The PIM Architecture for Wide-Area Multicast Routing

Stephen Deering, *Member, IEEE*, Deborah L. Estrin, *Senior Member, IEEE*, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei

***Abstract*— The purpose of multicast routing is to reduce the communication costs for applications that send the same data to multiple recipients. 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 *s*parsely across a wide area, these schemes are not efficient; data packets or membership report information are occasionally sent over many links that do *n*ot lead to receivers or senders, respectively. We have developed a multicast routing architecture that efficiently establishes distribution trees across wide area internets, where many groups will be sparsely represented. Efficiency is measured in terms of the router state, control message processing, and data packet processing, required across the entire network in order to deliver data packets to the members of the group. Our protocol independent multicast (PIM) architecture: a) maintains the traditional IP multicast service model of receiver-initiated membership, b) supports both shared and source-specific (shortest-path) distribution trees, c) is not dependent on a specific unicast routing protocol, and d) uses soft-state mechanisms to adapt to underlying network conditions and group dynamics. The robustness, flexibility, and scaling properties of this architecture make it well-suited to large heterogeneous internetworks.**

I. Introduction

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

The architecture proposed here complements existing multicast routing mechanisms such as those proposed by Deering in [9] and [10] and implemented in MOSPF [26] and distance vector multicast routing protocol (DVMRP) [29]. These traditional multicast schemes were intended for use within regions where a group is widely represented or bandwidth is universally plentiful. However, when group members, and senders to those group members, are distributed *sparsely* across a wide area, these schemes are not efficient. Data packets (in the case of DVMRP) or membership report information (in the case of MOSPF) are occasionally sent on links, and associated state is stored in routers, that do *not* lead to receivers or senders, respectively. The purpose of this work is to develop a multicast routing architecture that efficiently establishes distribution trees even when some or all members are sparsely distributed. Efficiency is measured in terms of the router 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.

A. Background

In the traditional IP multicast model, established by Deering [9], 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 nonmember, to send to a group. A group membership protocol [8] is used for routers to learn the existence of members on their directly attached subnetworks. This receiver-initiated joint 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 in 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 [24]. 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 source-specific, "shortest-path" tree (SPT) from the packet's source subnetwork to its destination group members.

Throughout this paper, when we use the term SPT, we mean shortest from the perspective of unicast routing. If the unicast routing metric is hop counts, then the branches of the multicast

Manuscript received February 8, 1995; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Partridge.

S. Deering is with Xerox PARC, Palo Alto, CA 94304 USA (e-mail: deering@parc.xerox.com).

D. L. Estrin is with the Computer Science Department/ISI, University of Southern California, Los Angeles, CA 90089 USA (e-mail: estrin@usc.edu).

D. Farinacci is with Cisco Systems Inc., San Jose, CA 95134 USA (e-mail: dino@cisco.com).

V. Jacobson is with the Lawrence Berkeley Laboratory, Berkeley, CA 94720 USA (e-mail: van@ee.lbl.gov).

C.-G. Liu is with the Computer Science Department, University of Southern California, Los Angeles, CA 90089 USA (e-mail: charley@catarina.usc.edu).

L. Wei is with Cisco Systems Inc., San Jose, CA 95134 USA (e-mail: lwei@cisco.com).

Publisher Item Identifier S 1063-6692(96)02719-7.

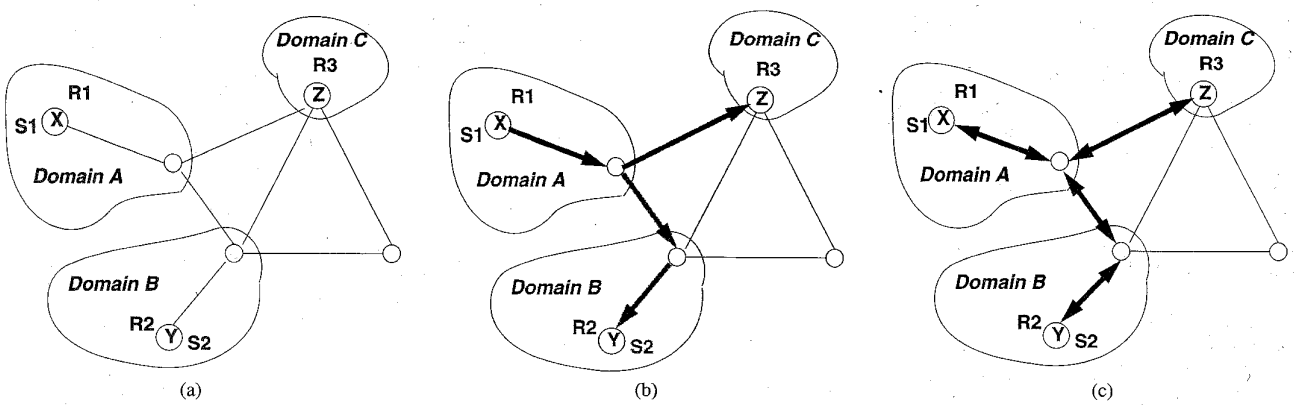


Fig. 1. Example of multicast trees.

SPT are minimum hop; if the metric is delay, then the branches are minimum delay. Moreover, in situations where paths are asymmetric, the multicast SPT's are actually reverse SPT's because we use unicast routings shortest path from the receiver to the source to build the branch of the distribution tree from the source to the receiver. Where route asymmetry results in poor quality distribution trees, it would be useful to obtain a shortest-path from route from unicast routing in order to build true SPT's.

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 SPT calculations performed to compute the delivery trees for all active multicast sources [25], 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) [7]. 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 of all interfaces except the incoming interface. Some schemes reduce the number of outgoing interfaces further by using the unicast routing protocol information to keep track of child-parent information [9], [29]. A special mechanism is used to avoid forwarding of data packets to leaf subnetworks with no members in that group (aka, truncated broadcasting). Also, if the arriving data packet does not come through the interface that the router uses to send packets to the source of the data packet, the data packet is silently dropped; thus the term RPF [7]. When a router attached to a leaf subnetwork receives a data packet addressed to a new group, if it finds no members present on its attached subnetworks, it will send a prune message upstream toward the source of the data packet. The prune messages prune the tree branches not leading to group members, thus resulting in a source-specific reverse-SPT 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 to the total number of destinations within the greater Internet, the number of destinations having group members of any particular *wide-area* group is likely to be small. In the case of distance-vector multicast schemes, routers that are not on the multicast delivery tree still have to carry the periodic truncated-broadcast of packets, and process the subsequent pruning of branches for all active groups. One protocol, DVMRP, has been deployed in hundreds of regions connected by the multicast backbone (MBONE) [18]. 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.

B. Extending Multicast to the Wide Area: Scaling Issues

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

Existing link–state and distance–vector multicast routing schemes have good scaling properties only when multicast groups densely populate the network of interest. When most of the subnets or links in the internetwork 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, regardless. The emphasis of our proposed work is to develop multicast protocols that will also efficiently support the sparsely distributed groups that are likely to be most prevalent in wide-area internetworks.

C. Overhead and Tree Types

The examples in Fig. 1 illustrate the inadequacies of the existing mechanisms. There are three domains that communicate via an internet. There is a member of a particular group, G, located in each of the domains. There are no other members of this group currently active in the internet. If a traditional IP multicast routing mechanism such as DVMRP is used, then, when a source in domain A starts to send to the

group, its data packets will be broadcast throughout the entire internet. Subsequently, all those sites that do not have local members will send prune messages and the distribution tree will stabilize to that illustrated with bold lines in Fig. 1(b). Periodically, however, the source's packets will be broadcast throughout the entire internet when the pruned-off branches time out.

Thus far, we have motivated our design by contrasting it to the traditional dense-mode IP multicast routing protocols. More recently, the core based tree (CBT) protocol [1] was proposed to address similar scaling problems. CBT uses a single delivery tree for each group, rooted at a "core" router and shared by all senders to the group. As desired for sparse groups, CBT does not exhibit the occasional broadcasting 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 Fig. 1(c) would be established. This distribution tree would also be used by sources sending from domains B and C. This would result in concentration of all the sources' traffic on the path indicated with bold lines. We refer to this as *traffic concentration.* This is a potentially significant issue with CBT, or any protocol that imposes a single shared tree per group for distribution of all data packets. 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 [30] proved that the bound on maximum delay of an optimal core-based tree (which he called a *center-based* tree) is two 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 a 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 SPT maximum delay in Fig. 2(a). For each node degree, we tried 500 different 50-node graphs with 10-member groups chosen randomly. It can be seen that the maximum delays of core-based trees with optimal core placement, are up to 1.4 times those of the SPT's. Note that although some error bars in the delay graph extend below one, there are no real data points below one (the distribution is not symmetric, for more details see [33]).

For interactive applications where low latency is critical, it is desirable to use the trees based on shortest-path routing 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) plots the measurements in networks with different node degrees. It is clear from this experiment that CBT exhibits greater traffic concentrations.

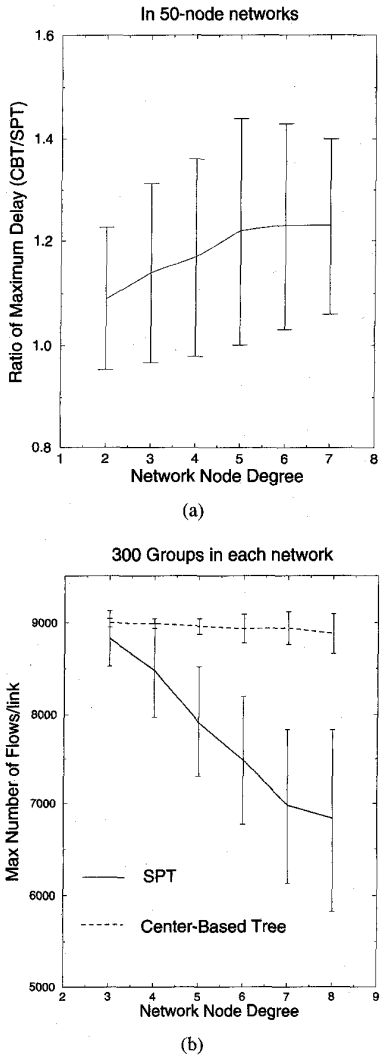


Fig. 2. Comparison of SPT's and center-based tree.

Despite the disadvantages of longer path length and traffic concentration, shared-tree schemes such as CBT (and PIM's shared tree) have the significant advantage of reduced multicast routing state. This is particularly true for applications that are not highly delay sensitive or data intensive.

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), a more complete analysis of these trade-offs can be found in [33]. 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 SPT's or a group-shared tree. The first-hop routers of the receivers can make this decision independently. A receiver could even choose different types of trees for different sources.

The capability to support different tree types is the fundamental difference between PIM and CBT. There are other significant protocol engineering differences as well. Two obvious engineering trade-offs are:

a) *Soft-State versus Explicit Reliability Mechanism:* CBT uses explicit hop-by-hop mechanisms to achieve reliable delivery of control messages. As described in the next section, PIM uses periodic refreshers 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.

b) *Incoming Interface Check on All Multicast Data Packets:* If multicast data packets loop, the result can be severe. Unlike unicast packets, multicast packets can fan out each time they loop. Therefore, we assert that all multicast data packets should be subject to an incoming interface check comparable to the one performed by DVMRP and MOSPF.

D. Paper Organization

In the remainder of this paper, we enumerate the specific design requirements for wide-area multicast routing (Section II), describe a specific protocol for realizing these requirements (Section III), and discuss open issues (Section IV).

II. 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 or 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 a hop-count limit or some other form of limiting the "scope" of multicast packet propagation, and c) the internetwork 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 SPT's are significantly shorter than in the shared tree.
- *Routing Protocol Independence:* The protocol should rely on existing unicast routing functionality to adapt to topology changes, but at the same time be independent of the particular protocol employed. We accomplish this by letting the multicast protocol make use of the unicast routing tables, independent of how those tables are computed.
- *Robustness:* The protocol should be capable of gracefully adapting 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.
- *Interoperability:* We require interoperability with traditional RPF and link-state multicast routing, both intra- 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 inter-operate with some intra-domain routing protocols. In support of this inter-operation 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.

III. PIM Protocol

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

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

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

b) Dense mode IP multicast tree construction is all data driven, PIM must use per-group *Rendezvous points* (RP) for receivers to "meet" new sources. Rendezvous points are used by senders to announce their existence and by receivers to learn about new senders of a group. Source-specific trees in PIM are data driven, however, and the RP-tree is receiver-join driven in anticipation of data.

The SPT state maintained in routers is of the same order 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

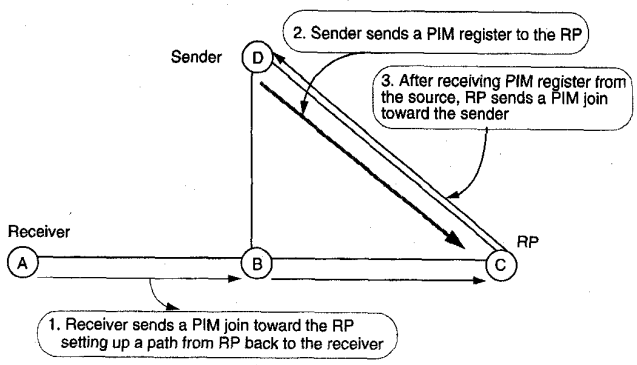


Fig. 3. How senders rendezvous with receivers.

interface set (oif), incoming interface (iif). We refer to this forwarding information as the *multicast forwarding entry* for (S, G). The oif's and iif's of (S, G) entries in all routers together form an SPT rooted at S.

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

Figure 3 shows a simple scenario of a receiver and a sender joining a multicast group via an RP. When the receiver signals that it wants to join a PIM multicast group (i.e., by sending an IGMP message [8]), its first hop PIM router (A in Fig. 3) sends a PIM-join message toward the RP advertised for the group. Processing of this message by intermediate routers sets up the multicast tree branch from the RP to the receiver. When sources start sending to the multicast group, the first hop PIM-router (D in Fig. 3) sends a PIM-register message, piggybacked on the data packet, to the RP's for that group. The RP responds by sending a join toward the source. Processing of these messages by intermediate routers (there are no intermediate routers between the RP and the source in Fig. 3) sets up a packet delivery path from the source to the RP.

If source-specific distribution trees are desired, the first hop PIM router for each member eventually joins the source-rooted distribution tree for each source by sending a PIM-join message toward the source. After data packets are received on the new path, router B in Fig. 3 sends a PIM-prune message toward the RP. B knows, by checking the incoming interface in its routing table, that it is at a point where the SPT 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 SPT has completed. This provides a smooth transition, e.g., there is no loss of data packets.

An RP is 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 addresses associated with the group in question. The mapping information may be configured, derived algorithmically, or may be learned through another protocol mechanism.

PIM avoids explicit enumeration of receivers, but does require enumeration of sources. If there are very large numbers of sources sending to a group but the sources' average data rates are low, then one possibility is to support the group with a shared tree, which has less per-source overhead. If SPT's are desired, then when the number of sources grows very large, some form of aggregation or proxy mechanism will be needed; see Section IV. We selected this trade-off because in many existing and anticipated applications, the number of receivers is much larger than the number of sources. And when the number of sources is very large, the average data rate tends to be lower (e.g., resource discovery).

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

A. Local Hosts Joining a Group

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

When a *designated router* (DR) receives a report for a new group G, it checks to see if it has RP addresses associated with G. The mechanism for learning this mapping of G to RP's is somewhat orthogonal to the specification of this protocol, however, we require some mechanism in order for the protocol to work. For the purposes of this description, we assume that each DR listens to a "well-known" multicast group to obtain the group-address (or group-address-range) to RP mappings for all multicast groups.

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

B. Establishing the RP-Rooted Shared Tree

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

Each upstream router creates or updates its multicast forwarding entry for $(*, G)$ when it receives a PIM-join with the WC and RP bits set. The interface on which the PIM-join

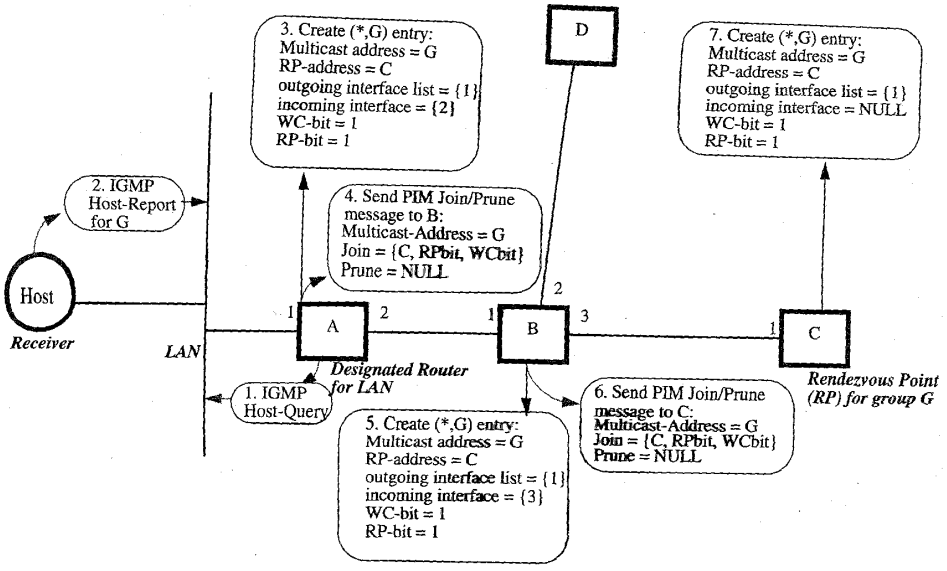


Fig. 4. Example: How a receiver joins and sets up shared tree. Actions are numbered in the order they occur.

message arrived is added to the list of outgoing interfaces for $(*, G)$. Based on this entry, each upstream router between the receiver and the RP sends a PIM-join message in which the join list includes the RP. The packet payload contains multicast-address $= G$, PIM-join $=$ RP, RPbit, WCbit, PIM-prune $=$ NULL.

The RP recognizes its own address and does not attempt to send join messages for this entry upstream. The incoming interface in the RP's $(*, G)$ entry is set to null.

C. Switching from Shared Tree (RP Tree) to SPT

When a PIM-router with directly-connected members receives packets from a source via the shared RP-tree, the router can switch to a source-specific tree. We refer to the source-specific tree as an SPT, however, if unicast routing is asymmetric, the resulting tree is actually a reverse-SPT. As shown in Fig. 5, router A initiates a new multicast forwarding entry for the new source, Sn which, in turn, triggers a join message to be sent toward Sn with Sn in the join list. The newly-created Sn, G forwarding entry is initialized with the SPT bit cleared, indicating that the SPT branch from Sn has not been completely setup. This allows the router to continue to accept packets from Sn via the shared tree until packets start arriving via the source specific tree. A timer is set for the (Sn, G) entry.

A PIM-join message will be sent upstream to the best next hop toward the new source, Sn, with Sn in the join list: multicast-address $= G$, PIM-join $= Sn$, PIM-prune $=$ NULL. The best next hop is determined by the unicast routing protocol.

When a router that has an (Sn, G) entry with the SPT bit cleared starts to receive packets from the new source Sn on the interface used to reach Sn, it sets the SPT-bit. The router will send a PIM-prune toward the RP if its shared tree incoming interface differs from its SPT incoming interface, indicating that it no longer wants to receive packets from Sn via the RP tree. In the PIM message toward the RP, it includes Sn in the prune list, with the WC-bit set indicating that a negative cache should be set up on the way to the RP. A negative cache entry is an (S, G) entry with null outgoing interface list. Data packets matching the negative cache are discarded silently.

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

Note that a DR may adopt a policy of not setting up a (S, G) entry (and therefore, not sending a PIM-join message toward the source) until it has received m data packets from the source within some interval of n seconds. This would eliminate the overhead of (S, G) state upstream when small numbers of packets are sent sporadically (at the expense of data packet delivery over the suboptimal paths of the shared RP tree). The DR may also choose to remain on the RP-distribution tree indefinitely instead of moving to the SPT. Note that if the DR does join the SPT, the path changes for all directly connected and downstream receivers. As a result, we do not guarantee that a receiver will remain on the RP tree; if receiver A's RP tree overlaps with another receiver B's SPT, receiver A may receive its packets over the SPT. A multicast distribution tree is a resource shared by all members of the group. To satisfy individual receiver-specific requirements or policies the multicast tree might degenerate into a set of receiver-specific unicast paths.

D. Steady-State Maintenance of Router State

In the steady state, each router sends periodic refreshers of PIM messages upstream to each of the next hop routers that is en route to each source, $(S, *)$ for which it has a multicast

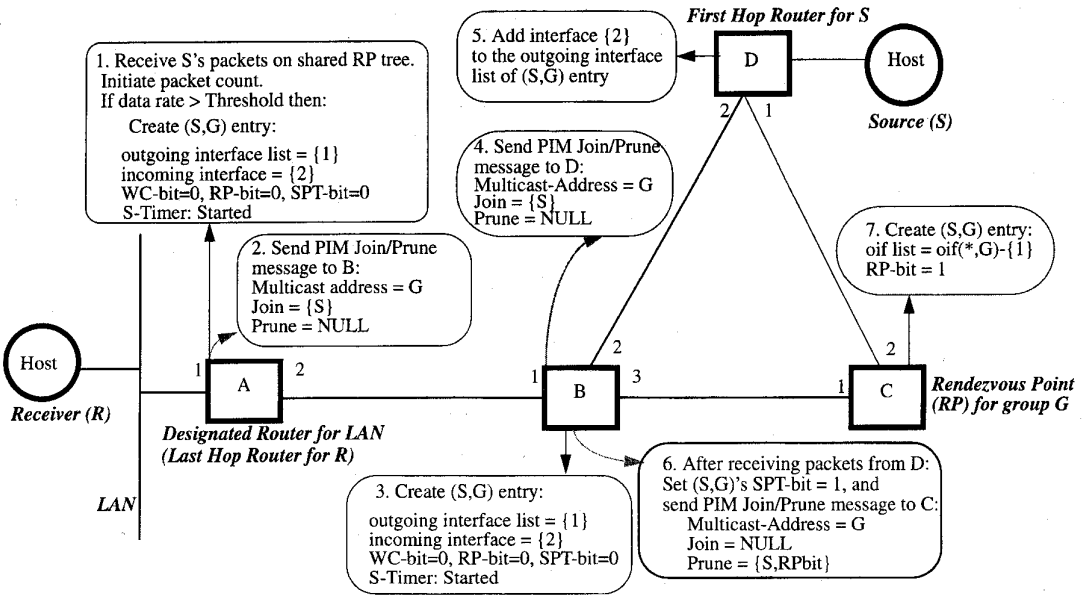


Fig. 5. Example: Switching from shared tree to SPT. Actions are numbered in the order they occur.

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

E. Multicast Data Packet Processing

Data packets are processed in a manner similar to existing multicast schemes. An incoming interface check is performed and if it fails, the packet is dropped, otherwise the packet is forwarded to all the interfaces listed in the outgoing interface list (whose timers have not expired). There are two exception actions that are introduced if packets are to be delivered continuously, even during the transition from a shared to SPT.

1) When a data packet matches on an (S, G) entry with a cleared SPT bit, if the packet does not match the incoming interface for that entry, then the packet is forwarded according to the *, G entry, i.e., it is sent to the outgoing interfaces listed in *, G *if* the incoming interface matches that of the *, G. The *, G RPF check is needed because the packet should be dropped if it does not pass the RPF check of either the *, G or Sn, G entry. The iif of the *, G entry points toward the RP.
2) When a data packet matches on an (S, G) entry with a cleared SPT bit, *and* the incoming interface of the packet matches that of the (S, G) entry, then the packet is forwarded and the SPT bit is set for that entry.

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

F. Timers

A timer is maintained for each outgoing interface listed in each (S, G) or *, G entry. The timer is set when the interface is added. The timer is reset each time a PIM-join message is received on that interface for that forwarding entry [i.e., (S, G) or $(*, G)$]. Recall that all PIM, control messages are periodically refreshed.

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

G. PIM Routers on Multiaccess Subnetworks

Certain multiaccess subnetwork configurations require special consideration. When a local area network (LAN)-connected router receives a prune from the LAN, it must detect whether there remain other downstream routers with active downstream members. The following protocol is used when a router whose incoming interface is the LAN has all of its outgoing interfaces go to null, the router multicasts a prune message for (S, G) onto the LAN. All other routers hear this prune and if there is any router that has the LAN as its incoming interface for the same (S, G) and has a non-null outgoing interface list, then the router sends a join message onto the LAN to override the prune. The join and prune should go to a single upstream router that is the right previous hop to the source or RP; however, at the same time we want others to hear the join and prune so that they suppress their own

joins/prunes or override the prune. For this reason, the join is sent to a special multicast group of which all routers on the same LAN (and only those on the same LAN) are members. The IP address of the intended recipient of the message is included in the IGMP header.

H. Unicast Routing Changes

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

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

I. Protocol Summary

In summary, once the PIM-join messages have propagated upstream from the RP, data packets from the source will follow the (S, G) distribution path state established. The packets will travel to the receivers via the distribution paths established by the PIM-join messages sent upstream from receivers toward the RP. Multicast packets will arrive at some receivers before reaching the RP if the receivers and the source are both "upstream" from 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 in order to reach new receivers. Similarly, receivers continue to receive some data packets via the RP tree in order to pick up new senders. However, when source-specific tree distribution is used, most data packets will arrive at receivers over a shortest path distribution tree.

IV. Open Issues

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

- *Aggregation of Information in PIM:* One of the most significant scaling issues faced by PIM and other known multicast routing schemes is the amount of memory consumed by multicast forwarding entries as the number of active sources and groups grows.

  The most straightforward approach for reducing source-specific state is to aggregate across source addresses, for example by 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 sources where more specific routes exist, there will be a large fanout, and PIM messages will travel toward all members of the aggregate, which would be inefficient in most cases.

  On closer consideration, it seems that source-specific state might not be the dominant concern. In PIM, as well as other multicast schemes such as DVMRP, source-specific state is created in a data-driven manner. Moreover, in PIM, source-specific state is only created when the source's data rate exceeds some threshold. Therefore, we know that the amount of source-specific state can not grow without bound, because the amount of available bandwidth, and therefore the number of active sources, is bounded. In fact, the number of simultaneously-active sources is not just bounded by the capacity of the links (which may be quite large in the future), but by the limited input capacity of the members of the group (which is growing but not at the same rate as backbone link bandwidth, for example).

  Of greater concern is the potential explosion of simultaneously-active multicast groups, and the associated group-specific state. Unlike source-specific trees, group-specific shared trees are not built or maintained in a data-driven manner and therefore are not subject to the same bounds described above. Two approaches to group-specific state reduction are under consideration. Both are targeted for central backbone regions of the network where group-specific state proliferation is of most concern. In the first, a region does not maintain group-specific shared tree state in the absence of data traffic. Instead, only the border routers of the region retain group specific state, and only when data packets arrive for a particular group is routing state built inside of the region. In effect, the region emulates dense mode behavior. To carry this out, border routers must still maintain group-specific state in order to stay on the shared group tree, and PIM-join messages must still be propagated across the region to reach the border routers on the other side. In other words, state reduction can be reduced for low duty-cycle groups, however, control messaging is not affected. In the second approach for group-specific state reduction, a region can aggregate (S, G) entries into $(S, *)$ or (S, group-range) entries. This approach appears quite promising, particularly when (S, G) entries are only aggregated when their outgoing interface lists are the same.
- *Interaction with Policy-Based and TOS Routing:* PIM messages and data packets may travel over policy-constrained routes to the same extent that unicast routing does, so long as the policy does not prohibit this traffic explicitly.

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

  If the path to reach the RP, or some source, is indicated as having the appropriate quality of service (QoS), and as being symmetric, then a PIM router can forward its joins upstream and expect that the data packets will be allowed to travel downstream. This implies that BGP/IDRP [20],

[28] 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 [16] 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 [36]:* Many interesting opportunities and issues arise when PIM-style explicit join multicast routing is used to support reservations, particularly, receiver-oriented reservations.

  For example, RSVP reservation messages travel from receivers toward sources according to the state that multicast routing installs. When a reservation is shared among multiple sources (e.g., a shared audio channel where there is generally only one or two speakers at a time), it is appropriate to set up the reservation on the shared, RP-tree. However, for source-specific reservations (e.g., video channels), one wants to avoid establishing them over the shared tree if, shortly thereafter, receivers are going to switch to a source-specific tree. In this situation, routing could be configured to not send source-specific reservations over a shared-tree, for example.

  Another interesting issue involves the need for alternate path routing when and if reservation requests are denied due to insufficient resources along the route that unicast routing considers to be best. To support this situation, PIM should be updated to allow explicit routing (i.e., often referred to as source routing) of PIM-join messages so that the reservation may be attempted along an alternate branch.

V. Conclusion

We have presented a solution to the problem of routing multicast packets in large, wide area internets. Our approach uses 1) constrained, receiver-initiated, membership advertisement for sparsely distributed multicast groups, 2) supports both shared and shortest path tree types in one protocol, 3) does not depend on the underlying unicast protocols, and 4) uses soft-state mechanisms to reliably and responsively maintain multicast trees. The architecture accommodates graceful and efficient adaptation to different network conditions and group dynamics.

A prototype of PIM has been implemented using extensions to existing IGMP message types. Simulation and implementation efforts conducted characterize configuration criteria and deployment issues. A complete specification document is available as an IETF internet-draft.

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

References

[1] A. J. Ballardie, P. F. Francis, and J. Crowcroft, "Core based trees," in *Proc. ACM SIGCOMM*, San Francisco, 1993.

[2] M. W. Bern and R. L. Graham, "The shortest-network problem," *Scientific American*, pp. 84–89, Jan. 1989.

[3] C. Topolcic (Ed.), *Experimental Internet Stream Protocol:* Version 2 (st-ii), RFC1190, Oct. 1990.

[4] B. Cain, A. Thyagarajan, and S. Deering, *Internet Group Management Protocol*, Version 3, Working draft, July 1995.

[5] S. Casner, "Second ietf internet audiocast," in *Internet Society News*, vol. 1, no. 3, p. 23, 1992.

[6] D. D. Clark, "The design philosophy of the darpa internet protocols," in *Proc. ACM SIGCOMM*, 1988.

[7] Y. K. Dalal and R. M. Metcalfe, "Reverse path forwarding of broadcast packets," *Commun. ACM*, vol. 21, no. 12, pp. 1040–1048, 1978.

[8] S. Deering, *Host Extensions for IP Multicasting*, RFC1112, Aug. 1989.

[9] ——, "Multicast Routing in a Datagram Internetwork," *Ph.D. Thesis*, Stanford University, 1991.

[10] S. Deering and D. Cheriton, "Multicast routing in a datagram internetworks and extended lans," in *ACM Trans. Comput. Syst.*, pp. 85–111, May 1990.

[11] S. Deering, D. Estrin, D. Farinacci, and V. Jacobson, *Protocol Independent Multicast (PIM), Dense Mode Protocol: Specification*, Internet Draft, Mar. 1994.

[12] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei, *Protocol Independent Multicast (PIM): Motivation and Architecture*, Working Draft, Nov. 1994.

[13] ——, *Protocol Independent Multicast (PIM): Specification*, Working Draft, Nov. 1994.

[14] ——, *Protocol Independent Multicast (PIM), Sparse Mode Protocol: Specification*, Working Draft, Sept. 1995.

[15] M. Doar and I. Leslie, "How bad is naive multicast routing," in *Proc. IEEE INFOCOM'93*, 1993.

[16] D. Estrin, T. Li, Y. Rekhter, and D. Zappala, *Source Demand Routing Protocol: Packet Format and Forwarding Specification*," IETF Working Draft, Mar. 1995.

[17] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," in *Proc. ACM SIGCOMM*, 1995.

[18] R. Frederick, "Left audio & videocast," *Internet Society News*, vol. 1, no. 4, p. 19, 1993.

[19] E. N. Gilbert and H. O. Pollak, "Steiner minimal trees," *SIAM J. Applied Mathematics*, vol. 16, no. 1, pp. 1–29, Jan. 1968.

[20] S. Hares and J. Scudder, *Idrp for IP*, Working Draft, Sept. 1993.

[21] R. M. Karp, *Reducibility Among Combinatorial Problem.* New York: Plennum, 1972.

[22] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, pp. 141–145, 1981.

[23] G. Malkin, *RIP Version 2 Carrying Additional Information*, RFC1388, June 1993.

[24] J. Moy, *OSPF Version 2*, RFC1247, Oct. 1991.

[25] ——, *MOSPF: Analysis and Experience*, Mar. 1994, RFC1585.

[26] ——, *Multicast Extension to OSPF*, RFC1584, Mar. 1994.

[27] V. J. Rayward–Smith and A. Clare, *On Finding Steiner Vertices*, Networks, vol. 16, pp. 284–294, 1986.

[28] Y. Rekhter and T. Li (Eds.), *A Border Gateway Protocol 4 (BGP-4)*, RFC1771, Mar. 1995.

[29] D. Waitzman, S. Deering, and C. Partridge, *Distance Vector Multicast Routing Protocol*, RFC1075, Nov. 1988.

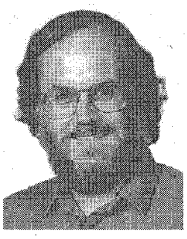
[30] D. Wall, "Mechanisms for broadcast and selective broadcast," Ph.D. thesis, Stanford University, Stanford, CA, Tech. Rep., no. 190, June 1980.

[31] B. M. Waxman, "Routing of multipont connections," *IEEE J. Select. Areas Commun.*, vol. 6, no. 9, Dec. 1988.

[32] L. Wei and D. Estrin, "A comparison of multicast trees and algorithms," Computer Science Department, University of Southern California, Tech. Rep. USC-CS-93-560, Sept. 1993.

[33] ——, "The trade-offs of multicast trees and algorithms," in *Proc. 1994 Int. Conf. Comput. Commun. Networks*, San Francisco, CA, Sept. 1994.

[34] L. Wei, F. Liaw, D. Estrin, A. Rowmano, and T. Lyon, "Analysis of a resequencer model for multicast over ATM networks," in *3rd Int. Workshop Network Operating Syst. Support Digital Audio Video*, San Diego, CA, Nov. 1992.
[35] P. Winter, "Steiner problem in networks: a survey," in *Networks*, vol. 17, no. 2, pp. 129–167, 1987.
[36] L. Zhang, R. Braden, D. Estrin, S. Herzog, and S. Jamin, *Resource Reservation Protocol (RSVP)*, Version 1, Functional Specification, IETF Working Draft, July 1995.



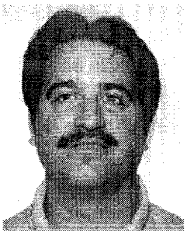
Stephen Deering (S'84–M'87) received the B.Sc. and M.Sc. degrees from the University of British Columbia, Canada, in 1973 and 1982, and the Ph.D. degree from Stanford University, Stanford, CA, in 1991, respectively.

He is currently a member of the research staff at Xerox PARC, engaged in research on advanced internetwork technologies, including multicast routing, mobile internetworking, scalable addressing, and support for multimedia applications over the Internet. He is present or past chair of numerous Working Groups of the Internet Engineering Task Force (IETF), a co-founder of the Internet Multicast Backbone (MBone), and the lead designer of the next generation Internet Protocol, IPv6.

Deborah L. Estrin (S'78–M'80–SM'95) received the Ph.D. degree in 1985 and the M.S. degree in 1982 from the Massachusetts Institute of Technology and the B.S. degree in 1980 from the University of California, Berkeley.

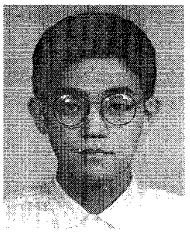
She is currently an Associate Professor of Computer Science at the University of Southern California, Los Angeles, where she joined the faculty in 1986. Estrin is a co-PI on the National Science Foundation (NSF) Routing Arbiter project at USC's Information Sciences Institute. She co-chairs the Source Demand Routing Working Group of the IETF and is an active participant in the Inter-Domain Multicast Routing and RSVP working groups. Estrin is a member of the ACM and AAAS. She has served on several panels for the NSF and National Research Council/CSTB, and is currently a member of ARPA's ISAT. Estrin was one of the founding Editors of Wiley's *Journal of Internetworking Research and Experience* and is currently an editor of the ACM/IEEE TRANSACTIONS ON NETWORKS.

Dr. Estrin received the NSF Presidential Young Investigator Award for her research in network interconnection and security in 1987.



Dino Farinacci has worked on transport and network layer protocols for 15 years. Currently, he works for Cisco Systems, Inc., San Jose, CA, where he has been designing and implementing IP and OSI routing protocols. He is a member of the IPng Directorate of the IETF where he has been involved in prototyping IP next generation proposals. In the last couple of years, he has been involved in the design, implementation, and deployment of IP multicast routing technology, notably PIM and DVMRP. Dino has been an active member of the Internet Engineering Task Force (IETF) for six years.

Van Jacobson photo and biography not available time of publication.



Ching-Gung Liu (ACM S'95) received the M.S. degree from the University of Southern California, Los Angeles, in 1991 and the B.S. degree from National Taiwan University, China, in 1988. He joined the Ph.D. program at the University of Southern California and started working on the design and implementation of PIM in 1993.

Liu is currently working on scalable reliable multicast protocol toward the Ph.D. degree.

Liming Wei received the Ph.D. and M.S. degrees from the University of Southern California in 1995 and 1990 and the B.S. degree from Xian JiaoTong University, China, in 1985.

He currently works in the internetwork operating systems (IOS) division of Cisco Systems, Inc., San Jose, CA. His research interests include the design and evaluations of multicast routing mechanisms, and transport protocol performance. He developed the first version of PIMSIM, a packet level protocol simulator for PIM, and is now working on the realization of ubiquitous multicast routing services.