Deployment of PIM-SO at Sprint

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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 and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC 2119].

1. Background and Overview

This document considers the requirements for implementing PIM Source-Only (PIM-SO) which will allow the creation of source-based multicast trees across multiple domains in the Internet. PIM-SO realizes the EXPRESS [HOLB99] multicast service model in which there is a single source for every multicast group, and group membership and addressing is based on a multicast group address (G) as well as the source’s
unicast address (S). Our short-term goal is to implement PIM-SO with minimal changes to the current multicast routing infrastructure by August 2000, and to provide proof of concept by deploying it on Sprintlab’s experimental network. By the end of the year, we envisage that PIM-SO will be ready to be incorporated in Sprint’s multicast backbone as a commercial point-to-multipoint service.

The current IP multicast service model is an open model where a set of hosts can be aggregated into a group with a single class-D IP address in the range of 224.0.0.0 to 239.255.255.255. Any end-host can send data to this multicast group (even without joining the group), and any end-host can receive data sent to this group by becoming a member of the group. Currently there is no mechanism to restrict a host from joining a multicast group. End-host register multicast group membership with edge-routers (i.e. routers with directly attached hosts) using the IGMP \[\text{FENN97, CAIN99}\] protocol. Multicast-capable routers then exchange messages with each other according to some routing protocol to construct a distribution tree connecting all the end-hosts. A number of different protocols exist for multicast routing, which differ mainly in the type of delivery tree constructed [DEER90, DEER96, PIMSM, PIMDM]. Of these, the Protocol Independent Multicast Sparse-Mode (PIM-SM) protocol is the most widely deployed in today’s public networks. PIM-SM, by default, constructs a single spanning multicast tree rooted at a core (rendezvous point or RP) for all group members. However, it also allows a multicast group member to switch to a source-based shortest path tree if it so desires. In the current protocol architecture, PIM-SM is used for intra-domain routing, while the Multicast Source Discovery Protocol (MSDP) [FARI00] is used for building trees across multiple administrative domains.

The deployment and use of IP multicast as an inter-domain commercial service is hindered by a number of shortcomings in the current service model and architecture. For example, global address allocation remains an open issue, and there is no support for group access control and network management. For a detailed discussion of some of these issues, refer to [DIOT00]. The recently proposed Explicit Request for Single Source (EXPRESS) multicast service model promises to address some of these deficiencies [HOLB99]. In the EXPRESS model, a multicast group is determined by specifying a source address S as well a group address G. Therefore two groups (S1,G) and (S2,G) are completely unrelated, even though they have the same multicast group address. The IANA has allocated the address range 232/8 for experimentation with a single-source multicast service. In the rest of this document, we refer to an address in this range as a PIM-SO ADDRESS.

The Internet multicast community is converging towards single-source
multicasting as an immediately deployable inter-domain solution. Although this model restricts each multicast group to a single source, it is sufficient for enabling large-scale point-to-multipoint applications such as Internet TV that are expected to dominate the Internet multicast application space in the near future. Hence a single-source multicast service will provide tremendous impetus to Internet multicast deployment and will pave the way for a more general multipoint-to-multipoint service in the future.

Our starting point for developing a single-source multicast service is the current multicast infrastructure deployed by large network service providers such as SPRINT, which consists of IGMP version 2 for end-host memberships, PIM-SM for intra-domain routing, and MSDP for inter-domain routing. We intend to implement PIM-SO as a combination of PIM-SM and IGMP, and eliminate the need for MSDP for single source multicast sessions.

A key goal of our PIM-SO deployment effort is COEXISTENCE WITH THE CURRENT WIDE-AREA MULTICAST INFRASTRUCTURE. Hence the changes that we propose will not cripple the current architecture and the applications that it supports. We believe that this will allow an incremental deployment strategy for PIM-SO which is appropriate for today’s wide-area networks. Moreover we intend to carefully examine interoperability issues arising out of the coexistence of the PIM-SO infrastructure and the classic infrastructure.

We feel that the changes needed for integrating PIM-SO into the PIM-SM/MSDP infrastructure are greatly simplified by a strict partition of the multicast address space between PIM-SO sessions and classic PIM-SM sessions. This would mandate that PIM-SO addresses be used ONLY for single-source multicast with source-based trees and non PIM-SO addresses are used for classic PIM-SM style multicasting. In the rest of this document we assume this strict partition for an initial deployment effort. However, in a later section, we discuss the implications of relaxing this requirement.

The changes needed for deploying PIM-SO can be broadly categorized as follows:

1. Changes to PIM-SM at core routers.

2. Changes at edge-routers, i.e., routers with directly connected end-hosts. These routers use the IGMP protocol to interact with end-hosts regarding group membership, and use the PIM-SM protocol to interact with core network routers for constructing distribution trees. Hence we need to consider changes to both IGMP and PIM-SM at these edge-routers.
(3) Changes at end-hosts. This involves changes to IGMP, which is used by end-hosts for joining/leaving multicast groups. Briefly, IGMP must be modified in order to enable end-hosts to register their interest in a multicast group identified as a (source, group) pair. This is supported by the recently proposed version 3 of IGMP (IGMPv3) [CAIN99]. We anticipate that only a subset of IGMPv3 capabilities will be required to provide PIM-SO functionality. Changes are also required for applications on end-hosts so that they can participate in single-source PIM-SO sessions.

In the following sections, we discuss the rationale behind deploying PIM-SO, changes needed to the current infrastructure, and the tradeoffs and open issues in deploying PIM-SO. We also outline our goals and milestones for a project that involves implementing and deploying PIM-SO on an experimental network to provide proof of concept.

2. Architectural Overview

In order to deploy PIM-SO, we need to provide support for constructing source-based trees and for routing multicast data based on both a group address and a source address (we henceforth refer to this as (S,G) routing). This in itself is not a radical departure from the current standards; however, on an architectural level, PIM-SO represents a significant rethinking about the separation of functionality between the ROUTING LAYER and the APPLICATION/CONTROL LAYER for wide-area multicasting. Specifically, it aims to restrict the role of a multicast routing protocol to building a multicast tree, and forwarding data packets along that tree. The responsibility of discovering the identity of a multicast source (or equivalently, a multicast service) is left to the application/control layer. This can be achieved through a session advertisement tool such as SDR [SDR], which would have to announce a source address and a group address for a PIM-SO multicast group.

The current IP multicast architecture, in its attempt to realize an open many-to-many service model, fails to provide this separation of functionality between service discovery and multicast routing [DIOT00]. This shortcoming is reflected most prominently in the design of the MSDP protocol. MSDP has been designed to connect multiple PIM-SM domains by "discovering" and announcing multicast data sources across domains. However, in practice, most MSDP source announcement (SA) messages carry data, even though this is not mandated by the specifications [FARI00].
Figure 1 is a simple high-level view of a multicast architecture based on PIM-SO. An advertising tool such as SDR can be used to announce multicast services using an abstract naming scheme (a discussion of an appropriate naming scheme is beyond the scope of this document). If an end-host is interested in a specific multicast service (e.g., channel 2 of content provider XYZ), it contacts a "well-known" multicast name server to resolve this name to an (S,G) address. Then it uses the (S,G) address to join the source-based multicast tree for that service.

Of course, the session advertisement tool can itself advertise the source and group address for a multicast service, in the same way as SDR. However, there are some inherent advantages to having a separate name server. For example, a content provider may choose to offer the same service from multiple locations, using the same group address G but different sources. In that case a name server, on receiving a query for a service, can map the service onto the source address "closest" to the requesting host. A multicast name server can also provide the point of control for a number of control functions that are likely to be an integral part of a commercial multicast service. For example, it may also function as an authentication server, authenticating a user and provide a decryption key for decrypting data that is sent in an encrypted form from a multicast source. The server can also support a host of other functions such as billing, access control, customer service, community management, etc.
We now discuss the changes to PIM-SM, IGMP and applications for deploying and using PIM-SO. However, we do not discuss in detail the specifics of IGMPv3 [CAIN99] since this being addressed in a systematic way in [HOLB00].

deployed version of IGMP (IGMPv2) allows end-hosts to register their interest in a multicast group by specifying a class-D IP address. However in order to implement the single-source multicast model, an end-host must specify a source’s unicast address as well as a group address. This capability is provided by the recently proposed IGMP version 3 (IGMPv3). IGMPv3 supports "source filtering", i.e., the ability of an end-system to express interest in receiving data packets sent only by SPECIFIC sources, or from ALL BUT some specific sources. Thus, IGMPv3 provides a superset of the capabilities required to realize the EXPRESS model. Hence an upgrade from IGMPv2 to IGMPv3 is an essential change for implementing single-source multicast. We believe that this is the MOST EXTENSIVE CHANGE FOR PIM-SO DEPLOYMENT as it involves changes to the Application Programming Interface (API) on ALL END-HOSTS that want to participate in PIM-SO sessions.

Let us now discuss one important change to the API on end-hosts that will be required to support IGMPv3. IGMPv3 requires the API to provide the following operation (or its logical equivalent) [CAIN99]:

IPMulticastListen (Socket, IF, G, filter-mode, source-list)

"Socket" is an implementation-specific parameter used to distinguish among different requesting entities. IF is a local identifier of the network interface on which the specified multicast address is to be enabled or disabled and G is the multicast group address to which the request pertains. filter-mode may be INCLUDE or EXCLUDE. In INCLUDE mode, reception of packets sent to the specified multicast address is requested only from the IP addresses listed in the source-list parameter. In EXCLUDE mode, reception of packets sent to the given multicast address is requested from all IP source addresses except for those listed in the source-list parameter.

As explained in the IGMPv3 specifications [CAIN99], the above IPMulticastListen() operation subsumes the group-specific join and leave operations of IGMPv2. Performing (S,G)-specific joins and leaves is also trivial. A join operation is equivalent to:

IPMulticastListen (Socket,IF,G,INCLUDE,{S})
and a leave operation is equivalent to

\[ \text{IPMulticastListen}(\text{Socket}, \text{IF}, \text{G}, \text{EXCLUDE}, \{S\}) \]\n
For IGMPv3 to support PIM-SO, it may be useful to include a check to see if the group address in an IGMP IPMulticastListen() message is a PIM-SO address. If it is, then IGMPv3 should ensure that one and exactly one source address is specified on the source-list. Moreover, if a system (end-host or edge-router) supports both IGMPv2 and IGMPv3, it must ignore any IGMPv2 message for a PIM-SO address.

There are a number of backward compatibility issues between IGMP versions 2 and 3 which have to be addressed. A detailed discussion of these issues is provided in [HOLB00]. One of our goals for the PIM-SO deployment effort is to implement IGMPv3 on a Linux platform; we expect to address and understand IGMP backward compatibility issues as part of that implementation.

4. PIM-SM Modifications

The Protocol Independent Multicast (PIM) protocol has two distinct flavors. The first, known as PIM Sparse Mode (PIM SM), supports sparsely populated multicast groups across wide areas by constructing receiver-driven multicast trees. The second, known as PIM Dense Mode (PIM-DM), supports the construction of data-driven trees for dense groups, using a DVMRP-like [DEER90] flood and prune mechanism. PIM-SM itself supports two types of trees, a shared tree rooted at a core (RP), and a source-based shortest path tree. THUS PIM-SM ALREADY SUPPORTS SOURCE-BASED TREES; however, PIM-SM is not designed to allow a router to choose between a shared tree and a source-based tree. In fact, a receiver always joins a PIM shared tree to start with, and may later be switched to a per-source tree by its adjacent edge router. Let us now look at this issue in some more detail.

We start with a brief overview of tree construction in PIM-SM. A PIM-SM router receives explicit Join/Prune messages from neighboring routers that have downstream members for a particular multicast group. The router forwards data packet received addressed to a multicast group G, only onto those outgoing interfaces on which explicit joins have been received. A Designated Router (DR) (an edge router with directly attached end-host, that talks IGMP to end-hosts, and PIM-SM to other PIM routers) sends periodic Join/Prune messages towards a group-specific Rendezvous Point (RP) for each group for which it has active members. Thus, by default, PIM-SM sets up a shared tree centered at the RP. When a data source first starts sending data to a multicast group, its DR forwards the data to the RP for that group, from where messages are multicast over the shared tree.
A router with directly connected members may switch to a source’s shortest-path tree if it so chooses. For this purpose, PIM-SM makes a distinction between the \((*,G)\) state which is specific to a group (the source is a wildcard), and the \((S,G)\) state in which both a source and a group are specified. A router switches to the per-source tree by initiating a Join message that indicates \((S,G)\) information (the details are omitted). Join messages then propagate back towards the source establishing \((S,G)\) state at intermediate routers that are subsequently used to forward data packets. Once the receiver starts receiving data from the source-based tree, the DR sends an \((S,G)\) Prune message towards the RP, indicating the packets from source \(S\) on multicast group \(G\) need not be forwarded towards it along the shared tree.

A key to implementing PIM-SO is to eliminate the need for starting with a shared tree and then switching to a source-specific tree. This involves the following steps:

-- Allow an end-host to specify the source whose group it wants to join. As discussed earlier, this ability is provided by IGMPv3.

-- When a DR receives an IGMPv3 Join message for a group with a PIM-SO group address \((232/8)\), it ALWAYs initiate a \((S,G)\) join and NEVER a \((*,G)\) join. Moreover, unlike PIM-SM, it need not send a \((S,G)\) prune towards the RP.

In addition, there are a number of PIM-SM protocol details that need attention. Some of these are listed below; we expect to make additions and changes to this list as we proceed with the PIM-SO implementation:

-- DRs (with directly attached members) must recognize the address range \(232/8\) as PIM-SO addresses, designated for source-only multicast. ALL OF THE FOLLOWING POINTS PERTAIN TO ONLY PIM-SO ADDRESSES:

-- Join/Prune messages:

  -- Wildcard bit \(W\) is always set to 0 indicating that this message applies to \((S,G)\) entry.

  -- RPT-bit \(R\) is always set to 0 indicating that information is to be sent towards the source, and not the RP.

  -- For end-hosts joining PIM-SO sessions, edge-routers must not initiate \((*,G)\) joins towards the RP.
--Hosts sending to a group: The source’s DR never encapsulates data in PIM register messages to unicast to the RP. Instead, it forwards data according to the (S,G) entry in its routing table.

--Switching from shared-tree to SPT should not be required

--The SPT bit is always set to indicate that the source specific tree has been set up.

--At core routers, ignore all (*,G) Join/Prune messages for the for PIM-SO group addresses.

--Ignore all (*,G) packets at core and edge routers for PIM-SO addresses.

--A DR does not need to send a Prune message towards the RP (to detach itself from the shared tree) when it starts receiving data over the source-based tree.

--There is no need for Register and Register-Stop messages.

--For data packet forwarding in the PIM-SO address range, a router needs to perform a lookup on (S,G) entries only.

We also need to examine whether changes are needed to PIM routers in the core of the network. It is desirable to restrict changes to end-hosts and edge-routers, since that simplifies the deployment of PIM-SO in a network that already supports PIM-SM (such as Sprint’s IP backbone network). In order to prevent RPs at the core of a network from receiving (and possibly forwarding) data from sources, it is necessary to block all Source Advertisement (SA) requests at anycast RPs. Moreover we need to pay attention to a configuration issue for certain multicast-capable core routers. Currently, every such router is configured in Sparse Mode (SM)-only or in Sparse/Dense mode (SM/DM). The difference between the two configurations lies in the action taken by the router when it does not have knowledge of the RP associated with a group address found in an incoming packet. If the router is configured as SM-only, the packet is forwarded only if there is a (*,G) entry for that group, otherwise the packet is dropped. On the other hand, if the router is configured as SM/DM, then the packet is FLOODED ON ALL OUTGOING INTERFACES.

Given that PIM-SO eliminates the need for having RP addresses associated with PIM-SO addresses, we have to carefully consider the implications of doing away entirely with a mapping between PIM-SO addresses and RPs. The lack of such mapping will lead to the possibility of the flooding phenomenon described above for routers.
configured in the SM-only mode. Such flooding does not affect the correctness of the implementation, but it definitely hurts robustness by generating large numbers of unwanted packets.

The changes discussed so far can be implemented either as router configuration options or can be enforced in the PIM-SM code in the kernel. From a deployment standpoint, the latter option is preferable, since it will be easier to reconfigure the routers if the PIM-SO address range changes (or expands) in the future.

5. Dial-up Hub Modifications

Figure 2 is a simple high level view of Sprint’s existing dial-up hub architecture.

```
+-------+
|  R1   | PIM-SM
+-------+
    | Ethernet
+-------+
|  RAS  | IGMP proxy
+-------+
    | PPP
+--------+
| Dial-up |
| End Host|
+--------+
```

Figure 2

Dial-up users connect to the remote access server, RAS, through a standard PPP connection. The RAS connects to Router R1 via ethernet, and sends all default IP traffic to R1. When the end host joins a multicast group, it sends an IGMP Join message to RAS. RAS acts as a proxy and forwards this to R1. By simply proxying IGMP, RAS does not need to run any multicast routing protocol. In this way, it can support any underlying multicast routing protocol. So to enable a PIM-SO deployment, RAS simply must proxy IGMPv3 in the same manner it
proxies IGMPv1 and v2 today. Since R1 supports PIM-SM today, it too needs only to add support for IGMPv3.

6. Requirements for Multicast Applications

Thus far, we have discussed modifications to IGMP and PIM-SM in order to realize PIM-SO. Let us now consider how we can enable multicast applications to use the underlying PIM-SO routing infrastructure. Two important conditions have to be satisfied for this:

-- For single-source multicast sessions using PIM-SO addresses, session advertisement tools must advertise a source address as well as a PIM-SO address.

-- In order to join a PIM-SO based multicast tree, an application must specify a source address in addition to a PIM-SO group address. In other words, the application must be "IGMPv3-aware".

The issue of application requirements has to be considered from two perspectives. First, PIM-SO is expected to enable a large class of new point-to-multipoint services from large content providers. We can expect these content providers to write their applications to be IGMPv3-aware. At the same time, we need to consider modifications for existing applications. The single-source multicast model is a natural fit for popular commercial applications such as IPTV, Real's media player, and Windows Media Player since data is streamed from a single source. The client side of these applications must be capable of specifying both a source address and a group address in order to join PIM-SO sessions.

We now briefly outline our initial plans for implementing IGMPv3 on an IGMPv2-capable Linux platform, and for allowing users to participate in single-source PIM-SO sessions with minimal changes to existing applications. As shown in Figure 3, we plan to implement IGMPv3 as a separate kernel-loadable module. All incoming IGMPv3 messages to the IGMPv3-capable end-host will be handled by the IGMPv3 module while all IGMPv2 messages will be handled by IGMPv2-capable kernel as before. However, IGMPv3 messages for non PIM-SO addresses (e.g. a group-specific query) and IGMPv2 messages for PIM-SO addresses will be ignored (Figure 3). Similarly, on the outbound path, the IGMPv3 module will initiate joins and leaves for PIM-SO addresses, while all other addresses will be handled by IGMPv2 in the kernel.
We plan to use popular applications such as sdr, vic and vat for initial demonstrations of our PIM-SO implementation. Since will be used to "announce" PIM-SO sessions, it has to be modified to advertise a source address for every PIM-SO session. Sdr uses an RP-based infrastructure for discovering the multicast sessions that it announces; hence it is debatable whether it is suitable for a PIM-SO infrastructure which aims to eliminate the need for RPs altogether. However, we are planning to use sdr only for demonstrations in the short term, while we wait for more current applications such as IPTV to be made IGMPv3 awareness.

While we must modify sdr to support PIM-SO, we want to minimize the changes required for applications such as vic and vat. A possible way of achieving this is as follows. Suppose a single-source session (S,G) is advertised by sdr and a user wants to join the session with vat as the application tool. When vat is started up, it requests a "join" with only the group address G. However when G is found to be a PIM-SO address it is intercepted by the IGMPv3 module for processing. At the same time, sdr informs the IGMPv3 module of the source address S corresponding to that session. The IGMPv3 module then uses both the group and source addresses for initiating an IGMPv3 "join" for that session.

```
+-------+        G         +------+
|  vat  |<-----------------| sdr  |
+-------+                  +------+
      |
      |
      |
      |
      +---------------+      |
      +-->| IGMPv3 module |<-----+
      +---------------+
      |
      |
      |
      |
      +-----------+        |
      | IGMPv3 (S,G)|        |
      +-----------+        |
      | Join       |
      +-----------+        |
      | USER       |
      +-----------+        |
      |-----------------+   |
      | KERNEL        |
      +-----------+   |
      | IGMPv2  YES  +-------->
      +-----------+   |
      | NO          |
      +-----------+   |
      | v3?        |
```
7. Tradeoffs and Open Issues

In this section, we summarize some of the advantages and disadvantages of a PIM-SO based IP multicasting architecture, and discuss how some of the current shortcomings may be addressed in the future. Some of the advantages of PIM-SO are as follows:

-- Clear separation between routing and application/control level functionality, leading to a simpler, cleaner multicast architecture.

-- The architecture is much simpler to manage than the current IGMPv2/PIM-SM/MSDP architecture, making it attractive to network operators such as SPRINT who will deploy and offer multicast as a commercial service.

-- The service model is capable of supporting the requirements of 99% of Sprint’s customers today.

-- A number of functionalities such as access control, billing and authentication are expected to be a key part of a commercial multicast service offering. We discussed in Section 2 how a PIM-SO based routing architecture can support these key functionalities at the application/control level.

The most serious criticism of the PIM-SO architecture is that it does not support shared trees which may be useful for supporting many-to-many applications. In the short-term this is not a serious concern since the multicast application space is likely to be dominated by one-to-many applications. Some other classes of multicast applications that are likely to emerge in the future are few-to-few (e.g. private chat rooms, whiteboards), few-to-many (e.g., video conferencing, distance learning) and many-to-many (e.g., large chat rooms, multi-user games). The first two classes can be easily handled using a few one-to-many source-based trees.

The issue of many-to-many multicasting service on top of a PIM-SO architecture is an open issue at this point. However, we feel that even many-to-many applications should be handled with multiple one-to-many instead of shared trees. The rationale behind this lies in receiver interest filtering [LEVI00] - as the number of members in a multicast group increases, there is a decrease in the overlap of interest among members in a group. Hence a shared tree actually
wastes bandwidth by pushing data towards receivers that have no use for the data. Multiple source-based trees achieve bandwidth efficiency at the expense of a minimal increase in router forwarding state. But we believe that router memory will be less expensive than bandwidth; hence the tradeoff will work in favor of source-based trees.

8. Address Space Overlap

Moving to PIM-SO will create two kinds of problems in term of addressing. In the transition period, PIM-SO and PIM-SM need to co-exist. There will consequently be two types of multicast groups on the Internet: shared groups and single-sender groups. From an addressing standpoint, it is important that PIM-SO group use the 232/8 address range and that PIM-SM groups DO NOT use 232/8 addresses. Depending on whether a host is PIM-SO and/or PIM-SM enabled, it will need to have the following capabilities:

- a PIM-SM sender MUST use a class D address outside the 232/8 range.
- a PIM-SM receiver CAN issue a (*, G) join to PIM-SM group.
- a PIM-SO sender MUST use a class D address within the 232/8 range.
- a PIM-SO receiver CAN issue a (S, G) join to any multicast group, irrespective of whether it is a PIM-SO address or not.

These requirements are very important for addressing backward compatibility issues between PIM-SO and PIM-SM/MSDP. Once PIM-SO is deployed, shared groups will be supported by PIM-SM/MSDP and single-sender groups will be supported by PIM-SO. In the future, relying on PIM-SO only might require some more constraints on the addressing scheme to support shared groups. This problem is not addressed in this draft.

9. Experimental Testbed

Our near-term goal is to provide proof of concept that PIM-SO is capable of providing a robust and manageable multicast service. We propose to demonstrate this by deploying PIM-SO on the experimental network maintained by Sprintlabs. Figure 4 shows the current architecture (this architecture may change somewhat in the next few months) of this testbed. We now provide a brief description of our initial deployment plans over this network. Many of the details about the testbed are omitted for the sake of clarity.

As part of this testbed, one hundred residences are provided with a connectivity of 10 Mbps to an experimental network. The experimental network is firewall from the public Internet but connected at OC-3
speeds to Sprintlab’s internal testbed. In the initial experiment with PIM-SO, we intend to have a source located at Sprintlabs multicasting data to the residences participating in the testbed. A Linux-based PC in each residence will support IGMPv3 in the kernel, and will exchange IGMPv3 messages with router R3, which will support IGMPv3. Router R3 is also expected to support PIM-SM, modified to support single-source multicast for PIM-SO addresses. Routers R1 and R2 have PIM-SM capability and, together with R3, will build a tree reaching the hundred homes from a source in Sprintlabs. For the time being, a tunnel will be created through the firewall to allow multicast packets through.

![Diagram of PIM-SO Testbed]

We are exploring the possibility of involving a commercial content provider to participate in our experiments on the PIM-SO testbed. We also intend to examine the feasibility and security implications of allowing multicast sources from the public Internet to send data to
the testbed receivers. Finally, we are actively pursuing the upgrade of 3Com’s Total Control Hub (TCH) to support IGMPv3. This will allow receivers to connect to our testbed over a dial-up connection and receive multicast data sent over a PIM-SO tree.

10. Project Goals

May-July 2000 :
--Implement IGMPv3 for the Linux kernel on end-hosts.
--Modify end-systems to provide support for "IGMPv3-aware" applications.
--Work with router vendors to provide support for IGMPv3 on edge-routers.
--Test modified end-hosts and modified edge routers.

July-August 2000 :
--Deploy and test PIM-SO on the Sprintlabs testbed.
--Test dial-up multicast service using a IGMPv3-capable Total Control Hub (TCH) from 3Com.

September-October 2000 :
--Perform exhaustive testing on PIM-SO testbed.

November-December 2000 :
--Complete the implementation and testing of all components required for deploying PIM-SO on SprintLink’s backbone.

11. Acknowledgments

We would like to thank a number of people at Sprint Labs -- Gene Bowen, Ed Kress, Bryan Lyles, Sue Moon, Timothy Roscoe and JayCee Straley -- for participating in lengthy discussions on PIM-SO, and providing feedback on this document. Thanks are also due to Mujahid Khan and Ted Seely at SprintLink, Tom Pusateri at Juniper Networks, Isidor Kouvelas at Cisco Systems, Bill Fenner at AT&T Research, Kevin
Almeroth at the University of California Santa Barbara, Brian Levine at the University of Massachusetts Amherst, Brad Cain at Nortel Networks and Hugh Holbrook at Cisco Systems for their valuable insights and continuing support.

11. References:


2236, November 1997.


[HOLB00] H. Holbrook and B. Cain.
Use of the 232/8 Address Space. Work in Progress.


[DEER96] S. Deering et al.


[DEER99] S. Deering et al.

[FARI00] Farinacci et al.

[DIOT00] C. Diot, B. Levine, B. Lyles, H. Kassem and D. Balensiefen.
[SDR] Session directory (sdr).

[LEVI00] B. Levine et al.

12. Authors’ Address:

Supratik Bhattacharyya
Christophe Diot
Sprint Advanced Technology Labs
One Adrian Court
Burlingame CA 94010
{supratik,cdiot}@sprintlabs.com
http://www.sprintlabs.com

Leonard Giuliano
Robert Rockell
Sprint Internet Service Center
Reston, Virginia
{giuliano,rrockell}@sprintlink.net