Inter-Domain Multicast Routing (IDMR)
Internet Engineering Task Force
INTERNET-DRAFT
draft-thaler-interop-00.ps

D. Thaler
U. Michigan

November 11, 1996
Expires: May, 1997

Interoperability Rules for Multicast Routing Protocols

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 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."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

The rules described in this document will allow efficient interoperation among multiple independent multicast routing domains. Specific instantiations of these rules are given for the DVMRP, MOSPF, PIM-DM, PIM-SM, and CBT multicast routing protocols.

1 Introduction

To allow sources and receivers inside multiple autonomous multicast routing domains (or "regions") to communicate, the regions must be connected by multicast border routers (MBRs). To prevent black holes or routing loops among regions, we assume that these regions are organized into one of the following topologies:

- A tree (or star) topology (figure 1) with a backbone region at the root, stub regions at the leaves, and possibly "transit" regions as branches between the root and the leaves. Each pair of adjacent regions is connected by one or more MBRs. The root of each subtree of regions receives all globally-scoped traffic originated anywhere within the subtree, and forwards traffic to its parent and children where needed. Each parent region's MBR injects a default route into its child regions, while child regions' MBRs inject actual (but potentially aggregated) routes into parent regions. Thus, the arrows in the figure indicate both the direction in which the default route points, as well as the direction in which all globally-scoped traffic is sent.

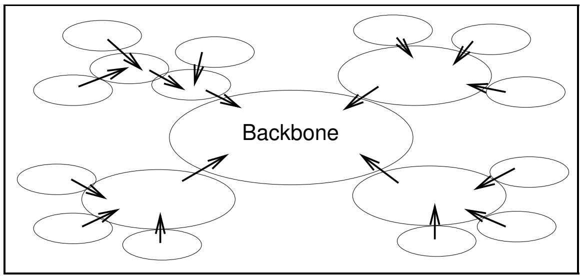

Figure 1: Tree Topology of Regions

- An arbitrary topology, in which a higher level (inter-domain) routing protocol, such as HDVMRP [1], is used to calculate paths among regions. Each pair of adjacent regions is connected by one or more MBRs. In this scheme, external routes are not known with regions. Instead, the default route point towards the closest MBR. In addition, all globally-scoped traffic must reach all of the originating region's MBRs.

Section 2 describes rules allowing interoperability between existing multicast routing protocols [2, 3, 4, 5, 6], and reduces the interoperability problem from $O(N^2)$ ($N-1$ per protocol) potential protocol interactions, to just N (1 per protocol) instantiations of the same set of invariant rules. This document specifically applies to Multicast Border Routers (MBRs) which meet the following assumptions:

1. The MBR consists of two or more active routing components, each running an instance of some routing protocol. No assumption is made about the type of routing protocol (e.g., broadcast-and-prune or explicit-join, distance-vector or link-state) any component runs, or the nature of a "component". Multiple components running the same protocol are allowed.

2. The router is configured to forward packets between two or more independent regions. The router has one or more active interfaces in each region, and one component per region.

3. Only one multicast routing protocol is active per interface (we do not consider mixed multicast protocol LANs). Each interface on which multicast is enabled is thus "owned" by exactly one of the components.

4. All components share a common forwarding cache of (S,G) entries, which are created when data packets are received, and can be deleted at any time. Only the component owning an interface may change information about that interface in the forwarding cache. Each forwarding cache entry has a single incoming interface (iif) and a list of outgoing interfaces (oiflist). Each component typically keeps a separate routing table with any type of entries.

Note that the guidelines in this document are implementation-independent. The same rules given in Section 2 apply in some form, regardless of the implementation. For example, they apply to each of the following architectural models:

Single process (e.g. gated): Several routing components in the same user-space process, running on top of a multicast-capable kernel.

Multiple peer processes: Several routing components, each as a separate user-space process, all sitting on top of a multicast-capable kernel, with $N * (N - 1)$ interaction channels.

Multiple processes with arbiter: Multiple independent peer routing component processes which interact with each other and with the kernel solely through an independent arbitration daemon.

Monolith: Several routing components which are part of the "kernel" itself.

We describe all interactions between components in terms of *alerts*. The nature of an *alert* is implementation dependent (e.g., it may consist of a simple function call, writing to shared memory, use of IPC, or some other method) but alerts of some form exist in every model. Similarly, the originator of an alert is also implementation-dependent; for example, alerts may be originated by a component effecting a change, by an independent arbiter, or by the kernel.

2 Requirements

To insure that a MBR fitting the above assumptions exhibits correct interdomain routing behavior, each MBR component MUST adhere to the following rules:

Rule 1 *All components must agree on which component owns the incoming interface (iif) for a forwarding cache entry.*

When a multicast routing change occurs, causing the iif to change to an interface owned by a different component, both the component previously owning the entry's iif and the component afterwards owning the entry's iif MUST notice the change (so the first can prune upstream and the second can join/graft upstream, for example). Typically, noticing such changes will happen as a result of normal protocol behavior.

Rule 2 *The component owning an interface specifies the criteria for which packets received on that interface are to be accepted or dropped (e.g., whether to perform an RPF check, and what scoped boundaries exist on that interface). Once a packet is accepted, however, it is processed according to the forwarding rules of* all *components.*

Rule 3 *Whenever a new (S,G) forwarding cache entry is created (upon accepting a packet destined to a non-local group), all components MUST be alerted [(S,G) Creation alert] so that they can set the forwarding state on their own outgoing interfaces (oifs) before the packet is forwarded.*

Note that (S,G) Creation alerts are not necessarily generated by one of the protocol components themselves.

Rule 4 *When a component removes the last oif from an (S,G) forwarding cache entry whose iif is owned by another component, the component owning the iif MUST be alerted [(S,G) Prune alert] (so it can send a prune, for example).*

Rule 5 *When the first oif is added to an (S,G) forwarding cache entry whose iif is owned by another component, the component owning the iif MUST be alerted [(S,G) Join alert] (so it can send a join or graft, for example).*

The oif list in rules 4 and 5 must also logically include any virtual encapsulation interfaces such as those used for tunneling or for sending encapsulated packets to an RP/core.

Rule 6 *Unless a component reports the aggregate group membership in the direction of its interfaces, it MUST be a* wildcard receiver *for all sources whose RPF interface is owned by another component ("externally-reached" sources).*

In addition, a component MUST be a wildcard receiver *for all sources whose RPF interface is owned by that component ("internally-reached" sources) if any other component of the MBR is a wildcard receiver for externally-reached sources.*

For example, if the backbone does not keep global membership information, all MBR components in the backbone in a tree topology of regions, as well as all components owning the RPF interface towards the backbone are wildcard receivers for externally-reached sources.

MBRs need not be wildcard receivers (for internally- or externally-reached sources) if a higher-level routing protocol, such as HDVMRP, is used for routing between domains.

2.1 Deleting Forwarding Cache Entries

Special care must be taken to follow Rules 4 and 5 when forwarding cache entries can be deleted at will. Specifically, a component must be able to determine when the combined oiflist for (S,G) goes from null to non-null, and vice versa.

This can be done in any implementation-specific manner, including, but not limited to, the following possibilities:

- Whenever a component would modify the oiflist of a single forwarding cache entry if one existed, one is first created. The oiflist is then modified and Rules 4 and 5 applied *after* an (S,G) Creation alert is send to all components and all components have updated the oiflist. Or,
- When a forwarding cache entry is to be deleted, a new alert [(S,G) Deletion alert] is sent to all components, and the entry is only deleted if all components then grant permission. Each component could then grant permission only if it had no (S,G) route table entry.

2.2 Additional Recommendation

Using (*,G) Join alerts and (*,G) Prune alerts can reduce bandwidth usage by avoiding broadcast-and-prune behavior among regions when it is unnecessary. This optimization requires that each component be able to determine which other components are interested in any given group.

Although this may be done in any implementation-dependent method, one example would be to maintain a common table (which we call the Component-Group Table) indexed by group-prefix, listing which components are interested in each group(prefix). Thus, any components which are wildcard receivers for *externally*-reached sources (i.e., those whose RPF interface is owned by another component) would be listed in all entries of this table, including a default entry. This table is thus loosely analogous to a forwarding cache of (*,G) entries, except that no distinction is made between incoming and outgoing interfaces.

3 DVMRP

In this section we describe how the rules in section 2 apply to DVMRP. We assume that the reader is familiar with normal DVMRP behavior as specified in [2].

As with all broadcast-and-prune protocols, DVMRP components are automatically wildcard receivers for internally-reached sources. Unless some form of Regional-Membership-Reports (RMRs) (described in [1]) are added to DVMRP in the future, all DVMRP components also act as wildcard receivers for externally-reached sources. If RMRs are available for the region, then a DVMRP component acts as a wildcard receiver for externally-reached sources only if internally-reached regions exist which do not support some form of RMRs.

3.1 Generating Alerts

A (S,G) Prune alert is sent to the component owning the iif for a forwarding cache entry whenever the last oif is removed from the entry, and the iif is owned by another component. In DVMRP, this may happen when:

- A DVMRP (S,G) Prune message is received on the logical interface.

A (S,G) Join alert is sent to the component owning the iif for a forwarding cache entry whenever the first logical oif is added to an entry, and the iif is owned by another component. In DVMRP, this may happen when any of the following occur:

- The oif's prune timer expires, or

- A DVMRP Graft message is received on the logical interface, or
- IGMP [7] notifies DVMRP that directly-connected group members now exist on the interface.

3.2 Processing Alerts

When a DVMRP component receives a (S,G) Creation alert, all the component's interfaces are added to the entry's oif list (according to normal DVMRP behavior) EXCEPT:

- the iif,
- leaf networks without local members of the entry's group,
- and interfaces with scoped boundaries covering the group.

When a DVMRP component receives a (S,G) Prune alert, and the forwarding cache entry's oiflist is empty, it sends a DVMRP (S,G) Prune message to the upstream neighbor according to normal DVMRP behavior.

When a DVMRP component receives a (*,G) Prune alert, it is treated as if a (S,G) Prune alert were received for every existing (S,G) route entry for the group G.

When a DVMRP component receives a (S,G) Join alert, and a prune was previously sent upstream, it sends a DVMRP (S,G) Graft message to the upstream neighbor according to normal DVMRP behavior.

When a DVMRP component receives a (*,G) Join alert, it is treated as if a (S,G) Join alert were received for every existing (S,G) route entry for the group G.

4 MOSPF

In this section we describe how the rules in section 2 apply to MOSPF. We assume that the reader is familiar with normal MOSPF behavior as specified in [3]. We note that MOSPF allows joining and pruning entire groups, but not individual sources within groups.

Although interoperability between MOSPF and dense-mode protocols (such as DVMRP) is specified in [3], we describe here how an MOSPF implementation may interoperate with all other multicast routing protocols.

An MOSPF component acts as a wildcard receiver for internally-reached sources if and only if any other component is a wildcard receiver for externally-reached sources. An MOSPF component acts as a wildcard receiver for externally-reached sources only if internally-reached regions exist which do not support some form of Regional-Membership-Reports (RMRs) (described in [1]). Since MOSPF floods membership information throughout the domain, MOSPF itself is considered to support a form of RMRs natively.

4.1 Generating Alerts

When it is known that there are no longer any members of a group G in the MOSPF region, and exactly one other component wants to receive data for G, a (*,G) Prune alert is sent to that component. When it is known that there are no longer any members of a group G in the MOSPF region, and no other components want to receive data for G, a (*,G) Prune alert is sent to *all* other components. In MOSPF, these may happen when either:

- IGMP notifies MOSPF that there are no longer any directly-connected group members on an interface, or
- Any router's group-membership-LSA for G is aged out.

When it is first known that there are members of a group G in the MOSPF region, and exactly one other component wants data for G, a (*,G) Join alert is sent to that component. When it is first known that there are members of a group G in the MOSPF region, any no other component wants data for G, a (*,G) Join alert is sent to *all* other components. In MOSPF, these may happen when any of the following occur:

- IGMP notifies MOSPF that directly-connected group members now exist on the interface, or
- A group-membership-LSA is received for G.

4.2 Processing Alerts

When an MOSPF component receives a (S,G) Creation alert, it calculates the shortest path tree for the MOSPF region, and adds the downstream interfaces to the entry's oif list according to normal MOSPF behavior.

When an MOSPF component receives a (S,G) Prune alert, the alert is ignored, since MOSPF can only prune entire groups at a time.

When an MOSPF component receives a (*,G) Prune alert, and there are no directly-connected members on any MOSPF interface, the router "prematurely ages" out its group-membership-LSA for G in the MOSPF region according to normal MOSPF behavior.

When a MOSPF component receives either a (S,G) Join alert or a (*,G) Join alert, and G was not previously included in the router's group-membership-LSA (and the component is not a wild-card multicast receiver), it originates a group-membership-LSA in the MOSPF region according to normal MOSPF behavior.

5 PIM-DM

In this section we describe how the rules in section 2 apply to Dense-mode PIM. We assume that the reader is familiar with normal PIM-DM behavior as specified in [6].

As with all broadcast-and-prune protocols, PIM-DM components are automatically wildcard receivers for internally-reached sources. Unless some form of Regional-Membership-Reports (RMRs) (described in [1]) are added to PIM-DM in the future, all PIM-DM components also act as wildcard receivers for externally-reached sources. If RMRs are available for the region, then a PIM-DM component acts as a wildcard receiver for externally-reached sources only if internally-reached regions exist which do not support some form of RMRs.

5.1 Generating Alerts

A (S,G) Prune alert is sent to the component owning the iif for a forwarding cache entry whenever the last oif is removed from the forwarding cache entry, and the iif is owned by another component. In PIM-DM, this may happen when:

- A PIM (S,G) Join/Prune message with S in the prune list is received on a point-to-point interface.
- The Oif-Timer in an (S,G) route table entry expires.
- A PIM (S,G) Assert message from a preferred neighbor is received on the interface.

A (S,G) Join alert is sent to the component owning the iif for a forwarding cache entry whenever the first oif is added to an entry, and the iif is owned by another component. In PIM-DM, this may happen when any of the following occur:

- The oif's prune timer expires, or
- A PIM-DM (S,G) Graft message is received on the interface, or
- IGMP notifies PIM-DM that directly-connected group members now exist on the interface.

5.2 Processing Alerts

When a PIM-DM component receives a (S,G) Creation alert, all the component's interfaces are added to the entry's oif list (according to normal PIM-DM behavior) EXCEPT:

- the iif,
- leaf networks without local members of the entry's group,
- and interfaces with scoped boundaries covering the group.

When a PIM-DM component receives a (S,G) Prune alert, and the forwarding cache entry's oiflist is empty, it sends a PIM-DM (S,G) Prune message to the upstream neighbor according to normal PIM-DM behavior.

When a PIM-DM component receives a (*,G) Prune alert, it is treated as if a (S,G) Prune alert were received for every existing (S,G) forwarding cache entry for the group G.

When a PIM-DM component receives a (S,G) Join alert, and a prune was previously sent upstream, it sends a PIM-DM (S,G) Graft message to the upstream neighbor according to normal PIM-DM behavior.

When a PIM-DM component receives a (*,G) Join alert, it is treated as if a (S,G) Join alert were received for every existing (S,G) route entry for the group G.

6 PIM-SM

In this section we describe how the rules in section 2 apply to Sparse-mode PIM. We assume that the reader is familiar with normal PIM-SM behavior, as specified in [4].

To achieve correct PIM-SM behavior within the region, the PIM-SM region MUST be convex so that Bootstrap messages reach all routers in the region. That is, the shortest-path route from any internal router to any other internal router must lie entirely within the PIM region.

Unless some form of Regional-Membership-Reports (RMRs) (described in [1]) are added to PIM-SM in the future, all PIM-SM components act as wildcard receivers for externally-reached sources. If RMRs are available for the region, then a PIM-SM component acts as a wildcard receiver for externally-reached sources only if internally-reached regions exist which do not support some form of RMRs.

A PIM-SM component acts as a wildcard receiver for internally-reached sources if and only if any other component is a wildcard receiver for externally-reached sources. It does this by periodically sending (*,*,RP) Joins to all RPs for non-local groups (for example, 239.x.x.x is considered locally-scoped, and PIM-SM components do not send (*,*,RP) Joins to RPs supporting only that portion of the address space). The period is set according to standard PIM-SM rules for periodic Join/Prune messages.

To properly instantiate Rule 1, whenever PIM creates an (S,G) route entry for an externally-reached source, and the next hop towards S is reached via an interface owned by another component, the iif should always point towards S and not towards the RP for G. In addition, the Border-bit is set in all PIM Register messages for this entry.

Finally, the PIM-SM component acts as a DR for externally-reached receivers in terms of being able to switch to the shortest-path tree for internally-reached sources.

6.1 Generating Alerts

A (S,G) Prune alert is sent to the component owning the iif for a forwarding cache entry whenever the last oif is removed from the entry and the iif is owned by another component. In PIM-SM, this may happen when:

- A PIM (S,G) Join/Prune message with S in the prune list is received on a point-to-point interface, or
- A PIM (S,G) Assert from a preferred neighbor was received on the interface, or
- A PIM Register-Stop message is received for (S,G), or
- The interface's Oif-Timer for PIM's (S,G) route table entry expires.
- The Entry-Timer for PIM's (S,G) route table entry expires.

When it is known that there are no longer any members of a group G in the PIM-SM region which receive data for externally-reached sources from the local router, and exactly one other component wants data for G, a (*,G) Prune alert is sent to that component. When it is known that there are no longer any members of a group G in the PIM-SM region which receive data for externally-reached sources from the local router, and no other components want data for G, a (*,G) Prune alert is sent to *all* other components. In PIM-SM, these may happen when:

- A PIM (*,G) Join/Prune message with G in the prune list is received on a point-to-point interface, or
- A PIM (*,G) Assert from a preferred neighbor was received on the interface, or

- IGMP notifies PIM-SM that directly-connected members no longer exist on the interface.
- The Entry-Timer for PIM's (*,G) route table entry expires.

A (S,G) Join alert is sent to the component owning the iif for a forwarding cache entry whenever the first logical oif is added to an entry and the iif is owned by another component. In PIM-SM, this may happen when any of the following occur:

- A PIM (S,G) Join/Prune message is received on the interface, or
- The Register-Suppression-Timer for (S,G) expires, or
- The Entry-Timer for a (S,G) negative-cache state route table entry expires.

When it is first known that there are members of a group G in the PIM-SM region, and exactly one other component wants data for G, a (*,G) Join alert is sent to that component. When it is first known that there are members of a group G in the PIM-SM region, and no other components want data for G, a (*,G) Join alert is sent to *all* other components. In PIM-SM, these may happen when any of the following occur:

- A PIM (*,G) Join/Prune message is received on the interface, or
- A PIM (*,*,RP) Join/Prune message is received on the interface, or
- (*,G) negative cache state expires, or
- IGMP notifies PIM that directly-connected group members now exist on the interface.

6.2 Processing Alerts

When a PIM-SM component receives a (S,G) Creation alert, a longest match ((S,G), then (*,G), then (*,*,RP)) search is done in the routing table. All outgoing interfaces of that entry are then added to the (S,G) forwarding cache entry, except that the forwarding cache entry's iif is never added to the oiflist. Unless the PIM-SM component owns the iif, the oiflist is also modified to support sending PIM Registers with the Border-bit set to the corresponding RP.

When a PIM-SM component receives a (S,G) Prune alert, and the forwarding cache entry's oiflist is empty, it sends a (S,G) Join/Prune message with S in the prune list to the upstream neighbor according to normal PIM-SM behavior.

When a PIM-SM component receives a (*,G) Prune alert, and PIM's (*,G) route entry's oiflist is empty, it sends a (*,G) Join/Prune message with G in the prune list to the upstream neighbor towards the RP for G, according to normal PIM-SM behavior.

When a PIM-SM component receives a (S,G) Join alert, it sends a (S,G) Join/Prune message to the next-hop neighbor towards S, and resets the (S,G) Entry-timer, according to normal PIM-SM behavior.

When a PIM-SM component receives a (*,G) Join alert, and PIM's (*,G) route entry's oiflist is empty, it sends a (*,G) Join/Prune message to the next-hop neighbor towards the RP for G, and resets the (*,G) Entry-timer, according to normal PIM-SM behavior.

7 CBT

In this section we describe how the rules in section 2 apply to CBT. We assume that the reader is familiar with normal CBT behavior as specified in [5]. We note that, like MOSPF, CBT allows joining and pruning entire groups, but not individual sources within groups.

Interoperability between a single CBT stub region and a DVMRP backbone is outlined in [8]. Briefly, CBT MBR components are statically configured such that, whenever an external route exists between two or more MBRs, one is designated as the primary, and the others act as non-forwarding (to prevent duplicate packets) backups. Thus, a CBT region must not serve as transit between two regions if another route between them exists.

We now describe how a CBT implementation may extend this to interoperate with all other multicast routing protocols. A CBT component acts as a wildcard receiver for internally-reached sources if and only if any other component is a wildcard receiver for externally-reached sources. It does this by sending JOIN-REQUESTs for all non-local group ranges to all known cores, as described in [8].

Unless some form of Regional-Membership-Reports (RMRs) (described in [1]) are added to CBT in the future, all CBT components act as wildcard receivers for externally-reached sources. If RMRs are available for the region, then a CBT component acts as a wildcard receiver for externally-reached sources only if internally-reached regions exist which do not support some form of RMRs.

7.1 Generating Alerts

When the last oif is removed from the core tree for G, and exactly one other component wants data for G, a (*,G) Prune alert is sent to that component. When the last oif is removed from the core tree for G, and no other components want data for G, a (*,G) Prune alert is sent to *all* other components. Since CBT *always* sends all data to the core, the only time these can occur after the entry is created is when the MBR is the core. In this case, the last oif is removed from the entry when:

- A QUIT-REQUEST is received on the logical interface, and there are no directly-connected members present on the interface, or
- IGMP notifies CBT that there are no longer directly-connected members present on the interface, and the interface is not a CBT child interface for group G.

Whenever the first CBT outgoing interface is added to an existing core tree, and exactly one other component wants to receive data for G, a (*,G) Join alert is sent to that component. Whenever the first CBT outgoing interface is added to an existing core tree, and no other components want to receive data for G, a (*,G) Join alert is sent to *all* other components. Since CBT *always* sends all data to the core, the only time these can occur after the entry is created is when the MBR is the core. In this case, the first logical oif is added to an entry when:

- A JOIN-REQUEST for G is received on the interface, or
- IGMP notifies CBT that directly-connected group members now exist on the interface.

7.2 Processing Alerts

When a CBT component receives a (S,G) Creation alert, and the router is functioning as the designated BR, any CBT interfaces which are on the tree for G are added to the forwarding cache entry's oif list (according to normal CBT behavior).

When a CBT component receives a (S,G) Prune alert, the alert is ignored, since CBT cannot prune specific sources. Thus, it will continue to receive packets from S since it must receive packets from other sources in group G.

When a CBT component receives a (*,G) Prune alert, and the router is not the primary core for G, and the only CBT on-tree interface is the interface towards the core, it sends a QUIT-REQUEST to the next-hop neighbor towards the core, according to normal CBT behavior.

When a CBT component receives either a (S,G) Join alert or a (*,G) Join alert, and the router is not the primary core for G, and the router is not already on the core-tree for G, it sends a CBT (*,G) JOIN-REQUEST to the next-hop neighbor towards the core, according to normal CBT behavior.

References

[1] Ajit S. Thyagarajan and Stephen E. Deering. Hierarchical distance-vector multicast routing for the MBone. In *Proceedings of the ACM SIGCOMM*, pages 60–66, October 1995.

[2] T. Pusateri. Distance vector multicast routing protocol. *Internet Draft*, September 1996. Available from ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-dvmrp-v3-03.ps.

[3] J. Moy. Multicast extensions to OSPF. *RFC1584*, July 1993. Available from ftp://ds.internic.net/rfc/rfc1585.txt.

[4] Estrin, Farinacci, Helmy, Thaler, Deering, Handley, Jacobson, Liu, Sharma, and Wei. Protocol independent multicast-sparse mode (PIM-SM): Specification. *Internet Draft*, September 1996. Available from ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-pim-sm-spec-07.ps.

[5] A. J. Ballardie, S. Reeve, and N. Jain. Core based trees (CBT) multicast: Protocol specification. *Internet Draft*, April 1996. Available from ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-cbt-spec-06.txt.

[6] Estrin, Farinacci, Helmy, Jacobson, and Wei. Protocol independent multicast (PIM), dense mode protocol specification. *Internet Draft*, September 1996. Available from ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-pim-dm-spec-04.ps.

[7] W. Fenner. Internet group management protocol, version 2. *Internet Draft*, May 1996. Available from ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-igmp-v2-03.txt.

[8] A. J. Ballardie. Core based tree (CBT) multicast interoperability - stage 1. *Internet Draft*, April 1996. Available from ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-cbt-interop1-00.txt.

8 Security Considerations

Security considerations are not discussed in this document. All operations described herein are internal to multicast border routers.

9 Address of Author

Dave Thaler
EECS Department
University of Michigan
Ann Arbor, MI 48109
Phone: (313) 763-5243
Email: thalerd@eecs.umich.edu