived, call it I. The router accepts this Join/Prune message if this Join/Prune message is addressed to the router itself. If the Join/Prune is for this router the following actions are taken:

   (a) If an address Sj in the join list has RP-bit and WC-bit set, then Sj is the RP address used by the downstream router and the following actions are taken:

      i. If Sj is not the same as the receiving router's RP mapping for G, the receiving router may ignore that group entry in the Join/Prune message. If the router does not have any RP-Set information, it may use the address Sj included in the Join/Prune message as the RP for the group.

      ii. If Sj is the same as the receiving router's RP mapping for G, it adds I to the outgoing interface list of the (*,G) forwarding entry and sets the timer for that interface (if there is no (*,G) entry, the router creates one first). If a (*,*,RP) exists, for the RP associated with G, then the oif list of the newly created (*,G) is copied from that (*,*,RP) state, excluding iif(*,G),

      iii. For each (Si,G) entry associated with group G, if Si is not included in the prune list, and if I is not the iif then interface I is added to the *oif* list and the timers are restarted for that interface in each affected entry. If the G in the join message is '*'[12], then every (*,G) and (S,G) entry, whose group address hashes to the RP indicated in the (*,*,RP) join message, is updated accordingly,

      iv. If the (Si,G) entry is an RP-bit entry and its *oif* list is the same as (*,G) *oif* list, then the (Si,G,RPbit) entry is deleted,

      v. 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.

   (b) For each address Si in the join list whose RP-bit and WC-bit are *not* set, and for which there is no existing (Si,G) forwarding entry, the router initiates one. [13]

      i. The outgoing interface for (Si,G) is set to I. The incoming interface for (Si,G) is set to the interface used to send unicast packets to Si (i.e., the RPF neighbor).

      ii. If the interface used to reach Si is the same as the outgoing interface being built, I, this represents an error and the Join/Prune should not be processed.

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

      i. If the RP-bit is not set for Si listed in the Join/Prune message, but the RP-bit is set on the existing (Si,G) entry, the router clears the RP-bit on (Si,G) entry, sets the incoming interface to point towards Si for that (Si,G) entry, and sends a Join/Prune to the new incoming interface; and

[12] A '*' in the group field of the Join/Prune is represented by a group address 224.0.0.0 and a group mask length of 4, indicating a (*,*,RP) Join.

[13] The router creates a (S,G) entry and copies all outgoing interfaces, excluding iif(S,G), from the (S,G)RP-bit, (*,G), or (*,*,RP), entry, if it exists. If a router does not copy all outgoing interfaces from the (*,G), or (*,*,RP) 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), or (*,*,RP), entry, and will be dropped because the incoming interface is incorrect.

   ii. The router adds I to the list of outgoing interfaces if I is not the same as the existing incoming interface; the timer for I is restarted.
   iii. The (Si,G) SPT bit is initialized to be cleared until data comes down the shortest path tree.

(d) For each address Si in the prune list, with the RP-bit is either set or cleared, and the WC-bit cleared:
   i. If there is an existing (Si,G) forwarding entry, the router schedules a deletion of I from the list of outgoing interfaces by lowering that oif timer to 5 seconds (unless it is already lower). The deletion is not executed until this timer expires, allowing for other downstream routers on a multi-access LAN to override the prune.
   ii. If the router has a current (*,G), or (*,*,RP), forwarding entry, and if a (Si,G)RP-bit entry also exists then the (Si,G)RP-bit entry is maintained even if its outgoing interface list is null.

(e) For any Si in the prune list that has the RP-bit set, and the WC-bit cleared:
   i. If (*,G), or corresponding (*,*,RP), state exists, but there is no (Si,G) entry, an (Si,G)RP-bit entry is created . The outgoing interface list is copied from the (*,G), or (*,*,RP), entry, with the interface, I, on which the prune was received deleted. Packets from the pruned source, Si, match on this state and are not forwarded toward the pruned receivers.
   ii. If there exists a (Si,G) entry, with or without the RPbit set, the iif on which the prune was received, I, is deleted from the *oif* list, and the entry timer is restarted.

(f) For each address Si in the prune list, with the RP-bit and the WC-bit set:
   i. If there is an existing (*,G) entry, with Si as the RP for G, the router schedules a deletion of I from the list of outgoing interfaces by lowering that oif timer to 5 seconds (unless it is already lower). The deletion is not executed until this timer expires, allowing for other downstream routers on a multi-access LAN to override the prune.
   ii. If the corresponding (*,*,RP) state exists, but there is no (*,G) entry, a (*,G) entry is created. The outgoing interface list is copied from (*,*,RP) entry, with the interface, I, on which the prune was received, deleted.
   iii. If there exists a (*,G) entry, the interface on which the prune was received, I, is deleted from the *oif* list, and the entry timer is restarted.

2. If the received Join/Prune does not indicate the router as its target, then if the Join/Prune is for a (S,G) pair for which the router has an active (S,G) entry, and if the Join/Prune arrived on the *iif* for that entry, then the router compares the IP address of the generator of the Join/Prune, to its own IP address.

   (a) If its own IP address is higher, the Joiner-bit in the (S,G) entry is set.

   (b) If its own IP address is lower, the Joiner-bit in the (S,G) entry is cleared, and the Joiner-bit timer is activated.

   After the timer expires the Joiner-bit is set indicating further periodic Join/Prunes should be sent for this entry. The Joiner-bit timer is restarted each time a Join/Prune message is received from a higher-IP-addressed PIM neighbor.

   For any new (S,G), (*,G) or (*,*,RP) entry created by an incoming Join/Prune message, the Joiner-bit is set and the SPT-bit is cleared.

3.3 Register and Register-Stop

When a source first starts sending to a group its packets are encapsulated in Register messages and sent to the RP. If the data rate warrants source-specific paths, the RP sets up source specific state and starts sending (S,G) Join/Prune messages toward the source.

3.3.1 Sending Registers and Receiving Register-Stops

Register messages are sent as follows:

1. When a DR receives a packet from a directly connected source, S[14]:
   (a) If there is no corresponding (S,G) entry, and the router has RP-Set information, the DR creates one with the Register-bit set to 1 and the RP address set according to the hash function mapping for the corresponding group. The Register-bit-timer is initialized to zero; the Register-bit-timer is non-zero only when the Register-bit is set to 0.
   (b) If there is a (S,G) entry in existence, the DR simply restarts the corresponding S-timer (entry timer).
2. If the new or previously-existing (S,G) entry has the Register-bit set, the data packet is encapsulated in a Register message and unicast to the RP for that group. The data packet is also forwarded according to (S,G) state in the DR if the oif list is not null; since a receiver may join the SP-tree while the DR is still registering to the RP.
3. If the (S,G) entry has the Register-bit cleared, the data packet is not sent in a Register message, it is just forwarded according to the (S,G) oif list.

The DR processes Register-Stop messages as follows:

1. The DR clears the Register-bit and restarts the Register-bit-timer in the corresponding (S,G) entry(ies).

When a Register-bit-timer expires, the corresponding entry(ies) Register-bit is set to 1 to reinstigate encapsulation of data packets in Register messages.

3.3.2 Receiving Register Messages and Sending Register-Stops

When a router (i.e., the RP) receives a Register message, the router does the following:

1. Decapsulates the data packet, and checks for a corresponding (S,G) entry.
   (a) If a (S,G) entry exists, the packet is forwarded but the SPT bit is left cleared (0). If the SPT bit is 1, the packet is dropped, and Register-Stop messages are triggered. Register-Stops are rate limited.[15]
   (b) If there is no (S,G) entry, but there is a (*,G) entry, or a (*,*,RP) entry with the RP corresponding to G, the packet is forwarded according to that entry.

[14]When a border router (e.g., a router that connects the PIM-SM region to a dense mode region running DVMRP or PIM-DM) receives a packet from a source in the dense mode region, the router treats the packet as if it were from a directly connected source. See the Appendix on Interoperability for more details.

[15]Register-Stops should be rate limited so that no more than a few are sent per round trip time. This prevents a high datarate stream of packets from triggering a large number of Register-stop messages between the time that the first packet is received and the time when the source receives the first Register-Stop.

(c) If there is a (*,*,RP) entry but no (*,G) entry, a (*,G) or (S,G) entry is created and the oif is copied from the (*,*,RP) entry to the new entry.

(d) If there is no G or (*,*,RP) entry corresponding to G, the packet is dropped, and a Register-Stop is triggered.

(e) A "Border bit" bit is added to the Register message, to facilitate interoperability mechanisms. PIM MBRs set this bit when registering for external sources (see Sections 2.10 and 6). If the "Border bit" is set in the Register, the RP does the following:

i. If there is no matching (S,G) state, the RP creates one, with a 'PMBR' field. This field holds the source of the Register (i.e. the outer IP address of the register packet). The RP triggers a (S,G) join towards the source of the data packet, and clears the SPT bit for the (S,G) entry, else

ii. If the 'PMBR' field for the corresponding (S,G) entry matches the source of the Register packet, the decapsulated packet is forwarded to the oif list of that entry, else

iii. The packet is dropped, and a Register-stop is triggered towards the source of the Register.

The (S,G) state timer is restarted by Registers arriving from that source to that group.

2. If the matching (S,G) or (*,G) state contains a null oif list, the RP unicasts a Register-Stop message to the source of the Register message; in the latter case, the source-address field, within the Register-Stop message, is set to the wildcard value (all 0's). This message is not processed by intermediate routers, hence no (S,G) state is constructed between the RP and the source.

3. If the Register message arrival rate warrants it and there is no existing (S,G) entry, the RP sets up a (S,G) forwarding entry with the outgoing interface list, excluding iif(S,G), copied from the (*,G) outgoing interface list, its SPT-bit is initialized to 0. If a (*,G) entry does not exist, but there exists a (*,*,RP) entry with the RP corresponding to G , the oif list for (S,G) is copied -excluding the iif- from that (*,*,RP) entry.

   A timer is set for the (S,G) entry and this timer is restarted by receipt of data packets for (S,G). The (S,G) entry causes the RP to send a Join/Prune message for the indicated group towards the source of the register message.

   If the (S,G) oif list becomes null, Join/Prune messages will not be sent towards the source, S.

3.4 Multicast Data Packet Forwarding

Processing a multicast data packet involves the following steps:

1. Lookup forwarding state based on a longest match [16] of the source address, and an exact match of the destination address in the data packet and compare the RPF check on the source address in the packet header with the *iif* specified in the forwarding entry.

2. If the packet arrived on the interface found in the matching-entry's *iif* field, and the *oif* list is not null:

   (a) Forward the packet to the *oif* list for that entry and restarted the entry's timer if the matching entry is (S,G)[17].

[16] The longest match is performed in the following order: (1) (S,G), (2) (*,G). If neither is matched, then a lookup is performed on (*,*,RP) entries.

[17] Optionally, the (S,G) timer may be restarted by periodic checking of the matching packet count.

(b) If the entry is a (S,G) entry with a cleared SPT-bit, and a (*,G) or associated (*,*,RP) also exists whose incoming interface is different than that for (S,G), set the SPT-bit for the (S,G) entry and trigger an (S,G) RP-bit prune towards the RP.

(c) If the source of the packet is a directly-connected host and the router is the DR on a multi-access network, check the Register-bit associated with the (S,G) entry. If it is set, then the router encapsulates the data packet in a register message and sends it to the RP.

This covers the common case of a packet arriving on the RPF interface to the source or RP and being forwarded to all joined branches. It also detects when packets arrive on the SP-tree, and triggers their pruning from the RP-tree. If it is the DR for the source, it sends data packets encapsulated in Registers to the RPs.

3. If the packet matches to an entry but did not arrive on the interface found in the entry's *iif* field, check the SPT-bit of the entry. If the SPT-bit is set, drop the packet. If the SPT-bit is cleared, then lookup the (*,G), or (*,*,RP), entry for G. If the packet arrived on the *iif* found in (*,G), or the corresponding (*,*,RP), forward the packet to the *oif* list of the matching entry. This covers the case when a data packet matches on a (S,G) entry for which the SP-tree has not yet been completely established upstream.

4. If the packet does not match to any entry, but the source of the data packet is a local, directly-connected host, and the router is the DR on a multi-access LAN and has RP-Set information, the DR uses the hash function to determine the RP associated with the destination group, G. The DR then checks the Register-bit associated with the local sender (if there is no such a Register-bit, a new register flag, associated with the local sender, is created and set), and encapsulates the data packet in a Register message and unicasts it to the RP.

5. If the packet does not match to any entry, and it is not a local host or the router is not the DR, drop the packet.

3.4.1 Data triggered switch to shortest path tree (SP-tree)

Different criteria can be applied to trigger switching over from the RP-based shared tree to source-specific, shortest path trees.

One proposed example is to do so based on data rate. For example, when a (*,G), or corresponding (*,*,RP), entry is created, a data rate counter may be initiated at the last-hop routers. The counter is incremented with every data packet received for directly connected members of an SM group, if the longest match is (*,G) or (*,*,RP). If and when the data rate for the group exceeds a certain configured threshold (t1), the router initiates 'source-specific' data rate counters for the following data packets. Then, each counter for a source, is incremented when packets matching on (*,G), or (*,*,RP), are received from that source. If the data rate from the particular source exceeds a configured threshold (t2), a (S,G) entry is created and a Join/Prune message is sent towards the source. If the RPF interface for (S,G) is *not* the same as that for (*,G) -or (*,*,RP), then the SPT-bit is cleared in the (S,G) entry.

Other configured rules may be enforced to cause or prevent establishment of (S,G) state.

3.5 Assert

Asserts are used to resolve which of the parallel routers connected to a multi-access LAN is responsible for forwarding packets onto the LAN.

3.5.1 Sending Asserts

The following Assert rules are provided when a multicast packet is received on an outgoing multi-access interface of an existing (S,G) entry:

1. Do unicast routing table lookup on source IP address from data packet, and send assert on interface for source IP address in data packet; include metric preference of routing protocol and metric from routing table lookup.

2. If route is not found, use metric preference of 0x7fffffff and metric 0xffffffff.

3. When an assert is sent for a (*,G) entry, the first bit in the metric preference (the RP-bit) is set to 1, indicating the data packet is routed down the RP-tree.

Asserts are rate-limited by the router.

3.5.2 Receiving Asserts

When an assert is received the router performs a longest match on the source and group address in the assert message. The router checks the first bit of the metric preference (RP-bit). If the RP-bit is set, the router does a match on (*,G), or (*,*,RP), entries, otherwise, the router matches (S,G) entries. If the matching entry is (*,*,RP), the router creates a (*,G) entry.

If the interface that received the Assert message is in the *oif* list of the matched entry, then this assert should be processed by this router as follows:

1. Compare the metric received in the Assert with the one the router would have advertised in an assert. The metric preference should be treated as the high-order part of an assert metric comparison. If the value in the assert is less than the router's value, delete the interface from the entry. If the value is the same, compare IP addresses, if the routers address is less than the assert sender, delete the interface.

2. If the router has won the election and there are directly connected members on the multi-access LAN, the router keeps the interface in its outgoing interface list. It acts as the forwarder for the LAN.

3. If the router won the election but there are no directly connected members on the multi-access LAN, the router schedules to delete the interface. The LAN might be a stub LAN with no members (and no downstream routers). If no subsequent Join/Prunes are received, the router deletes the interface from the outgoing interface list; otherwise it keeps the interface in its outgoing interface and acts as the forwarder for the LAN.

The winning router should send out an assert message including its own metric to that outgoing interface. This will cause other routers on the LAN to prune that interface from their forwarding entries.

Note that when an Assert is received, the router performs an exact match based on the source address, group address and the RP-bit of the metric preference in the assert message. This is not a longest match; only exact state will be matched. If there is no such state, then the router drops the Assert message. Otherwise, If the interface that received the Assert matches the incoming interface of the exactly matched entry, then the Assert message is processed as follows:

1. Downstream routers will select the upstream router with the smallest metric as their RPF neighbor. If two metrics are the same, the highest IP address is chosen to break the tie.[18]

2. If the downstream routers have downstream members, they must schedule a join to inform the upstream router that packets should be forwarded on the multi-access network. This will cause the upstream forwarder to cancel its scheduled deletion of the interface.

3.6 Candidate-RP-Advertisements and RP-Set messages

Candidate-RP-Advertisements (C-RP-Advs) are periodic PIM messages unicast by those routers that are configured as Candidate-RPs (C-RPs).

RP-Set messages are periodic PIM messages originated by the Bootstrap router (BSR) within a domain, and forwarded hop-by-hop to distribute the current RP-set to all routers in that domain. The RP-Set messages also support a simple mechanism by which the Candidate BSR (C-BSR) with the highest BSR-priority and IP address (referred to as the preferred BSR) is elected as the BSR for the domain[19].

3.6.1 Sending Candidate-RP-Advertisements

C-RPs periodically unicast C-RP-Advs to the BSR for that domain. The interval for sending these messages is subject to local configuration at the C-RP. A recommended default value is 60 seconds.

Candidate-RP-Advertisements carry group address and group mask fields. This enables the advertising router to limit the advertisement to certain prefixes or scopes of groups. The advertising router may enforce this scope acceptance when receiving Registers or Join/Prune messages.

3.6.2 Receiving C-RP-Advs and Originating RP-Set

Upon receiving a C-RP-Adv, a router does the following:

1. If the router is not the elected BSR, it ignores the message, else

2. The BSR adds the RP address to its local pool of candidate RPs, according to the associated group prefix(es) in the C-RP-Adv message[20]. The BSR may override the prefix indicated in a C-RP-Adv.

The BSR keeps an RP-timer per RP in its local RP-set. The RP-timer is initialized to three times the holdtime in the RP's C-RP-Adv. When the timer expires, the corresponding RP is removed from the RP-set. The RP-timer is restarted by the C-RP-Advs from the corresponding RP.

The BSR also keeps an RP-Set timer to send RP-Set messages periodically. In particular, when the RP-Set timer expires, the BSR originates an RP-Set message on each of its interfaces. The message is sent with a TTL of 1 to the 'ALL-PIM-ROUTERS' group. In steady state, the BSR originates RP-Set messages every 60 seconds. At startup, the RP-Set timer is initialized to 180 seconds, causing the first RP-Set message to be originated after 180 seconds, when/if the timer expires. For timer details see Section 3.6.3.

A DR unicasts an RP-Set message to new PIM neighbors starting up, after receiving their Query messages. (since after DR election the new neighbor may become the new DR.)

The RP-Set message is subdivided into sets of {group-prefix,RP-Count,RP-addresses}. The format of the RP-Set message allows 'semantic fragmentation', if the length of the original RP-Set message exceeds

[18] This is important so that downstream routers send subsequent Joins/Prunes (in SM) to the correct neighbor. An Assert timer is initiated when changing the RPF neighbor to the Assert winner. When the timer expires the router resets its RPF neighbor according to its unicast routing tables to capture failures of the Assert winner.

[19] We recommend that each router configured as a C-RP also be configured as a C-BSR.

[20] The BSR may apply a local policy to limit the number of Candidate RPs included in the RP-Set message.

the packet maximum boundaries (see Section 4). However, we recommend against configuring a large number of routers as C-RPs, to reduce the semantic fragmentation required.

3.6.3 Receiving and Forwarding RP-Set

Each router keeps an RP-Set timer, initialized to 180 seconds at startup.

When a router receives RP-Set message sent to 'ALL-PIM-ROUTERS' group, it performs the following:

1. If the message was not sent by the RPF neighbor towards the BSR address included, the message is dropped. Else
2. If the included BSR is **not** preferred over, and not equal to, the currently active BSR:
   (a) If the RP-Set timer is **not** yet expired, or if the receiving router is a C-BSR, then the RP-Set message is dropped. Else
   (b) The RP-Set timer **is** expired and the receiving router is not a C-BSR, so the receiving router stores the RP-Set and BSR address found in the message. The RP-Set message is then forwarded out all PIM interfaces, excluding the one over which the message arrived, to 'ALL-PIM-ROUTERS' group, with a TTL of 1.
3. If the RP-Set message includes a BSR address that **is** preferred over, or equal to, the currently active BSR, the router resets its RP-Set timer to 180 seconds, and stores the BSR address and RP-Set information. The RP-Set message is then forwarded out all PIM interfaces, excluding the one over which the message arrived, to 'ALL-PIM-ROUTERS' group, with a TTL of 1.

If the receiving router has no current RP set information and the RP-set was unicast to it from a directly connected neighbor, the router stores the information as its new RP-set. This covers the startup condition when a newly booted router obtains the RP-Set and BSR address from its DR.

When a router receives a new RP-Set it checks if each of the RPs referred to by existing state (i.e., by (*,G), (*,*,RP), or (S,G)RPbit entries) is in the new RP-Set. If an RP is not in the new RP-set, that RP is considered unreachable and the hash algorithm (see below) is re-performed for each group with locally active state that previously hashed to that RP. This will cause those groups to be distributed among the remaining RPs. When the new RP-Set contains a new RP, the value of the new RP is calculated for each group covered by that C-RP's Group-prefix. Any group for which the new RP's value is greater than the previously active RP's value is switched over to the new RP.

3.7 Hash Function

The hash function is used by all routers within a domain, to map a group to one of the C-RPs from the RP-Set. For a particular group, G, the hash function uses only those C-RPs whose Group-prefix covers G. The algorithm takes as input the group address, and the addresses of the Candidate RPs, and gives as output one RP address to be used.

The protocol requires that all routers hash to the same RP within a domain (except for transients). The following hash function must be used in each router:

1. For each candidate RP address C_i in the Candidate-RP-Set, whose Group-prefix covers G, compute a value:

$$Value(G,M,C_i) = 1103515245 \cdot ((1103515245 \cdot (G\&M) + 12345) \text{ XOR } C_i) + 12345 \qquad \text{mod } 2^{31}$$

where M is a hash-mask included in RP-Set messages. This hash-mask allows a small number of consecutive groups (e.g., 4) to always hash to the same RP. For instance, hierarchically-encoded data can be sent on consecutive group addresses to get the same delay and fate-sharing characteristics.

In standard C, this corresponds to:

```
srand(G & M);
srand(rand() ^ Ci);
value = rand();
```

2. The candidate with the highest resulting value is then chosen as the RP for that group, and its identity and hash value are stored with the entry created.

   Ties between C-RPs having the same hash value, are broken in advantage of the highest address.

The hash function algorithm is invoked by a DR, upon reception of a packet, or IGMP Host-Membership-Report, for a group, for which the DR has no entry. It is invoked by any router that has (*,*,RP) state when a packet is received for which there is no corresponding (S,G) or (*,G) entry. Furthermore, the hash function is invoked by all routers upon receiving a Join/Prune message with WC-bit set.

3.8 Processing Timer Events

Editors Note: Timers are also discussed individually in the sections that pertain to the protocol messages that they trigger/affect. Until we finalize this section, if discrepencies exist, then assume that the individual sections are authoritative over this table.

In this subsection, we enumerate all timers that have been discussed or implied. Since some critical timer events are not associated with the receipt or sending of messages, they are not fully covered by earlier subsections.

In many cases, the values for timers come from Holdtime fields in PIM control messages, in which case the default values used in these Holdtime fields are shown in the tables below. Otherwise, the default value used when setting the timer is shown. In general, the default timeout value for state information is three times the refresh period. For example, Queries refresh Neighbor state and the default Query-timer period is 30 seconds, so a default Neighbor-timer duration of 90 seconds is included in the Holdtime field of the Queries.

In this version of the spec we suggest particular numerical timer settings. A future version of the specification will specify a mechanism for timers to be set as a function of the outgoing link bandwidth.

Timers related to tree maintenance

Each (S,G), (*,G), and (*,*,RP) entry has multiple timers associated with it: one for each interface in the outgoing interface list, one for the multicast routing entry itself, and one for the Joiner-bit. Each (S,G) and (*,G) entry also has an Assert timer and an Assert-rate-limit timer. In addition, DR's have a Register-bit-timer for each (S,G) entry and every router has a single Join/Prune timer.

Because some of the outgoing interfaces in an (S,G) entry are copied from the (*,G) outgoing interface list, they may not have explicit (S,G) join messages from some of the downstream routers (i.e., where members are joining to the (*,G) tree only). Thus, when a timer is reset for an outgoing interface listed in a (*,G) entry, the timers are reset for that interface in each existing (S,G) entry whose oif list contains that interface[21]. The same rule applies to (*,G) and (S,G) entries when resetting an oif timer on a (*,*,RP) entry.

```
Timer               DefVal Notes

Joiner-bit              90 Started : When Joiner bit is cleared
per route entry            Reset by: Receiving Join from higher-IP neighbor on iif
                           Action  : Set Joiner bit

Join/Prune              60 Started : When booting
                           Reset by: Nothing
                           Action  : Send Join/Prune to each RPF neighbor, restart timer

oif                    180 Started : When adding oif to oiflist
per (*,*,RP) oif           Restarted by: Receiving (*,*,RP) Join on that iface
                           Action  : Remove oif from oiflist

oif                    180 Started : When adding oif to oiflist
per (*,G) oif              Restarted by: Receiving (*,G) Join or IGMP
                           Host-Membership-Report for G on that iface, or
                           restartedting oif timer in (*,*,RP).
                           Action  : Remove oif from oiflist

oif                    180 Started : When adding oif to oiflist
per (S,G) oif              Restarted by: Receiving (S,G) Join on that
                           iface, or restartedting oif timer in (*,G) or
                           (*,*,RP).
                           Action  : Remove oif from oiflist

(*,*,RP) entry         180 Started : When entry is created
per (*,*,RP)               Restarted by: Restartedting timer on any oif
                           Action  : Delete entry

(*,G) entry            180 Started : When entry is created
per (*,G)                  Restarted by: Receiving (*,G) prune,
                           restarting timer on any oif, or receiving an
                           Assert with RP-bit set.
                           Action  : Delete entry and any associated
                           (S,G)RP-bit entries

(S,G) entry            180 Started : When entry is created
aka S-timer                Restarted by: Forwarding data packet,
per (S,G)                  receiving Register, receiving (S,G) RP-bit
                           prune, restarting timer on any oif,
```

[21]If there are sources in the prune list of the (*,G) join, then the timers for the arriving interface will first be reset for those sources, and then this interface will be deleted from these same entries; producing a correct result, even though the updating of the timers was unnecessary. An implementation could optimize this by checking the prune list before processing the join list.

```
                          or receiving an Assert without RP-bit set.
                          Action  : Delete entry

Register-bit          60  Started : When Register bit is cleared by
per (S,G)                 receiving a Register-Stop
                          Restarted by: Receiving Register-Stop
                          Action  : Set Register bit

Assert               180  Started : Receiving an Assert where the
per (S,G)                 upstream RPF neighbor is not your unicast RPF
and (*,G)                 neighbor.
                          Restarted by: Receiving an Assert where the
                          upstream RPF neighbor is not your unicast
                          RPF neighbor.
                          Action  : Change RPF neighbor to unicast RPF neighbor

Assert-Rate-limit      5  Started : When an Assert is sent
per (S,G)                 Restarted by: Nothing
and (*,G)                 Action  : Allow asserts to be triggered by
                          data packets
```

Timers relating to neighbor discovery

```
Timer               DefVal Notes

Query                 30  Started : When booting
                          Restarted by: Nothing
                          Action  : Send Query on all ifaces, restart timer

Neighbor              90  Started : When receive first Query from neighbor
per neighbor              Restarted by: When receive subsequent Queries
                          Action  : Delete neighbor entry
```

Timers relating to RP information

```
Timer               DefVal Notes

C-RP-Adv              60  Started : When booting if you're a Cand-RP
                          Restarted by: Nothing
                          Action  : Send C-RP-Adv, restart C-RP-Adv timer

RP                   180  Started : When adding an RP to the RP-Set if
per RP                    you are BSR
                          Restarted by: Receiving C-RP-Adv
                          Action  : Remove RP from RP-Set

RP-Set            180/60  Started : Set to 180 when booting if
                          you're a C-BSR
```

```
Restarted by: Restarted to 180 when receive
RP-Set from preferred router if you're a C-BSR
Action  : Send RP-Set and restart  timer to 60 secs
```

3.9 Summary of flags used

Following is a summary of all the flags used in our scheme.

Bit	Used in	Definition
Border	Register	Register is coming from a PIM border router.
Joiner	Route entry	Periodic Join/Prunes should be sent for this entry.
Register	(S,G) entry	Encapsulate packets from directly connected sources in Register messages unicast to the RP for that group.
RP	Route entry	Entry represents state on the RP-tree.
RP	Join/Prune	Join is associated with the shared tree and therefore the Join/Prune message is propagated along the RP-tree.
RP	Assert	The data packet was routed down the shared tree; thus, the path indicated corresponds to the RP tree.
SPT	(S,G) entry	Packets have arrived on the iif towards S, and the iif is different from the (*,G) iif.
WC	Join	Included address is an RP and the receiver expects to receive packets from all sources via this (shared tree) path. Thus, the Join/Prune applies to a (*,G) entry.
WC	Route entry	Wildcard entry; if there is no more specific match for a particular source, packets will be forwarded according to this entry.

3.10 Security

Editors Note: this section is to be completed.

All PIM control messages may use IPsec[5] to address security concerns.

4 Packet Formats

This section describes the details of the packet formats for PIM control messages.

All PIM control messages have protocol number 103.

Basically, PIM messages are either unicast (e.g. Registers and Register-Stop), or multicast hop-by-hop to 'ALL-PIM-ROUTERS' group '224.0.0.13' (e.g. Join/Prune, Asserts, etc.).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Ver PIM Version number is 2.

Type Types for specific PIM messages. PIM Types are:

0 = Query
1 = Register
2 = Register-Stop
3 = Join/Prune
4 = RP-Set
5 = Assert
6 = Graft (used in PIM-DM only)
7 = Graft-Ack (used in PIM-DM only)
8 = Candidate-RP-Advertisement

Addr length Address length in bytes. Throughout this section this would indicate the number of bytes in the Address field of an address, including unicast and group addresses.

Checksum The checksum is the 16-bit one's complement of the one's complement sum of the entire PIM message, (excluding the data portion in the Register message). For computing the checksum, the checksum field is zeroed.

4.1 Encoded Source and Group Address formats

1. Unicast address: Only the address is included. The length of the unicast address in bytes is specified in the 'Addr length' field in the header.

2. Encoded-Group-Address: Takes the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Reserved  |  Mask Len     | Group multicast Address ...   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...Group multicast Address ...|
+-+-+-+-+-+-+-+-+-+-+-+~+~+~+~+~+~+
```

   Reserved Transmitted as zero. Ignored upon receipt.

   Mask Len The Mask length is 8 bits. The value is the number of contiguous bits left justified used as a mask which describes the address. It is less than or equal to Addr length * 8. If the message is sent for a single group then the Mask length should equal Addr length * 8 (i.e. 32 for IPv4 and 128 for IPv6).

   Group multicast Address contains the group address, and has number of bytes equal to that specified in the Addr length field.

3. Encoded-Source-Address: Takes the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Rsrvd   |S|W|R|  Mask Len     | Source Address ...            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ...  Source Address          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+~+~+-+
```

   Reserved Transmitted as zero, ignored on receipt.

   S,W,R See Section 4.5 for details.

   Mask Length Mask length is 8 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. If the message is sent for a single source then the Mask length should equal Addr length * 8. In version 2 of PIM, it is strongly recommended that this field be set to 32 for IPv4.

   Source Address The address length is indicated from the Addr length field at the beginning of the header. For IPv4, the address length is 4 octets.

4.2 Query Message

It is sent periodically by routers on all interfaces.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |           Holdtime            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Type, Addr length, Checksum Described above.

Reserved Transmitted as zero, ignored on receipt.

Holdtime The amount of time a receiver should keep the neighbor reachable, in seconds.

4.3 Register Message

It is sent by the Designated Router (DR) to the RP when a multicast packet needs to be transmitted on the RP-tree. Source IP address is set to the address of the DR, destination IP address is to the RP's address.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|B|                         Reserved                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       Multicast data packet                   ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Type, Addr length, Checksum Described above. *Note that the checksum for Registers is done only on the PIM header, excluding the data packet portion.*

B The Border bit. Set to zero by all DRs. Set to '1' by the PIM Multicast Border Routers, when registering for external sources.

Multicast data packet The original packet sent by the source.

4.4 Register-Stop Message

A Register-Stop is unicast from the RP to the sender of the Register message. Source IP address is the address to which the register was addressed. Destination IP address is the source address of the register message.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Encoded-Group Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Unicast-Source Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Type, Addr length, Checksum Described above.

Encoded-Group Address Format described above. Note that for Register-Stops the Mask Len field should contain Addr length * 8 (32 for IPv4), if the message is sent for a single group.

Unicast-Source Address IP host address of source from multicast data packet in register. The length of this field in bytes is specified in the Addr length field. A special wild card value (0.0.0.0), can be used to indicate any source.

4.5 Join/Prune Message

It is sent by routers towards upstream sources and RPs. A join creates forwarding state and a prune destroys forwarding state. Joins are sent to build shared trees (RP trees) or source trees (SPT). Prunes are sent to prune source trees when members leave groups as well as sources that do not use the shared tree.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Unicast-Upstream Neighbor Address                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Reserved     | Num groups    |          Holdtime             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Encoded-Multicast Group Address-1                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Number of Joined  Sources   |   Number of Pruned Sources    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Encoded-Joined Source Address-1                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             .                                 |
|                             .                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Encoded-Joined Source Address-n                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Encoded-Pruned Source Address-1                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             .                                 |
|                             .                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Encoded-Pruned Source Address-n                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           .                                   |
|                           .                                   |
|                           .                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Encoded-Multicast Group Address-n              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Number of Joined  Sources   |   Number of Pruned Sources    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Encoded-Joined Source Address-1                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             .                                 |
|                             .                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Encoded-Joined Source Address-n                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|               Encoded-Pruned Source Address-1                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              .                                |
|                              .                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Encoded-Pruned Source Address-n                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Type, Addr length, Checksum Described above.

Upstream Neighbor Address The IP address of the RPF or upstream neighbor.

Reserved Transmitted as zero, ignored on receipt.

Holdtime The amount of time a receiver should keep the Join/Prune state alive, in seconds.

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

Encoded-Multicast group address For format description see Section 4.1. A wild card group in the (*,*,RP) join is represented by a 224.0.0.0 in the group address field and '4' in the mask length field. A (*,*,RP) join also has the WC-bit and the RP-bit set.

Number of Joined Sources Number of join source addresses listed for a given group.

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.

See format section 4.1. The fields explanation for the Encoded-Source-Address format follows:

Reserved Described above.

S The Sparse bit is a 1 bit value, set to 1 for PIM-SM. It is used for PIM v.1 compatability.

W The WC bit is a 1 bit value. If 1, the join or prune applies to the (*,G) or (*,*,RP) entry. If 0, the join or prune applies to the (S,G) entry where S is Source Address. Joins and prunes sent towards the RP should have this bit set.

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

Mask Length, Source Address Described above.

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 :

$$< 0 >< 0 >< 16 >< 192.1.0.0 >$$

Number of Pruned Sources Number of prune source addresses listed for a group.

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[22].

[22] If the Join/Prune message boundary exceeds the maximum packet size, then the join and prune lists for the same group must be included in the same packet.

4.6 RP-Set

The RP-Set messages are multicast to 'ALL-PIM-ROUTERS' group, out all interfaces having PIM neighbors (excluding the one over which the message was received). RP-Set messages are sent with TTL value of 1. RP-Set messages originate at the BSR, and are forwarded by intermediate routers.

RP-Set message is divided up into 'semantic fragments', if the original message exceeds the maximum packet size boundaries.

The semantics of a single 'fragment' is given below:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Fragment Tag          | Hash Mask len | BSR-priority  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Unicast-BSR-Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Encoded-Group Address-1               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RP-Count-1    | Frag RP-Cnt-1 |         Reserved              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Unicast-RP-Address-1                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Unicast-RP-Address-m                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Encoded-Group Address-n               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RP-Count-m    | Frag RP-Cnt-m |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Unicast-RP-Address-1                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Unicast-RP-Address-m                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Type, Addr length, Checksum Described above.

Fragment Tag A randomly generated number, acts to distinguish the fragments belonging to different RP-Set messages; fragments belonging to same RP-Set message carry the same 'Fragment Tag'.

Hash Mask len The length (in bits) of the mask to use in the hash function. For IPv4 we recommend a value of 30. For IPv6 we recommend a value of 126.

BSR-priority Contains the BSR priority value of the included BSR. This field is considered as a high order byte when comparing BSR addresses.

Unicast-BSR-Address The IP address of the bootstrap router for the domain. The length of this field in bytes is specified in Addr length.

Encoded-Group Address-1..n The group prefix (address and mask) with which the Candidate RPs are associated. Format previously described.

RP-Count-1..n The number of Candidate RP addresses included in the whole RP-Set message for the corresponding group prefix[23].

Frag RP-Cnt-1..m The number of Candidate RP addresses included in this fragment of the RP-Set message, for the corresponding group prefix. The 'Frag RP-Cnt' field facilitates parsing of the RP-Set for a given group prefix, when carried over more than one fragment.

Unicast-RP-address-1..m The address of the Candidate RPs, for the corresponding group prefix. The length of this field in bytes is specified in Addr length.

[23]A router does *not* replace its old RP-Set for a given group prefix until/unless it receives 'RP-Count' addresses for that prefix; the addresses could be carried over several fragments. If only part of the RP-Set for a given group prefix was received, the router discards it, without updating that specific group prefix's RP-Set.

4.7 Assert Message

The Assert message is sent when a multicast data packet is received on an outgoing interface corresponding to the (S,G) or (*,G) associated with the source.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Encoded-Group Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Unicast-Source Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|                         Metric Preference                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Metric                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Type, Addr length, Checksum Described above.

Encoded-Group Address The group address to which the data packet was addressed, and which triggered the Assert. Format previously described.

Unicast-Source Address Source IP address from IP multicast datagram that triggered the Assert packet to be sent. The length of this field in bytes is specified in Addr length.

R RP bit is a 1 bit value. If the IP multicast datagram that triggered the Assert packet is routed down the RP tree, then the RP bit is 1; if the IP multicast datagram is routed down the SPT, it is 0.

Metric Preference Preference value assigned to the unicast routing protocol that provided the route to Host address.

Metric The unicast routing table metric. The metric is in units applicable to the unicast routing protocol used.

4.8 Graft Message

Used in dense-mode. Refer to PIM dense mode specification.

4.9 Graft-Ack Message

Used in dense-mode. Refer to PIM dense mode specification.

4.10 Candidate-RP-Advertisement

Candidate-RP-Advertisements are periodically unicast from the C-RPs to the BSR.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PIM Ver| Type  | Addr length   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix-Cnt    | Reserved      |             Holdtime          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Unicast-RP-Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Encoded-Group Address-1               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               .                               |
|                               .                               |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Encoded-Group Address-n               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PIM Version, Type, Addr length, Checksum Described above.

Prefix-Cnt The number of encoded group addresses included in the message; indicating the group prefixes for which the C-RP is advertising. A Prefix-Cnt of '0' implies a prefix of 224.0.0.0 with mask length of 4; i.e. all multicast groups. If the C-RP is not configured with Group-prefix information, the C-RP puts a default value of '0' in this field.

Holdtime The amount of time the advertisement is valid. This field allows advertisements to be aged out.

Unicast-RP-Address The address of the interface to advertise as a Candidate RP. The length of this field in bytes is specified in Addr length.

Encoded-Group Address-1..n The group prefixes for which the C-RP is advertising. Format previously described.

5 Appendix I: Changes and Updates to the Spec

This appendix populates the major changes in the specification document as compared to 'draft-ietf-idmr-pim-spec-01.{ps,txt}'.

5.1 Major Changes

List of changes since March '96 IETF:

1. (*,*,RP) Joins state and data forwarding check; replaces (*,G-Prefix) Joins state for interoperability. (*,G) negative cache introduced for the (*,*,RP) state supporting mechanisms.
2. Semantic fragmentation for the RP-Set message.
3. Appendix II on interoperability details with DVMRP in preparation.

List of changes incurred since version 1 of the spec.:

1. Proposal and refinement of bootstrap router (BSR) election mechanisms
2. Introduction of hash functions for Group to RP mapping
3. New RP-liveness indication mechanisms based upon the the Bootstrap Router (BSR) and the RP-Set messages.
4. Removal of reachability messages, RP reports and multiple RPs per group.

5.2 Packet Format Changes

Packet Format incurred updates to accommodate different address lengths, and address aggregation.

1. The 'Addr length' field was added to the PIM fixed header to specify the address length in bytes of the underlying protocol, see section 4.
2. The Encoded source and group address formats were introduced, with the use of a 'Mask length' field to allow aggregation, section 4.1.
3. Packet formats are no longer IGMP messages; rather PIM messages.

PIM message types and formats were also modified:
[*Note: most changes were made to the May 95 version, unless otherwise specified*].

1. Obsolete messages:
   (a) Register-Ack [Feb. 96]
   (b) Poll and Poll Response [Feb. 96]
   (c) RP-Reachability [Feb. 96]
   (d) RPlist-Mapping [Feb. 96]
2. New messages:
   (a) Candidate-RP-Advertisement [change made in October 95]

   (b) RP-Set [Feb. 96]

3. Modified messages:
   (a) Join/Prune [Feb. 96]
   (b) Register [Feb. 96]
   (c) Register-Stop [Feb. 96]

6 Appendix II: Interoperability with Dense Mode Protocols

Editors Note: This section is to be completed.

7 Acknowledgments

Tony Ballardie, Scott Brim, Jon Crowcroft, Bill Fenner, Paul Francis, Joel Halpern, Horst Hodel, Polly Huang, Stephen Ostrowski, and Lixia Zhang provided detailed comments on previous drafts. The authors of CBT[6] 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, L. Wei, P. Sharma, and A. Helmy. Protocol independent multicast (pim) : Motivation and architecture. *Internet Draft*, May 1995.

[2] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The pim architecture for wide-area multicast routing. *ACM Transactions on Networks*, April 1996.

[3] D. Estrin, D. Farinacci, V. Jacobson, C. Liu, L. Wei, P. Sharma, and A. Helmy. Protocol independent multicast-dense mode (pim-dm) : Protocol specification. *Internet Draft*, November 1995.

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

[5] R. Atkinson. Security architecture for the internet protocol, August 1995. RFC-1825.

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