Network Working Group
Request for Comments: 2715
Category: Informational

D. Thaler
Microsoft

October 1999

Interoperability Rules for Multicast Routing Protocols

Status of this Memo

Copyright Notice

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, as well as for IGMP-only links. Future versions of these protocols, and any other multicast routing protocols, may describe their interoperability procedure by stating how the rules described herein apply to them.

1. Introduction

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

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

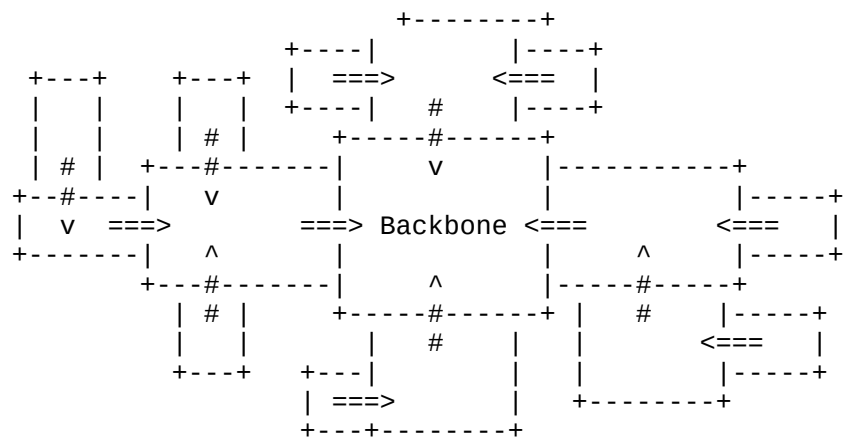

Figure 1: Tree Topology of Domains

- An arbitrary topology, in which a higher level (inter-domain) routing protocol, such as HDVMRP [1] or BGMP [9], is used to calculate paths among domains. Each pair of adjacent domains is connected by one or more 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) 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:

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

- The router is configured to forward packets between two or more independent domains. The router has one or more active interfaces in each domain, and one component per domain. The router also has an inter-component "alert dispatcher", which we cover in Section 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.

- 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 multicast 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.

1.1. Specification Language

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.

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. This component, which we call the "iif owner" is determined by the dispatcher (see Section 3). The incoming component may select ANY interface it owns as the iif according to its own rules.

When a routing change occurs which causes 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.

Furthermore, some multicast routing protocols (e.g. PIM) also require the ability to react to packets received on the "wrong" interface. To support these protocols, an MBR must allow a component to place any of its interfaces in "WrongIf Alert Mode". If a packet arrives on such an interface, and is not accepted according to Rule 2, then the component owning the interface MUST be alerted [(S,G) WrongIf alert]. Typically, WrongIf alerts must be rate-limited.

Rule 3: Whenever a new (S,G) forwarding cache entry is to be created (e.g., 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, or when such an (S,G) forwarding cache entry is created with an empty oif list, 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; this will happen naturally as a result of Rule 5 when it receives a (*,*) Join alert.

For example, if the backbone does not keep global membership information, all MBR components in the backbone in a tree topology of domains, 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 BGMP, 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:

- o 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 sent to all components and all components have updated the oiflist. OR,

- o 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 domains 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. Alert Dispatchers

We assume that each MBR has an "alert dispatcher". The dispatcher is responsible for selecting, for each (S,G) entry in the shared forwarding cache, the component owning the iif. It is also responsible for selecting to which component(s) a given alert should be sent.

3.1. The "Interop" Dispatcher

We describe here rules that may be used in the absence of any inter-domain multicast routing protocol, to enable interoperability in a tree topology of domains. If an inter-domain multicast routing protocol is in use, another dispatcher should be used instead. The Interop dispatcher does not own any interfaces.

To select the iif of an (S,G) entry, the iif owner is the component owning the next-hop interface towards S in the multicast RIB.

The "iif owner" of (*,G) and (*,*) entries is the Interop dispatcher itself. This allows the Interop dispatcher to receive relevant alerts without owning any interfaces.

3.1.1. Processing Alerts

If the Interop dispatcher receives an (S,G) Creation alert, it adds no interfaces to the entry's oif list, since it owns none.

When the Interop dispatcher receives a (*,G) Prune alert, the following actions are taken, depending on the number of components N which want to receive data for G. If N has just changed from 2 to 1, a (*,G) Prune alert is sent to the remaining component. If N has just changed from 1 to 0, a (*,G) Prune alert is sent to ALL components other than the 1.

When the Interop dispatcher receives a (*,G) Join alert, the following actions are taken, depending on the number of components N which want to receive data for G. If N has just changed from 0 to 1, a (*,G) Join alert is sent to ALL components other than the 1. If N has just changed from 1 to 2, a (*,G) Join alert is sent to the original (1) component.

3.2. "BGMP" Dispatcher

This dispatcher can be used with an inter-domain multicast routing protocol (such as BGMP) which allows global (S,G) and (*,G) trees.

The iif owner of an (S,G) entry is the component owning the next-hop interface towards S in the multicast RIB.

The iif owner of a (*,G) entry is the component owning the next-hop interface towards G in the multicast RIB.

3.2.1. Processing Alerts

This dispatcher simply forwards all (S,G) and (*,G) alerts to the iif owner of the associated entry.

4. Multicast Routing Protocol Components

In this section, we describe how the rules in section 2 apply to current versions of various protocols. Future versions, and additional protocols, should describe how these rules apply in a separate document.

4.1. 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 Domain-Wide-Reports (DWRs) [10] (synonymous with Regional-Membership-Reports as described in [1]) are added to DVMRP in the future, all DVMRP components also act as wildcard receivers for externally-reached sources. If DWRs are available for the domain, then a DVMRP component acts as a wildcard receiver for externally-reached sources only if internally-reached domains exist which do not support some form of DWRs.

One simple heuristic to approximate DWRs is to assume that if there are any internally-reached members, then at least one of them is a sender. With this heuristic, the presense of any (S,G) state for internally-reached sources can be used instead. Sending a data packet to a group is then equivalent to sending a DWR for the group.

4.1.1. Generating Alerts

A (*,*) Join alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when the first component becomes a wildcard receiver for external sources. This may occur when a DVMRP component starts up which does not support some form of DWRs.

A (*,*) Prune alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when all components are no longer wildcard receivers for external sources. This may occur when a DVMRP component which does not support some form of DWRs shuts down.

An (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.

An (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:

- o The oif's prune timer expires, or
- o A DVMRP (S,G) Graft message is received on the logical interface, or
- o IGMP [7] notifies DVMRP that directly-connected members of G now exist on the interface.

When it is known, for a group G, that there are no longer any members in the DVMRP domain which receive data for externally-reached sources from the local router, a (*,G) Prune alert is sent to the "iif owner" for (*,G) according to the dispatcher. In DVMRP, this may happen when:

- o The DWR for G times out, or
- o The members-are-senders approximation is being used and the last (S,G) entry for G is timed out.

When it is first known that there are members of a group G in the DVMRP domain, a (*,G) Join alert is sent to the "iif owner" of (*,G). In DVMRP, this may happen when either of the following occurs:

- o A DWR is received for G, or
- o The members-are-senders approximation is being used and a data packet for G is received on one of the component's interfaces.

4.1.2. Processing Alerts

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

- o the iif,
- o interfaces without local members of the entry's group, and for which DVMRP (S,G) Prune messages have been received from all downstream dependent neighbors.
- o interfaces for which the router is not the designated forwarder for S,
- o and interfaces with scoped boundaries covering the group.

When a DVMRP component receives an (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) or (*,*) Prune alert, it is treated as if an (S,G) Prune alert were received for every existing DVMRP (S,G) entry covered. In addition, if DWRs are being used, a DWR Leave message is sent within its domain.

When a DVMRP component receives an (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) or (*,*) Join alert, it is treated as if an (S,G) Join alert were received for every existing DVMRP (S,G) entry covered. In addition, if DWRs are being used, the component sends a DWR Join message within its domain.

4.2. 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 domains exist which do not support some form of Domain-Wide-Reports (DWRs) [10]. Since MOSPF floods membership information throughout the domain, MOSPF itself is considered to support a form of DWRs natively.

4.2.1. Generating Alerts

A (*,*) Join alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when the first component becomes a wildcard receiver for external sources. This may occur when an MOSPF component starts up and decides to act in this role.

A (*,*) Prune alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when all components are no longer wildcard receivers for external sources. This may occur when an MOSPF component which was acting in this role shuts down.

When it is known that there are no longer any members of a group G in the MOSPF domain, a (*,G) Prune alert is sent to the "iif owner" for (*,G) according to the dispatcher. In MOSPF, this may happen when either:

- o IGMP notifies MOSPF that there are no longer any directly-connected group members on an interface, or
- o 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 domain, a (*,G) Join alert is sent to the "iif owner" of (*,G), according to the dispatcher. In MOSPF, this may happen when any of the following occur:

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

4.2.2. Processing Alerts

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

When an MOSPF component receives an (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 domain according to normal MOSPF behavior.

When an MOSPF component receives either an (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 wildcard multicast receiver), it originates a group-membership-LSA in the MOSPF domain according to normal MOSPF behavior.

When an MOSPF component receives a (*,*) Prune alert, it ceases to be a wildcard multicast receiver in its domain.

When an MOSPF component receives a (*,*) Join alert, it becomes a wildcard multicast receiver in its domain.

4.3. 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 Domain-Wide-Reports (DWRs) [10] are added to PIM-DM in the future, all PIM-DM components also act as wildcard receivers for externally-reached sources. If DWRs are available for the domain, then a PIM-DM component acts as a wildcard receiver for externally-reached sources only if internally-reached domains exist which do not support some form of DWRs.

One simple heuristic to approximate DWRs is to assume that if there are any internally-reached members, then at least one of them is a sender. With this heuristic, the presense of any (S,G) state for internally-reached sources can be used instead. Sending a data packet to a group is then equivalent to sending a DWR for the group.

4.3.1. Generating Alerts

A (*,*) Join alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when the first component becomes a wildcard receiver for external sources. This may occur when a PIM-DM component starts up which does not support some form of DWRs.

A (*,*) Prune alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when all components are no longer wildcard receivers for external sources. This may occur when a PIM-DM component which does not support some form of DWRs shuts down.

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.

When it is known that there are no longer any members of a group G in the PIM-DM domain which receive data for externally-reached sources from the local router, a (*,G) Prune alert is sent to the "iif owner" for (*,G) according to the dispatcher. In PIM-DM, this may happen when:

- o The DWR for G times out.
- o The members-are-senders approximation is being used and PIM-DM's last (S,G) entry for G is timed out.

When it is first known that there are members of a group G in the PIM-DM domain, a (*,G) Join alert is sent to the "iif owner" of (*,G), according to the dispatcher. In PIM-DM, this may happen when either of the following occurs:

- o A DWR is received for G.
- o The members-are-senders approximation is being used and a data packet for G is received on one of the component's interfaces.

4.3.2. Processing Alerts

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

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

When a PIM-DM component receives an (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) or (*,*) Prune alert, it is treated as if an (S,G) Prune alert were received for every matching (S,G) entry.

When a PIM-DM component receives an (S,G) Join alert, and an (S,G) 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) or (*,*) Join alert, then for each matching (S,G) entry in the PIM-DM routing table for which 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. In addition, if DWR's are being used, the component sends a DWR Join message within its domain.

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

Unless some form of Domain-Wide-Reports (DWRs) [10] are added to PIM-SM in the future, all PIM-SM components act as wildcard receivers for externally-reached sources. If DWRs are available for the domain, then a PIM-SM component acts as a wildcard receiver for externally-reached sources only if internally-reached domains exist which do not support some form of DWRs.

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 a PIM (S,G) 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.

4.4.1. Generating Alerts

A (*,*) Join alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when the first component becomes a wildcard receiver for external sources. This may occur when a PIM-SM component starts up and decides to act in this role.

A (*,*) Prune alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when all components are no longer wildcard receivers for external sources. This may occur when a PIM-SM component which was acting in this role shuts down.

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 domain which receive data for externally-reached sources from the local router, a (*,G) Prune alert is sent to the "iif owner" for (*,G) according to the dispatcher. In PIM-SM, this 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 an (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 domain, a (*,G) Join alert is sent to the "iif owner" of (*,G), according to the dispatcher. In PIM-SM, this may happen when any of the following occur:

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

4.4.2. Processing Alerts

When a PIM-SM component receives an (S,G) Creation alert, it does a longest match search ((S,G), then (*,G), then (*,*,RP)) in its multicast routing table. All outgoing interfaces of that entry are then added to the forwarding cache entry. 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 an (S,G) Prune alert, and the forwarding cache entry's oiflist is empty, then for each PIM (S,G) state entry covered, it sends an (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, 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 an (S,G) Join alert, it sends an (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, then 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.

When a PIM-SM component receives a (*,*) Join alert, then it sends (*,*,RP) Join/Prune messages towards each RP.

When a PIM-SM component receives a (*,*) Prune alert, then it sends a (*,*,RP) Prune towards each RP.

4.5. CBTv2

In this section we describe how the rules in section 2 apply to CBTv2. We assume that the reader is familiar with normal CBTv2 behavior as specified in [5]. We note that, like MOSPF, CBTv2 allows joining and pruning entire groups, but not individual sources within groups.

Interoperability between a single CBTv2 stub domain and a DVMRP backbone is outlined in [8]. Briefly, CBTv2 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 CBTv2 domain must not serve as transit between two domains if another route between them exists.

We now describe how a CBTv2 implementation may extend this to interoperate with all other multicast routing protocols. A CBTv2 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 Domain-Wide-Reports (DWRs) [10] are added to CBTv2 in the future, all CBTv2 components act as wildcard receivers for externally-reached sources. If DWRs are available for the domain, then a CBTv2 component acts as a wildcard receiver for externally-reached sources only if internally-reached domains exist which do not support some form of DWRs.

4.5.1. Generating Alerts

A (*,*) Join alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when the first component becomes a wildcard receiver for external sources. This may occur when a PIM-SM component starts up and decides to act in this role.

A (*,*) Prune alert is sent to the iif owner of the (*,*) entry (e.g., the Interop dispatcher) when all components are no longer wildcard receivers for external sources. This may occur when a PIM-SM component which was acting in this role shuts down.

When the last oif is removed from the core tree for G, a (*,G) Prune alert is sent to the "iif owner" for (*,G) according to the dispatcher. Since CBTv2 always sends all data to the core, the only time this 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:

- o A QUIT-REQUEST is received on the logical interface, and there are no directly-connected members present on the interface, or
- o 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.

When the first CBT outgoing interface is added to an existing core tree, a (*,G) Join alert is sent to the "iif owner" of (*,G) according to the dispatcher. Since CBTv2 always sends all data to the core, the only time these can occur, other than when the entry is created, is when the MBR is the core. In this case, the first logical oif is added to an entry when:

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

4.5.2. Processing Alerts

When a CBTv2 component receives an (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 CBTv2 behavior).

When a CBTv2 component receives an (S,G) Prune alert, the alert is ignored, since CBTv2 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 CBTv2 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 CBTv2 behavior.

When a CBTv2 component receives either an (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 CBTv2 behavior.

4.6. IGMP-only links

In this section we describe how the rules in section 2 apply to a link which is not within any routing domain, and hence no routing protocol messages are exchanged and the interface is not owned by any multicast routing protocol component. We assume that the reader is familiar with normal IGMP behavior as specified in [7]. We note that IGMPv2 allows joining and pruning entire groups, but not individual sources within groups.

An IGMP-only "component" may only own a single interface; hence an IGMP-only domain only consists of a single link. Since an IGMP-only component can only act as a wildcard receiver for internally-reached sources if all internally-reached sources are directly-connected, then either the IGMP-only domain (link) must be a stub domain, or else there must be no other components which are wildcard receivers for externally-reached sources.

4.6.1. Generating Alerts

When it is known that there are no longer any directly-connected members of a group G on the IGMP-only interface, a (*,G) Prune alert is sent to the "iif owner" for (*,G) according to the dispatcher. In IGMP, this may happen when:

- The group membership times out.

When it is first known that there are directly-connected members of a group G on the interface, a (*,G) Join alert is sent to the "iif owner" of (*,G), according to the dispatcher. In IGMP, this may happen when any of the following occur:

- A Membership Report is received for G.

4.6.2. Processing Alerts

When an IGMP-only component receives an (S,G) Creation alert, and there are directly-connected members of G present on its interface, it adds the interface to the entry’s oif list.

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

When an IGMP-only component receives a (*,G) Prune alert, the router leaves the group G, sending an IGMP Leave message if it was the last reporter, according to normal IGMPv2 behavior.

When an IGMP-only component receives a (*,*) Prune alert, it leaves promiscuous multicast mode.

When an IGMP-only component receives either an (S,G) Join alert or a (*,G) Join alert, and the component was not previously a member of G on the IGMP-only interface (and the component is not a wildcard receiver for internally reached sources), it joins the group on the interface, causing it to send an unsolicited Membership Report according to normal IGMP behavior.

When an IGMP-only component receives a (*,*) Join alert, it enters promiscuous multicast mode.

5. Security Considerations

All operations described herein are internal to multicast border routers. The rules described herein do not change the security issues underlying individual multicast routing protcols. Allowing different protocols to interact, however, means that security weaknesses of any particular protocol may also apply to the other protocols as a result.

6. 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] Pusateri, T., "Distance Vector Multicast Routing Protocol", Work in Progress.

[3] Moy, J., "Multicast Extensions to OSPF", RFC 1584, March 1994.

[4] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P. and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, June 1998.

[5] Ballardie, A., "Core Based Trees (CBT version 2) Multicast Routing", RFC 2189, September 1997.

[6] Estrin, Farinacci, Helmy, Jacobson, and Wei, "Protocol Independent Multicast (PIM), Dense Mode Protocol Specification", Work in Progress.

[7] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[8] Ballardie, A., "Core Based Tree (CBT) Multicast Border Router Specification", Work in Progress.

[9] Thaler, D., Estrin, D. and D. Meyer, "Border Gateway Multicast Protocol (BGMP): Protocol Specification", Work in Progress.

[10] Fenner, W., "Domain Wide Multicast Group Membership Reports", Work in Progress.

7. Author's Address

Dave Thaler
Microsoft
One Microsoft Way
Redmond, WA 98052

Phone: (425) 703-8835
EMail: dthaler@microsoft.com

8. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.