Protocol Independent Multicast (PIM): Motivation and Architecture

Stephen Deering

Xerox PARC
3333 Coyoty 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

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

Liming Wei

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

draft-ietf-idmr-pim-arch-01.ps

January 11, 1995

Status of This Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. (Note that other groups may also distribute working documents as Internet Drafts).

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

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

Editors Note

This document has been modified significantly since the March, 1994 version. In particular we:

- Integrated Dense Mode PIM description and explanation of SM/DM PIM interaction.
- Extended discussion of PIM/non-PIM interaction.
- Revised and extended discussion of scaling in terms of state and control traffic.

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

Abstract

Existing multicast routing mechanisms were intended for use within regions where a group is widely represented or bandwidth is universally plentiful. When group members, and senders to those group members, are distributed *sparsely* across a wide area, these schemes are not efficient; data packets or membership report information are periodically sent over many links that do *not* lead to receivers or senders, respectively. This characteristic lead us to develop a multicast routing architecture that efficiently establishes distribution trees across wide-area internets, where many groups will be sparsely represented and where bandwidth is not uniformly plentiful due to the distances and multiple administrations traversed. Efficiency is evaluated in terms of the state, control message processing, and data packet processing required across the entire network in order to deliver data packets to the members of the group. The architecture also includes a more traditional, dense mode of operation for use within campus networks or other regions characterized by plentiful bandwidth.

The Protocol Independent Multicast (PIM) architecture:

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

(b) can be configured to adapt to different multicast group and network characteristics;

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

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

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

This document motivates and describes the PIM architecture. A companion document describes the protocol mechanisms for both sparse and dense modes PIM [1].

1 Introduction

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

The most significant innovation in this architecture is the efficient support of sparse, wide area groups. This *sparse mode* (SM) of operation complements the traditional *dense-mode* approach to multicast routing for campus networks, as developed by Deering [2, 3] and implemented previously in MOSPF and DVMRP [4, 5]. These traditional dense mode multicast schemes were intended for use within regions where a group is widely represented or bandwidth is universally plentiful. However, when group members, and senders to those group members, are distributed *sparsely* across a wide area, these schemes are not efficient; data packets (in the case of DVMRP) or membership report information (in the case of MOSPF) are occasionally sent over many links that do *not* lead to receivers or senders, respectively. The purpose of this work is to develop a multicast routing architecture that efficiently establishes distribution trees even when some or all members are sparsely distributed. Efficiency is evaluated in terms of the state, control message, and data packet overhead required across the entire network in order to deliver data packets to the members of the group.

1.1 Definition of terms

Asserts The process of choosing which router will forward a multicast packet from a particular source on to a LAN segment when there are multiple routers on that LAN segment with routes to the source.

Dense Groups Group membership that is plentiful within a region of an internet.

Dense-mode (DM) A generic term referring to a multicast routing protocol that is optimized for dense groups. Dense-mode PIM is an example of such a protocol.

Designated Router (DR) The highest IP addressed router on a multi-access network becomes the DR. It is responsible for sending IGMP Query packets to the LAN; and for sending PIM Register packets and PIM Join packets towards the RP. DRs need to know the (list of) RP address(es) for a sparse-mode group.

Grafts Grafts are used by new members to add themselves onto an existing distribution tree when a system becomes a member of a group. Grafts are used to undo pruned tree branches and are sent towards known sources for dense-mode groups. They are acknowledged hop-by-hop with Graft-Ack packets.

Joins Joins are sent toward the RP or toward a source, to create or refresh a branch of a multicast distribution tree. Joins are transmitted periodically.

Last-hop Routers A last-hop router is one that is directly connected to members of a group. Last-hop routers need to know the (list of) RP address(es) for a sparse-mode group.

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

Prunes Prunes are sent toward a source by a router when it wishes to leave the distribution tree. In sparse-mode, prunes are also sent toward the RP when a router switches from the RP tree to the source-rooted shortest path tree.

Rendezvous Point (RP) An RP is used for groups that operate in sparse-mode. It is a router that rendezvous' receivers and senders for a group. There may be more than one RP per group. A router may be the RP for multiple groups.

Registers A method for new sources to be known to the RP and for existing sources to learn about new receivers. Registers start the process to create multicast routing state in routers between the source and RP.

Reverse Path Forwarding (RPF) The algorithm used to provide loop-free delivery of multicast datagrams on a distribution tree. The RPF interface is the expected interface to receive multicast packets from a source. The RPF interface is also the interface used to send unicast packets to the source.

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

(S,G) state Pronounced "S comma G", it is the multicast routing table state a router has for a shortest-path tree rooted at the source. The incoming interface for this entry is the RPF interface to S. There is a (S,G) for each source sending to each group.

(S,*) state Pronounced "S comma star", it is a multicast routing table state a router has for each source sending to any group. The incoming interface for this entry is the RPF interface to S.

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

Shortest-Path Tree (SPT) A shortest-path tree rooted from the source is know as the SPT. Each source sending to a group has a distinct tree.

Sparse Groups Group membership that is spread out across regions of an internet. This does not imply that the group has a few number of members.

Sparse-mode (SM) A generic term referring to a multicast protocol that is optimized for sparse groups. Sparse-mode PIM and CBT are examples of such protocols.

(*,G) state Pronounced "star comma G", it is the multicast routing table state a router has for the RP tree. The incoming interface for this entry is the RPF interface to the RP for sparse-mode groups. There is one (*,G) for each group.

1.2 Background

In the traditional dense-mode IP multicast model, established by Deering [3], a *multicast address* is assigned to the collection of receivers for a multicast group. Senders simply use that address as the destination address of a packet to reach all members of the group. The separation of senders and receivers allows any host, member or non-member, to send to a group. A group membership protocol [6] is used for routers to learn the existence of members on their directly attached subnetworks. This receiver-initiated join procedure has very good scaling properties; as the group grows, it becomes more likely that a new receiver will be able to splice onto a nearby branch of the distribution tree. A multicast routing protocol, in the form of an extension to existing unicast protocols (e.g. DVMRP, an extension to a RIP-like distance-vector unicast protocol; or MOSPF, an extension to the link-state unicast protocol OSPF), is executed on routers to construct multicast packet delivery paths and to accomplish multicast data packet forwarding.

In the case of link-state protocols, changes of group membership on a subnetwork are detected by one of the routers directly attached to that subnetwork, and that router broadcasts the information to all other routers in the same routing domain [7]. Each router maintains an up-to-date image of the domain's topology through the unicast link-state routing protocol. Upon receiving a multicast data packet, the router uses the topology information and the group membership information to determine the shortest-path tree (SPT) from the packet's source subnetwork to its destination group members. Broadcasting of membership information is one major factor preventing link-state multicast from scaling to larger, wide-area, networks — every router must receive and store membership information for every group in the domain. The other major factor is the processing cost of the Dijkstra shortest-path-tree calculations performed to compute the delivery trees for all active multicast sources [8] for all groups, thus limiting its applicability on an internet-wide basis.

Distance-vector multicast routing protocols construct multicast distribution trees using variants of Reverse Path Forwarding (RPF) [9]. When the first data packet is sent to a group from a particular source subnetwork, and a router receiving this packet has no knowledge about the group, the router forwards the incoming packet out all interfaces except the incoming interface.[1] A special mechanism is used to avoid forwarding of data packets to leaf subnetworks with no members in that group (also known as truncated broadcasting). Also if the arriving data packet does not come through the interface that the router uses to send packets to the source of the data packet, the data packet is silently dropped; thus the term Reverse Path Forwarding [9]. When a router attached to a leaf subnetwork, receives a data packet addressed to a new group, if it finds no members present on its attached subnetworks, it will send a prune message upstream towards the source of the data packet. The prune messages prune the tree branches not leading to group members, thus resulting in a source-specific shortest-path tree with all leaves having members. Pruned branches will "grow back" after a time-out period; these branches will again be pruned if there are still no multicast members and data packets are still being sent to the group.

Compared with the total number of destinations within the greater internet, the number of destinations having group members of any particular *wide-area* group is likely to be small. More importantly, bandwidth limitations, and therefore data and control message overhead, should not be ignored in a wide area context. In the case of distance-vector multicast schemes, routers that are not on the multicast delivery tree still have to carry the periodic truncated-broadcast of packets, and process the subsequent pruning of branches for all active groups. One particular distance-vector multicast protocol, DVMRP, has been deployed in hundreds of regions connected by the MBONE [10]. However, its occasional broadcasting behavior severely limits its capability to scale to larger networks supporting much larger numbers of groups, many of which are sparse.

1.3 Extending multicast to the wide area: scaling issues

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

Existing dense-mode link-state and distance-vector multicast routing schemes have good scaling properties only when multicast groups densely populate the network of interest, or when the overhead of dense-mode operation is negligible relative to the network resources. When most of the subnets or links in the (inter)network have group members, then the bandwidth, storage and processing overhead of broadcasting membership reports (link-state), or data packets (distance-vector) is warranted, since the information or data packets are needed in most parts of the network anyway. The emphasis of our proposed work is

[1] Some schemes reduce the number of outgoing interfaces further by using unicast routing protocol information to keep track of child-parent information [3, 5].

Figure 1: Example of Multicast Trees

to develop multicast protocols that will also efficiently support the sparsely distributed groups that are likely to be most prevalent in wide-area, multi-administration, inter-networks where resources must be used more conservatively.

1.4 Overhead and tree types

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

Thus far we have motivated our design by contrasting it to the traditional dense-mode IP multicast routing protocols. More recently, the Core Based Tree (CBT) protocol [11] was proposed to address similar scaling problems in support of sparse-mode multicast. CBT uses a single delivery tree for each group, rooted at a "core" router and shared by all senders to the group. As desired for sparse groups, CBT does not exhibit the occasional broadcasting or flooding behavior of earlier protocols. However, CBT does so at the cost of imposing a single shared tree for each multicast group.

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

We need to know the kind of degradations a core-based tree can incur in average networks. David Wall [12] proved that the bound on maximum delay of an optimal core-based tree (which he called a *center-based* tree) is 2 times the shortest-path delay. To get a better understanding of how well optimal core-based trees perform in average cases, we simulated an optimal core-based tree algorithm over large number of different random graphs. We measured the maximum delay within each group, and experimented with graphs of different node degrees. We show the ratio of the CBT maximum delay versus shortest-path

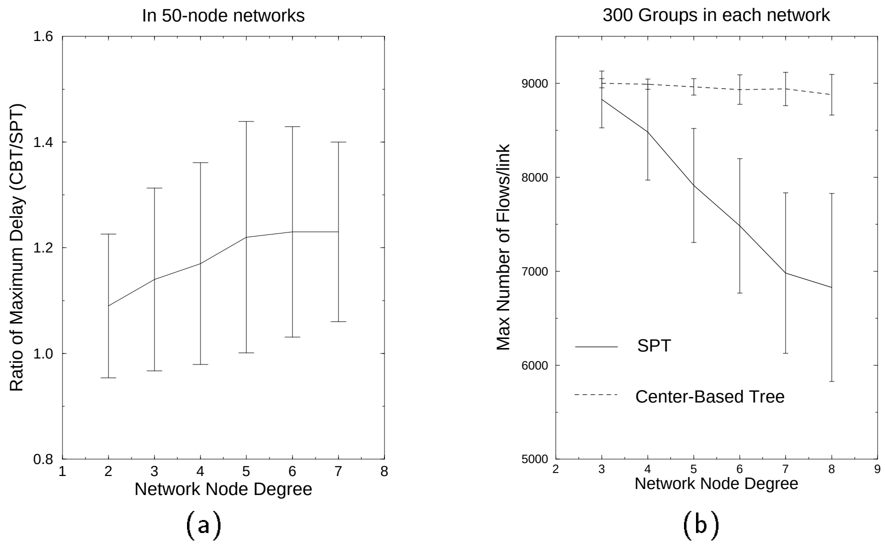

Figure 2: Comparison of shortest-path trees and center-based tree

tree maximum delay in Figure 2(a). For each node degree, we tried 500 different 50-node graphs with 10-member groups chosen randomly. It can be seen that the maximum delays of core-based trees with optimal core placement, are around 100% to 140% that of the shortest-path trees[2].

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

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

It is evident to us that both tree types have their advantages and disadvantages. One type of tree may perform very well under one class of conditions, while the other type may be better in other situations. For example, shared tress may perform very well for large numbers of low data rate sources (e.g., resource discovery applications), while SPT(s) may be better suited for high data rate sources (e.g., real time teleconferencing).[3] It would be ideal to flexibly support both types of trees within one multicast architecture, so that the selection of tree types becomes a configuration decision within a multicast protocol.

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

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

The capability to support different tree types is the fundamental difference between PIM and CBT.

[2] Note that although some error bars in the delay graph extend below 1, there are no real data points below 1 — the distribution is not symmetric, for more details see [13].

[3] A more complete analysis of these tradeoffs can be found in [13].

There are other significant protocol engineering differences as well, the most significant of which is PIM's use of soft state reliability mechanisms. CBT uses explicit hop-by-hop mechanisms to achieve reliable delivery of control messages. As described in the next section, PIM uses periodic refreshes as its primary means of reliability. This approach reduces the complexity of the protocol and covers a wide range of protocol and network failures in a single simple mechanism. On the other hand, it can introduce additional message protocol overhead.

1.5 Integrated dense-mode and sparse-mode protocol

While this new architecture was motivated primarily by the need for sparse-mode functionality, it also specifies a new dense-mode protocol instead of relying on existing dense-mode protocols such as DVMRP and MOSPF. PIM-Dense Mode (PIM-DM) is similar in behavior to DVMRP in that it relies on a form of Reverse Path Forwarding [9, 3]. However, PIM-DM has two important differences:

- PIM-DM makes use of unicast routing tables, independent of the protocol that created them. DVMRP carries around its own unicast routing information and makes use of a RIP-like [14] protocol to compute needed unicast routing information; as a result it has the scaling limitations of RIP and does not take advantage of information and computation already being carried out by the network's unicast routing protocol. MOSPF does take advantage of the unicast routing protocol, but it is specific to that one protocol, OSPF [7].
- PIM-DM control message processing and data packet forwarding is integrated with PIM-SM operations so that a single router can run different modes for different groups.

Note that while we have developed a new dense-mode protocol to accompany PIM-SM, we also recognize and address the need for interoperability with existing dense-mode protocols.

1.6 Document organization

In the remainder of this document we enumerate the specific design requirements for wide-area multicast routing (section 2), summarize the architectural components and functions (section 3), enumerate several protocol engineering choices made in the design of PIM protocols (section 4), consider the use of aggregation to address the scalability problem (section 5), and discuss open issues (section 6). Protocol details can be found in [1].

2 Requirements

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

- **Efficient Sparse Group Support**

  We define a sparse group as one in which

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

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

  (c) the inter-network spanned by the group is not sufficiently resource rich to ignore the overhead of current schemes.

  Sparse groups are not necessarily "small"; therefore we must support dynamic groups with large numbers of receivers.

- **High-Quality Data Distribution**

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

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

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

- **Routing Protocol Independent**

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

- **Accommodate Dense Mode Behavior**

  For those groups whose members and sources reside completeley within a contained campus network or region, we wish to operate in a mode that is more similar to traditional IP multicast, i.e., data-driven state creation with implicit joining and explicit pruning (default to send).

- **Interoperability**

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

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

- **Robustness**

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

  (a) using *soft state* refreshment mechanisms,

  (b) avoiding a single point of failure, and

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

- **Scalability**

  We provide mechanisms for scaling with group and network size. These mechanisms address the forms of overhead: control messages and state. Bandwidth consumed by data packets is already minimized through the use of explicit-join sparse mode. Control message overhead is limited to a fixed percentage of the link bandwidth by adjusting the frequency of periodic messages on a link by link basis.[4]

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

3 PIM Components and Functions: Overview

In this section we describe the architectural components of PIM. For clarity, we describe the general behavior of PIM-Dense Mode and PIM-Sparse Mode separately. However, the detailed protocol mechanisms developed to realize sparse and dense mode functionality are described in an integrated manner in [1].

3.1 PIM-Dense Mode (PIM-DM)

Dense-mode PIM uses Reverse Path Multicasting (RPM). RPM is a technique in which a multicast datagram is forwarded if the receiving interface is one used to forward unicast datagrams to the source of the datagram. The multicast datagram is then forwarded out all other interfaces. Dense-mode PIM builds source-based acyclic trees.

Dense-mode PIM is data driven; a node creates a multicast forwarding entry for a particular source-rooted distribution tree when a data packet from that source to the group first arrives. In creating this entry it is assumed that all downstream systems want to receive multicast datagrams. For densely populated groups, or in networks where the bandwidth is plentiful, this "default to send" behavior is optimal. If some areas of the network do not have group members, dense-mode PIM will prune branches of the source-based tree. When group members leave the group, branches will also be pruned.

Unlike DVMRP [5], packets are forwarded on all outgoing interfaces (except the incoming) until pruning and truncation occurs. DVMRP makes use of parent/child data to reduce the number of outgoing interfaces used before pruning. In both protocols, once truncation occurs pruning state is maintained and packets are only forwarded onto outgoing interfaces that in fact reach downstream members. We chose to accept additional overhead in favor of reduced dependency on the unicast routing protocol, and reduced overall protocol complexity.

Dense-mode PIM differs from sparse-mode PIM in two essential points:

(a) there are no periodic joins transmitted, only explicit triggered grafts/prunes, and

(b) there is no Rendezvous Point (RP).

3.1.1 Comparison to other dense-mode schemes

Reverse Path Broadcasting (RPB) is different from RPF because duplicate packets are avoided in the RPB that are sent in RPF. In general, the number of duplicates sent on a link can be as high as the number of routers directly connected to that link.

Reverse Path Multicasting (RPM) is different from RPF or RPB because pruning information is propagated upstream. Leaf routers must know that they are leaf routers so that in response to no IGMP reports for a group, those leaf routers know to initiate the prune process.

In DVMRP there are routing protocol dependencies for

(a) building a parent/child database so that duplicate packets can be eliminated,

[4] This method of controlling overhead was proposed by Van Jacobson.

(b) eliminating duplicate packets on multi-access LANs, and

(c) sending “split horizon with poison reverse” information to detect that a router is not a leaf router (if a router does not receive any poison reverse messages ¿from other routers on a multi-access LAN then that router acts as a leaf router for that LAN and knows to prune if there are not IGMP reports on that LAN for a group G).

Dense-mode PIM will accept some duplicate packets in order to avoid being routing protocol dependent and avoid building a parent/child database.

We introduce a simple prune mechanism for reducing duplicates on multi-access LANs. We introduce a simple graft mechanism to reduce join latency on previously pruned branches of a source-based multicast tree. We introduce an alternative leaf-router detection mechanism that does not rely on a specific unicast routing protocol mechanism such as split horizon with poison reverse. These mechanisms are described in detail in the protocol specification document.

3.2 PIM-Sparse Mode (PIM-SM)

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

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

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

The shortest-path-tree state maintained in routers is roughly the same as the forwarding information that is currently maintained by routers running existing IP multicast protocols such as MOSPF, i.e., source (S), multicast address (G), outgoing interface set (*oif*), incoming interface (*iif*)[5]. We refer to this forwarding information as the multicast forwarding entry for (S,G).

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

Figure 3 shows a simple scenario of a receiver and a sender joining a multicast group via an RP. When the receiver wants to join a PIM multicast group, its last-hop PIM router (A in fig 3) sends a PIM-Join message towards one of the RPs advertised for the group[6]. Processing of this message by intermediate routers sets up the multicast tree branch from the RP to the receiver. When sources start sending to the multicast group, the designated router (D in fig 3) sends a PIM-Register message, encapsulating the data packet, to the RP(s) for that group. The RP responds by sending a join towards the source. Processing

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

[6] If the last-hop router does not have RP information then the group is treated in dense mode

Figure 3: How senders rendezvous with receivers

of these messages by intermediate routers (there are no intermediate routers between the RP and the source in fig 3) sets up a packet delivery path from the source to the RP(s).

If source-specific distribution trees are desired, the last-hop PIM router for each member eventually joins the source-rooted distribution tree for each source by sending a PIM-Join message towards the source. After data packets are received on the new path, router B in fig 3 sends a PIM-prune message towards the RP[7].

One or more Rendezvous Points (RPs) are used *initially* to propagate data packets from sources to receivers. An RP may be any PIM-speaking router that is close to one of the members of the group, or it may be some other PIM-speaking router in the network. A sparse-mode group, i.e., one that the receiver's directly connected PIM router will join using PIM, is identified by the presence of RP address(es) associated with the group in question. The mapping information may be configured or may be learned through another protocol mechanism (e.g., a new IGMP message used by hosts to distribute information about RPs to their local routers [15]).

PIM avoids explicit enumeration of receivers, but does require enumeration of sources. If there are very large numbers of sources sending to a group but the sources' average data rates are low, then it may be more efficient to support the group with a shared tree instead which has less per-source overhead. If shortest-path trees are desired then when the number of sources grows very large, some form of aggregation can be employed; see section 5. We selected this tradeoff because in many existing and anticipated applications, the number of receivers is much larger than the number of sources. And when the number of sources is very large, the average data rate tends to be lower (e.g. resource discovery).

In summary, once the PIM-Join messages have propagated upstream from the RP, data packets from the source will follow the (S,G) distribution path state established. The packets will travel to the receivers via the distribution paths established by the PIM-Join messages sent upstream ¿from receivers towards the RP. Multicast packets will arrive at some receivers before reaching the RP if the receivers and the source are both "upstream" of the RP. When the receivers initiate shortest-path distribution, additional outgoing interfaces will be added to the (S,G) entry and the data packets will be delivered via the shortest paths to receivers. Data packets will continue to travel from the source to the RP(s) in order to reach new receivers. Similarly, receivers continue to receive some data packets via the RP tree in order to pick

[7] B knows, by checking the incoming interface in it routing table, that it is at a point where the shortest-path tree and the RP tree branches diverge. A flag, called SPT-bit, is included in (S,G) entries to indicate whether the transition from shared tree to shortest-path tree has finished. This minimizes the chance of losing data packets during the transition.

up new senders. However, when source-specific tree distribution is used, most data packets will arrive at receivers over a shortest-path distribution tree.

3.3 Sparse mode/dense mode interaction

There are two important points regarding the interaction of SM and DM:

1. If a multicast data packet arrives for which there is no multicast forwarding state, and no RP information, the data packet will be "flooded" as described in the Dense Mode protocol.

2. If a multicast group is "wide-area", i.e., it has RP(s) associated with it, then both tree management (joining, pruning, and registering) and data packet forwarding will be handled in a Sparse Mode manner.

To summarize, SM links are never treated in DM, but DM links are always treated in SM when the group itself is SM.

4 Protocol Engineering Design Features

In this section we describe engineering features embodied in the PIM protocols: robustness, sparse-mode/dense-mode interaction, PIM/non-PIM interaction and multicast service interfaces.

4.1 Robustness features

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

4.1.1 Lost PIM messages

The protocol is fairly robust to lost control messages. If a PIM-Register message gets lost then data packets will continue to be encapsulated in subsequent PIM-Register messages until the RP initializes an (S,G) entry, sends a PIM-Join message to the source, and until the associated PIM-Register-Stop messages propagate up to the source.

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

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

4.1.2 Multiple Rendezvous Points and RP failure scenarios

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

When the RP fails or becomes unreachable by receivers, members who have already joined will continue to receive packets from sources that had previously sent to the group and for which the receivers had already switched to the SPT (assuming the SPT is not affected by the same failure as makes the

RP unreachable). However, new members will send joins towards the unreachable RP and will *not* be successfully joined to the group unless their join packets reach existing SPTs of the sources before they reach the RP. New sources will attempt to register and send to the failed RP. As a result, their packets will not be delivered to any receivers and the SPT from the source to receivers will never be set up even if the paths that make up the SPT are available. This leads to the motivation for employing multiple RPs.

Unreachable RPs are detected by downstream routers using the RP-Reachability message. When a (*,G) entry is established by a router with local members, a timer is set. The timer is reset each time an RP-Reachability message is received. If this timer expires, the router looks up an alternate RP for the group, sends a join towards the new RP. A new (*,G) entry is established with the incoming interface set to the interface used to reach the new RP. The outgoing interface list includes only those interfaces on which IGMP-Reports for the group were received.

When multiple RPs are used, each source registers and sends data packets towards each of the RPs, but receivers only join towards a single RP. If one of the RPs fails, receivers that joined to that RP will stop receiving RP-Reachability messages and will start sending joins to one of the alternative RPs. Sources take different actions. When an RP is unreachable it will not receive the source's register messages and therefore will not respond with joins and so the outgoing interfaces in (S,G) pointing towards the unreachable RP will time out; without any explicit action on the part of the source. However, when the RP comes back up the first-hop routers need to inform the RP about sources it had previously sent Registers for. This allows the RP to join to those sources so data can travel natively rather than encapsulated.

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

4.1.3 Unicast routing changes

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

4.2 Dense mode and sparse mode interaction

The basic difference between PIM-DM and PIM-SM is that the former is data driven. Four important behavioral differences result:

- Dense mode sends and stores explicit prune state in response to unwanted data packets. Sparse mode requires explicit joining; the default action is to not send data packets where they have not been requested.
- Sparse mode stores join state in anticipation of data packets; Dense-mode routers only store state in response to arriving data packets (i.e. for active data sources).
- Sparse mode relies on the concept of an RP for data to be delivered to receivers who request to join the group. Dense-mode groups do not require an RP.

- Sparse mode relies on periodic refreshing of explicit join messages. Dense mode does not need to send prune messages periodically because of its data driven nature.

In simplified terms, the cost of dense mode is the default flooding behavior, whereas the cost of sparse mode is the need for RPs, RP-tree state for idle groups, and periodic refreshes.

If all members of a group are located within a region the group may be supported in a strictly dense mode. These groups require no RP to be configured or used, and shortest-path trees are built in a data driven manner.

Groups that do not make use of RPs will *not* be able to include any receivers that are beyond the scope of the multicast address. PIM-SM is designed to address the more general problem of groups that are not a priori limited to intra-domain membership and must therefore span sparse-mode interfaces and boundaries. Any such group that is not strictly local to a dense-mode configured domain must have at least one RP defined. All receivers join such inter-domain groups using periodic explicit join messages defined in the Sparse Mode protocol.

In other words, if a group is wide area and has RP's associated with it, then all members in PIM regions will Join the group using the PIM-SM protocol.

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

4.3 Interoperation with non-PIM networks

We wish to interoperate with networks that do not have hosts and routers modified to generate and interpret PIM-Join messages. We have to address two functions: pulling data down to the non-PIM DM cloud, and propagating data packets through the cloud even when they arrive on the RP tree.

- In PIM-SM, receivers are not passive, they must take explicit join action to receive data packets. This creates problems when a non-PIM, dense-mode region, wishes to interoperate with PIM-SM. In particular, when a receiver decides to join a group inside of a non-PIM cloud in which there are no other members then the PIM/non-PIM border routers (PIM-BRs) of that cloud must be notified in order to trigger sending of a PIM-Join message towards the RP. Similarly, if join comes upstream from a SM region, and the RP or source is on the other side of the non-PIM region, then the PIM-BRs must be notified in order to have the join message propagate upstream of the non-PIM region.

- Data packets will not be flooded through the non-PIM region if they arrive via the wrong incoming border router. Therefore we need to introduce some additional mechanism to cause RP-tree packets to be forwarded through the non-PIM region. In order for the non-PIM cloud to propagate a data packet ¿from the RP tree, through to any internal members and to other PIM-BRs (which might have downstream members), the packet must be injected via the PIM-BR(s) that are the shortest-path tree entry points from the packet source, S, to the routers inside the non-PIM region.

The details of these mechanisms are described in [1].

When a source inside of a non-PIM region is sending to a non-local group, all the PIM-BRs that have external shortest paths to the RP must send register messages. The RP(s) will resolve the problem by sending a join to only one of them and an indication to stop to all of them. The RP need not choose an optimal BR; any one will do.

4.4 Multicast service interface

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

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

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

We propose to develop an IGMP RP-Report message that is also sent in response to an IGMP-Query. The RP-Report lists all the RPs for a particular group.

RP-Report messages will be sent to the group. This will cause receiver's to participate in suppression. This seems acceptable given the other tradeoffs.[8]

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

5 Scaling and Aggregation

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

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

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

If PIM is being used for inter-domain routing, and routers are able to map from IP address to domain identifier, then one possibility is to use the domain level aggregate for a source in PIM messages (Autonomous System (AS) numbers or Routing Domain Identifiers (RDIs)). Then the PIM message will travel to the PIM-BRs of the domain and the PIM-BRs can use the internal multicast protocol's mechanism for propagating the join within the domain (e.g. send appropriate link-state advertisement

[8] For more information on RP-Report messages, see upcoming Internet-Draft on new IGMP messages by S. Deering.

in MOSPF or register a "local member" and do not prune in the case of RPF). However this approach requires that it is both possible and efficient to map from IP to domain address when processing data packets, as well as control packets.

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

5.1 Containing control traffic overhead

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

The time-out mechanism is determined by the sender of the information. Therefore, a router tells its neighbors how long to keep it reachable by advertising the holdtime in PIM-Query messages. Likewise, join/prune messages and RP-Reachable messages indicate how long state should be kept. This allows the sender to change its frequency without the receivers requiring any special configuration information.

Note that across regions that drop state (see below), the timer is no longer across a link, but is across the cloud as a whole. Routers within the cloud do not control the frequency hop-by-hop, but just pass thru control messages generated at the edges of the cloud. So the border routers have to set their timers so as to constrain protocol overhead across the cloud.

5.2 Containing state overhead

If the state in any particular router grows too big, that router can drop the state and reconstruct state for active data sources only. This technique has the important property that they do *not* require any coordinated action across routers; routers act unilaterally according to their aggregation needs.

When a router is overloaded with state, we propose that it drop state in an LRU fashion and rebuild needed state in a data-driven fashion as needed. Conceptually, this approach emulates dense-mode data-driven behavior, but builds SM state in order to reduce the amount of prune state stored in the SM region.

In other words, a router in this state, builds (S,G) entries in a data-driven manner. However to reach all downstream members it populates the outgoing interface list with all interfaces other than the incoming. The state is SM state because each outgoing interface is timed out after some period if an explicit, SM join is not received.

Although state is built in a data driven manner, SM joins that arrive ¿from downstream are still propagated upstream through the aggregating region. The Joins from downstream that match on local entries, reset the timers; however, joins that do *not* match are just passed through (state is NOT created in response to a join). State is not created because the whole point of the scheme is to not create state in advance of data. The Joins must still be propagated upstream so that upstream regions can keep regular SM state and are not forced to time out their explicit join state and cause black-holes as a result of an intermediate router dropping state. In summary, sparse-mode join information gets propagated up to the data source, the data packets thereby arrive at the aggregating region;s BR(s) and are automatically propagated through the aggregating region using the data-driven mechanisms.

However, as with PIM/non-PIM interaction, special actions must be taken to propagate RP-tree packets through an aggregating region. To do so, the BRs at the border between the aggregating and non-aggregating region, must encapsulate and decapsulate RP-tree packets as they enter and exit the region, respectively.

6 Open Issues

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

- **RPs** There are several open issues with respect to supporting RPs (this is not surprising since the concept does not exist in current IP multicast routing).
  - **Distinguishing between DM and SM groups**

    It would be useful to know explicitly if a particular group had RPs associated with it, and if therefore the SM protocol should be used to participate. To this end we are considering defining a portion of the multicast address space for use by wide-area, inter-domain groups that use RPs.
  - **Selecting RPs**

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

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

    Hosts must notify their DRs of the RPs associated with a particular group. We are developing an IGMP-RP-Report message to be used for just this purpose.
- **Interaction with policy-based and QOS routing**

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

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

  If the path to reach the RP or some source is indicated as being the appropriate QOS and indicated as being symmetric then PIM routers can determine that if they forward joins upstream that the data packets will allowed to travel downstream. This implies that BGP/IDRP [16, 17] should carry two QOS flags: symmetry flag and multicast willing flag.

  If the generic route computed by hop-by-hop routing does not have the symmetry and multicast bits set, but there is an SDRP [18] route that does, then the PIM message should be sent with an embedded SDRP route. This option needs to be added to PIM join messages. Its absence will indicate forwarding according to the router's unicast routing tables. Its presence will indicate forwarding according to the SDRP route. This implies that SDRP should also carry symmetry and multicast QOS bits *and* that PIM should carry an optional SDRP route inside of it to cause the PIM message and the multicast forwarding state to occur on an alternative distribution tree branch.

- **Interaction with receiver-initiated reservation setup such as RSVP** [19]

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

  In summary we expect that this receiver initiated routing is well suited to receiver initiated reservations since if a reservation is blocked the previous router or the receiver can select an alternative reverse path to the particular source(s). This is also a subject for future work that will affect the use of the protocol, and not the protocol itself.[9]

7 Conclusions

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

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

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

(c) does not depend on the underlying unicast protocols; and

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

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

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

8 Acknowledgments

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

References

[1] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. Protocol independent multicast (pim): Specification. *Working Draft*, November 1994.

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

[9] The interaction of PIM, SDRP, and RSVP is currently being investigated by D. Zappala, S. Shenker, and D. Estrin.

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

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

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

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

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

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

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

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

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

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

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

[14] G. Malkin. Rip version 2 carrying additional information, jun 1993. RFC1388.

[15] S. Deering. Igmp. *???*, November 1994.

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

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

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

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