INTERNET-DRAFT
Obsoletes: RFC 1075

T. Pusateri
NetEdge Systems

February 1996
Expires: August 1996

Distance Vector Multicast Routing Protocol

Status of this Memo

1. Introduction

DVMRP is a TCP/IP routing protocol that provides an efficient mechanism for connectionless datagram delivery to a group of hosts across an internetwork. It is a distributed protocol that dynamically generates multicast delivery trees for IP Multicast datagrams using a technique called Reverse Path Multicasting (RPM) [1]. This document describes Version 3 of the protocol replacing Version 1 specified in RFC 1075 [2].

1.1 Reverse Path Multicasting

Datagrams follow multicast delivery trees from a source to all members of a multicast group [3], replicating the packet only at necessary branches in the delivery tree. The trees are calculated and updated dynamically to track the membership of individual groups. When a datagram arrives on an interface, the reverse path to the source of the datagram is determined by examining a unicast routing table of known source networks. If the datagram arrives on an interface that would be used to transmit unicast datagrams back to the source, then it is forwarded to the appropriate list of downstream interfaces. Otherwise, it is not on the optimal delivery tree and should be discarded. In this way duplicate packets can be filtered when loops exist in the network topology. The source specific delivery trees are automatically pruned back as group membership changes or leaf routers determine that no group members are present. This keeps the delivery trees to the minimum branches necessary to reach all of the group members. New sections of the tree can also be added dynamically as new members join the multicast group by grafting the new sections onto the delivery trees.

1.2 IP-IP Tunnels

Because not all IP routers support native multicast routing, DVMRP includes direct support for tunneling IP Multicast datagrams through routers. The IP Multicast datagrams are encapsulated in unicast IP packets and addressed to the routers that do support native multicast routing. DVMRP treats tunnel interfaces in an identical manner to physical network interfaces.

1.3 Document Overview

Section 2 provides an overview of the protocol and the different message types exchanged by DVMRP routers. Those who wish to gain a general understanding of the protocol but are not interested in the more precise details may wish to only read this section. Section 3 explains the detailed operation of the protocol to accommodate developers needing to provide interoperable implementations. The interaction with the Internet Group Management Protocol (IGMP) [4] is discussed in Appendix A. A summary of the DVMRP constants and configurable parameters are included in Appendix B. A section on DVMRP support for tracing and troubleshooting procedures is the topic of Appendix C. Finally, a short DVMRP version history is provided in Appendix D to assist with backward compatibility issues.

Table of Contents

2. Protocol Overview

2.1 Neighbor Discovery

Neighbor DVMRP routers can be discovered dynamically by sending Neighbor Probe Messages on all of the local multicast capable network interfaces. These messages are sent periodically to the All-DVMRP-Routers IP Multicast group address. This address falls into the range of IP Multicast addresses that are to remain on the locally attached IP network and therefore are not forwarded by multicast routers. Beginning with Version 3 of DVMRP outlined in this document, each Neighbor Probe message should contain the list of Neighbor DVMRP routers for which Neighbor Probe messages have been received. In this way, Neighbor DVMRP routers can ensure that they are seen by each other. Care must be taken to interoperate with older implementations of DVMRP that do not include this list of neighbors. It can be assumed that older implementations of DVMRP will safely ignore this list of neighbors in the Probe message. Therefore, it is not necessary to send both old and new types of Neighbor Probes. In addition to the list of known neighbors, the Probe message should contain a set of flags describing the attributes of the DVMRP router. There are currently four attributes defined.

1. PRUNE - This router supports Pruning of multicast delivery trees
2. GENID - This router is including a Generation Identifier in the probe message that can be used to detect when a neighbor wants to re-synchronize its unicast routing database.
3. LEAF - This router considers itself a leaf router. (i.e. it is not aware of any downstream multicast routers.)
4. MTRACE - This router is capable of responding to a multicast trace route request.

As mentioned before, DVMRP versions prior to this specification may not include this set of flags and this information will have to be determined by the DVMRP Major and Minor version numbers.

2.2 Source Location

When an IP Multicast datagram is received by a router running DVMRP, it first looks up the interface of the unicast route back to the source of the datagram. This interface is called the upstream interface. If the datagram arrived on the correct upstream interface, then it is a candidate for forwarding to one or more downstream interfaces. If the datagram did not arrive on the anticipated upstream interface, it is discarded. This check is known as a reverse path forwarding check and must be performed by all DVMRP routers.

In order to ensure that all DVMRP routers have a consistent view of the unicast path back to a source, a unicast routing table is propagated to all DVMRP routers as an integral part of the protocol. Each router advertises the network number and mask of the interfaces it is directly connected to as well as relaying the routes received from neighbor routers. DVMRP allows for an interface metric to be configured on all physical and tunnel interfaces. When a route is received, the metric of the upstream interface over which the datagram was received must be added to the metric of the route being propagated. This adjusted metric should be computed before the route is compared to the metric of the current next hop gateway. As is customary with distance vector routing protocols, split horizon should be applied to the route propagation policy in order to prevent advertising a route to a destination over the interface from which it was received.

Although there is certainly additional overhead associated with propagating a separate unicast routing table, it does provide two nice features. First, since all DVMRP routers are using the same unicast routing protocol, there are no inconsistencies between routers when determining the upstream interface (aside from normal convergence issues related to distance vector routing protocols). By placing the burden of synchronization on the protocol as opposed to the network manager, DVMRP reduces the risk of creating routing loops or blackholes due to disagreement between neighbor routers on the upstream interface.

Second, by propagating its own unicast routing table, DVMRP makes it convenient to have separate paths for unicast vs. multicast datagrams. Although, ideally, many network managers would prefer to keep their unicast and multicast traffic aligned, tunneled multicast topologies may prevent this causing the unicast and multicast paths to diverge. Additionally, service providers may prefer to keep the unicast and multicast traffic separate for routing policy reasons as they experiment with IP multicast routing and begin to offer it as a service. For these benefits, DVMRP has chosen to accept the additional overhead of propagating unicast routes.

2.3 Dependent Downstream Routers

In addition to providing a consistent view of source networks, the exchange of unicast routes in DVMRP provides one other important feature. DVMRP uses the unicast route exchange as a mechanism for upstream routers to determine if any downstream routers depend on them for forwarding from particular source networks. DVMRP accomplishes this by using a well known technique called “Poison Reverse”. If a downstream router selects a particular upstream router as the best next hop to a particular source network, then it signifies this to the upstream router by echoing back the route to the upstream router with a metric equal to the original metric plus infinity. When the upstream router receives the report and sees a metric that lies between infinity and two times infinity, it can then add the downstream router from which it received the report to a list of dependent routers for this source.

This list of dependent routers per source network built by the “Poison Reverse” technique will provide the foundation necessary to determine when it is appropriate to prune back the source specific IP multicast trees.

2.4 Building Multicast Trees

As previously mentioned, when an IP multicast datagram arrives, the upstream interface is determined by looking up the interface that would be used if a datagram was being sent back to the source of the datagram. If the upstream interface is correct, then a DVMRP router will forward the datagram to a list of downstream interfaces.

2.4.1 Adding Leaf Networks

Initially, the DVMRP router must consider all of the remaining IP multicast capable interfaces (including tunnels) on the router. If the downstream interface under consideration is a leaf network (has no other IP multicast routers on it), then the IGMP local group database must be consulted. DVMRP routers can easily determine if a directly attached network is a leaf network by keeping a list of all routers from which DVMRP Router Probe messages have been received on the interface. Obviously, it is necessary to refresh this list and age out entries received from routers that are no longer being refreshed. The IGMP local group database is maintained by an elected IP multicast router on each physical, multicast capable network. The details of the election procedure are discussed in section 3. If the destination group address is listed in the local group database, then the interface should be included in the list of downstream interfaces. If there are no group members on the interface, then the interface can be pruned from the candidate list.

2.4.2 Adding Non-Leaf Networks

Initially, all non-leaf networks should be included in the downstream interface list when a forwarding cache entry is first being created. This allows all downstream routers to be aware of traffic destined for a particular (source, group) pair. The downstream routers will then have the option to prune and graft this (source, group) pair to and from the multicast delivery tree as requirements change from their downstream routers and local group members.

2.5 Pruning Multicast Trees

As mentioned above, routers at the edges with leaf networks will prune their leaf interfaces that have no group members associated with an IP multicast datagram. If a router prunes all of its downstream interfaces, it can notify the upstream router that it no longer wants traffic destined for a particular (source, group) pair. This is accomplished by sending a DVMRP Prune message upstream to the router it expects to forward datagrams from a particular source. Recall that a downstream router will inform an upstream router that it depends on the upstream router to receive datagrams from particular source networks by using the “Poison Reverse” technique during the exchange of unicast routes. This method allows the upstream router to build a list of downstream routers on each interface that are dependent upon it for datagrams from a particular source network. If the upstream router receives prune messages from each one of the dependent downstream routers on an interface, then the upstream router can in turn prune this interface from its downstream interface list. If the upstream router is able to prune all of its downstream interfaces in this way, it can then send a DVMRP Prune message to its upstream router. This continues until the minimum tree necessary to reach all of the receivers remains. Since IP multicast routers may be restarted at any time and lose state information about existing prunes, it is necessary to limit the life of a prune and periodically resume the flooding procedure. This will reinitiate the prune mechanism and the cycle will continue. When a router decides to prune one of its downstream interfaces, it will set a timer to indicate the lifetime of the prune. If all of its downstream interfaces become pruned off the multicast delivery tree and a DVMRP Prune message is sent upstream, the lifetime of the prune will be equal to the minimum of the remaining lifetimes of the pruned interfaces.

2.6 Grafting Multicast Trees

Once a tree branch has been pruned from a multicast delivery tree, packets from the pruned (source, group) pair will no longer be forwarded. There are two different ways for packets from the (source, group) pair to be forwarded again. First, since IP multicast supports dynamic group membership, new hosts may join the multicast group. In this case, DVMRP routers will need a mechanism to undo the prunes that are in place from the host back to the first branch that was pruned from the multicast tree.

This is accomplished with a DVMRP Graft message. A router will send a Graft message to its upstream neighbor if a group join occurs for a group that currently has pruned sources. Separate Graft messages must be sent to the appropriate upstream neighbor for each source that has been pruned. Since there would be no way to tell if a Graft message sent upstream was lost or the source simply quit sending traffic, it is necessary to acknowledge each Graft message with a DVMRP Graft Ack message. If an acknowledgment is not received within a Graft Time-out period, the Graft message should be retransmitted. Duplicate Graft Ack messages should simply be ignored. Second, if the prune interval expires, the negative cache entries are removed and the packets will automatically be forwarded again. This is a necessary feature since routers may be restarted and lose all prune state information or new routers may appear. Since these routers will not have prune state associated with the (source, group) pair, they will not realize that a DVMRP Graft message is necessary if a new host joins the group. Therefore, by periodically timing out the prunes and re-flooding the traffic, any new or restarted routers can have their prune state periodically refreshed.

2.7 Suppressing Duplicate Packets

3. Detailed Protocol Operation

This section contains a detailed description of DVMRP. It covers sending and receiving of DVMRP messages as well as the generation and maintenance of IP Multicast forwarding cache entries.

3.1 Protocol Header

DVMRP packets are encapsulated in IP datagrams, with an IP protocol number of 2 (IGMP) as specified in the Assigned Numbers RFC. All DVMRP packets use a common protocol header that specifies the IGMP Packet Type as hexadecimal 0x13 (DVMRP). A diagram of the common protocol header follows:

Table 1 - Common Protocol Header

0	8	16 31
Type (0x13)	Code	Checksum

The value of the Code field determines the DVMRP packet type. Currently, there are codes allocated for DVMRP protocol message types as well as protocol analysis and troubleshooting packets. The protocol message Codes are:

Table 2 - Standard Protocol Packet Types

Code	Packet Type	Description
1	DVMRP Probe	for neighbor discovery
2	DVMRP Report	for unicast route exchange
7	DVMRP Prune	for pruning multicast delivery trees
8	DVMRP Graft	for grafting multicast delivery trees
9	DVMRP Graft Ack	for acknowledging graft messages

There are additional codes used for protocol analysis and troubleshooting. These codes are discussed in Appendix B. The Checksum is the 16-bit one’s complement of the one’s complement sum of the DVMRP message. The checksum of the DVMRP message should be calculated with the checksum field set to zero.

3.2 DVMRP Probe Messages

When a DVMRP router is configured to run on an physical interface, it sends local IP Multicast discovery packets to inform other DVMRP routers that it is operational. These discovery packets are called DVMRP Probes and they serve three purposes.

1. Probes provide a mechanism for DVMRP routers to locate each other. The existence of other routers is used for network leaf detection and the list of addresses of the other routers are used in designated router election. In addition, this list of DVMRP neighbors provides a foundation for neighbor prune lists.
2. They provide a way for DVMRP routers to determine the capabilities of each other. This may be deduced from the major and minor version numbers in the Probe packet or directly from the capability flags. Examples of these capabilities are whether Generation IDs are used, if the neighbor supports pruning, if the neighbor is a leaf router, and whether the neighbor supports the multicast trace route function.
3. Probes provide a keep-alive function in order to quickly detect neighbor loss. DVMRP probes sent on each multicast capable interface configured for DVMRP SHOULD have an interval of 10 seconds. The neighbor time-out interval SHOULD be set at 140 seconds. This allows fairly early detection of a lost neighbor yet provides tolerance for busy multicast routers. These values MUST be coordinated between all DVMRP routers on a physical network segment.

3.2.1 Router Capabilities

In the past, there have been many versions of DVMRP in use with a wide range of capabilities. Practical considerations require a current implementation to interoperate with these older implementations that don't formally specify their capabilities. For instance, for major versions less than 3, it can be assumed that the neighbor does not support pruning. The formal capability flags were first introduced in version 3.5 in an attempt to take the guess work out which features are supported by a neighbor. A complete version history is summarized in Appendix A to assist with backward compatibility. A DVMRP router MUST specify its capabilities in a Probe message. The following capabilities are currently defined:

Table 3 - Probe Capability Flags

15 8	7							0
Reserved					M	G	P	L

LEAF - The LEAF bit is set if the router does not have any downstream DVMRP neighbor routers.

PRUNE - router is capable of handling prunes

GENID - router keeps a non-decreasing generation id for synchronization on restart.

MTRACE - router includes multicast trace route support

3.2.2 Generation ID

If a DVMRP router is restarted, it must immediately exchange unicast routing tables with all of its neighbors. If a neighbor doesn't automatically detect that the neighbor has restarted, then it will not send its entire routing table immediately. Instead, it will spread the updates over an entire routing update interval. In order for the neighbor to detect a router that is restarted, a non-decreasing number is placed in the periodic probe message called the generation ID. If a router detects an increase in the generation ID of a neighbor, it should exchange its entire unicast routing table with the neighbor. A time of day clock provides a good source for a non-decreasing 32 bit integer.

3.2.3 Neighbor Addresses

As a DVMRP router sees Probe messages from its DVMRP neighbors, it records the neighbor addresses on each interface and places them in the Probe message sent on the particular interface. This allows the neighbor router to know that its probes have been received by the sending router.

3.2.4 Probe Packet Format

The Probe packet is variable in length depending on the number of neighbor IP addresses included. The current Major Version is 3. To maintain compatibility with previous versions, implementations of Version 3 must include pruning and grafting of multicast trees. Non-pruning implementations are HIGHLY discouraged at this time.

Table 4 - DVMRP Probe Packet Format

0	8	16	24
Capabilities		Minor Version	Major Version
Generation ID			
Neighbor IP Address 1			
Neighbor IP Address 2			
...			
Neighbor IP Address N			

3.2.5 Designated Router Election

Since it is wasteful to have more than a single router sending IGMP Host Membership Queries on a given physical network, DVMRP provides a method to elect a single router on each physical network as the Designated Querier. Based on the list of neighbors from which DVMRP Probe messages were received, a router will pick the router with the lowest IP address as the designated querier. The router must be sure and include itself in this list of candidates.

3.3 Building Forwarding Cache Entries

In order to create optimal multicast delivery trees, IP Multicast was designed to keep separate forwarding cache entries for each (source network, destination group) pair. Because the possible combinations of these is quite large, forwarding cache entries are generated on demand as data arrives at a multicast router. Since the IP forwarding decision is made on a hop by hop basis (as with the unicast case), it is imperative that each multicast router has a consistent view of the reverse path back to the source network. For this reason, DVMRP includes its own unicast routing protocol.

3.3.1 Determining the upstream interface

When a multicast packet arrives, a DVMRP router will use the internal DVMRP unicast routing table to determine which interface leads back to the source. If the packet did not arrive on that interface, it should be discarded without further processing. Each multicast forwarding entry should cache the upstream interface for a particular source host or source network after looking this up in the DVMRP unicast routing table.

3.3.2 Determining the downstream interface list

The downstream interface list is built from the remaining list of multicast capable interfaces. Any interfaces designated as leaf networks and do not have members of the particular multicast group can be automatically pruned from list of downstream interfaces. The remaining interfaces will either have downstream DVMRP routers or directly attached group members. These interfaces may be pruned in the future if it is determined that there are no group members anywhere along the entire tree branch.

3.4 Unicast Route Exchange

It was mentioned earlier that since not all IP routers support IP multicast forwarding, it is necessary to tunnel IP multicast datagrams through these routers. One effect of using these encapsulated tunnels is that IP multicast traffic may not be aligned with IP unicast traffic. This means that a multicast datagram from a particular source can arrive on a different interface than the upstream interface back to the unicast source of the multicast datagram.

Therefore, when determining the reverse path back to a particular source it is not always possible to use the standard unicast routing table. DVMRP's solution to this problem is to create its own routing table of unicast routes for determining upstream routers for each source. This routing information is used exclusively for determining the reverse path back to source of multicast traffic. Tunnels are considered to be distinct interfaces and route reports are sent unicast between tunnel endpoints as though they arrived on the tunnel pseudo interface. The routing information that is propagated by DVMRP contains a list of unicast source networks and an appropriate metric. The metric used is a hop count which is incremented by the cost of the outgoing interface. Traditionally, physical interfaces use a metric of 1 while the metric of a tunnel interface varies with the distance and bandwidth in the path between the two tunnel endpoints. Users are encouraged to configure tunnels with the same metric in each direction in order to prevent routing loops although the protocol does not strictly enforce this.

3.4.1 Route Packing and Ordering

Since DVMRP Route Reports may need to refresh several thousand routes each Report interval, routers MUST attempt to spread the routes reported across the whole route update interval. This reduces the chance of synchronized route reports causing routers to become overwhelmed for a few seconds each report interval. Since the route report interval is 60 seconds, it is suggested that the total number routes being updated be split across multiple Route Reports sent at regular intervals. One implementation splits all unicast routes across 6 Report periods sent at 10 seconds intervals. Due to limitations of older implementations of DVMRP, Route Reports should contain source network/mask pairs sorted first by increasing network mask and then by increasing source network within each possible mask value.

In order to pack more source networks into a route report, source networks are often represented by less than 4 octets. The number of significant bytes in the mask value is used to determine the number of octets used to represent each source network within that particular mask value. For instance if the mask value of 255.255.0.0 is being reported, the source networks would only contain 2 octets each. DVMRP assumes that source networks will never be aggregated into networks whose prefix length is less than 8. Therefore, it does not carry the first octet of the mask in the Route Report since, given this assumption, the first octet will always be 0xFF. This means that the netmask value will always be represented in 3 octets.

Immediately following each source network is an octet containing the metric advertised to reach the source network.

3.4.2 Unicast Route Metrics

For each source network reported, a route metric is also contained in the route report. The metric is the sum of the outgoing interface metrics between the router originating the report and the source network. The Infinity metric is defined to be 32. This limits the breadth across the whole DVMRP network and is necessary to place an upper bound on the convergence time of the protocol.

As seen in the packet format below, Route Reports do not contain a count of the number of routes reported for each netmask. Instead, the high order bit of the metric is used to signify the last route being reported for a particular mask value. If a metric is read with the high order bit of the 8-bit value set, then if the end of the message has not been reached, the next value will be a new netmask to be applied to the subsequent list of routes. This technique is used to prevent wasting space in the Route Report message for a count of unicast source networks for each netmask value contained in the Report.

3.4.3 Unicast Route Dependencies

In order for pruning to work correctly, each DVMRP router needs to know which downstream routers depend on it for receiving datagrams from particular source networks. Initially, when a new datagram arrives from a particular source/group pair, it is flooded to all downstream interfaces that have DVMRP neighbors who have indicated a dependency on the receiving DVMRP router for that particular source. A downstream interface can only be pruned when it has received Prune messages from each of the dependent routers on that interface. Each downstream router uses a method called Poison Reverse to indicate to the upstream router which source networks it expects to receive from the upstream router. The downstream router indicates this by echoing back the source networks it expects to receive from the upstream router with infinity added to the advertised metric. This means that the legal values for the metric now become between `1` and `2*Infinity -1` or `1` and `63`. Values between `1` and `31` indicate reachable unicast source networks. The value of Infinity or `32` indicates the source network is not reachable. Values between `33` and `63` indicate that the downstream router originating the Report is depending upon the upstream router to provide multicast datagrams from the corresponding source network.

3.4.4 Sending Route Reports

Full Route Reports MUST be sent out every Route Report Interval. In addition, flash updates CAN be sent between full route reports. Flash updates can reduce the chances of routing loops and black holes occurring when source networks become unreachable through a particular path. Flash updates need only contain the source networks that have changed. It is not necessary to report all of the source networks from a particular mask value when sending an update.

3.4.5 Receiving Route Reports

3.4.6 Route Report Packet Format

The format of a sample Route Report Packet is shown in below. The packet shown is an example of how the source networks are packed into a Report. The number of octets in each Source Network will vary depending on the mask value. The values below are only an example for clarity and are not intended to represent the format of every Route Report.

Table 5 - Route Report Packet Format

0	8	16	31
Mask Octet 2	Mask Octet 3	Mask Octet 4	Src Net$_{11}$
Src Net$_{11}$ (cont.) ...		Metric	Src Net$_{12}$
Src Net$_{12}$ (cont.) ...		Metric + 0x80	Mask Octet 2
Mask Octet 3	Mask Octet 4	Src Net$_{21}$	
Src Net$_{21}$ (cont.) ...		Metric + 0x80	Mask Octet 2
Mask Octet 3	Mask Octet 4	...	

3.5 Pruning

DVMRP is described as a flood and prune multicast routing protocol since datagrams are initially sent out all dependent downstream interfaces and then pruned back to only the downstream interfaces that are on a reverse shortest path to a receiver. Prunes are data driven and are sent in response to receiving unwanted multicast traffic at the leafs of the multicast tree rooted at a particular source network.

3.5.1 Leaf Networks

Detection of leaf networks is very important to the pruning process. Routers at the end of a source specific multicast delivery tree must detect that there are no further downstream routers. This detection mechanism is covered above in section 3.2 titled **DVMRP Probe Messages.** If there are no group members present for a particular multicast datagram received, the leaf routers will start the pruning process by pruning their downstream interfaces and sending a prune to the upstream router for that source.

3.5.2 Source Networks

It is important to remember that because prunes are based on group membership, a prune sent upstream for a particular source applies to all sources on the source network. It is not possible to prune only one or a subset of hosts on a source network for a particular group. All or none of the sources on a source network sending to a particular group must be pruned.

3.5.3 Receiving a Prune

When a prune is received, the following steps should be taken:

1. Determine if a Probe has been received from this router recently.
2. If not, discard prune since there is no prior state about this neighbor.

3. If so, make sure the neighbor advertises it is capable of pruning.
4. Ensure the packet meets the minimum length requirement for a prune.
5. Extract the source address, group address, and prune time-out values
6. If no state exists for the (source, group) pair, then ignore the prune.
7. Verify that the prune was received from a dependent neighbor for the source network. If not, discard the prune.
8. Determine if a prune is currently active for this (source, group) pair.
9. If so, reset the timer to the new time-out value. Otherwise, create state for the new prune and set a timer for the prune lifetime.
10. Determine if all dependent downstream routers have now sent prunes.
11. If so, a prune should be sent upstream.

3.5.4 Sending a Prune

When sending a prune upstream, the following steps should be taken:

1. Decide if upstream neighbor is capable of receiving prunes.
2. If not, then proceed no further.
3. Stop any pending Grafts awaiting acknowledgments.
4. Determine the prune lifetime. This value should be the minimum of the prune lifetimes remaining from the downstream neighbors and the cache lifetime of the (source, group) pair.
5. Form and transmit the packet to the upstream neighbor for the source.

3.5.5 Prune Packet Format

In addition to the standard IGMP and DVMRP headers, a Prune Packet contains three additional fields: the source host IP address, the destination group IP address, and the Prune Lifetime in seconds.

Table 6 - Prune Packet Format

0 31

Source Address
Group Address
Prune Lifetime

3.6 Grafting

Once a multicast delivery tree has been pruned back, DVMRP Graft messages are necessary to join new receivers onto the multicast tree. Graft messages are sent upstream from the new receivers first-hop router until a point on the multicast tree is reached. Graft messages are re-originated between adjacent DVMRP routers and are not forwarded by DVMRP routers. Therefore, the first-hop router does not know if the Graft message ever reaches the multicast tree. To remedy this, each Graft message is acknowledged hop by hop. This ensures that the Graft message is not lost somewhere along the path between the receiver's first-hop router and the closest point on the multicast delivery tree.

3.6.1 Grafting All Sources

It is important to realize that prunes are source specific and are sent up different trees for each source. Grafts are sent in response to a new Group Member which is not source specific. Therefore, separate Graft messages must be sent to the appropriate upstream routers to counteract each previous source specific prune that was sent.

3.6.2 Sending a Graft

As mentioned above, a Graft message sent to the upstream DVMRP router should be acknowledged hop by hop guaranteeing end-to-end delivery. If a Graft Acknowledgment is not received within a Graft Retransmission Time-out period, the Graft should be resent to the upstream router. The time-out period is fixed at 5 seconds.

In order to send a Graft message, the following steps should be taken:

1. Verify a forwarding cache entry exists for the (source, group) pair and that a prune exists for the cache entry.
2. Verify that the upstream router is capable of receiving prunes (and therefore grafts).
3. Add the graft to the retransmission timer list awaiting an acknowledgment.
4. Formulate and transmit the Graft packet.

3.6.3 Receiving a Graft

The actions taken when a Graft is received depends on the state in the receiving router for the (source, group) pair in the received Graft message. If the receiving router has prune state for the (source, group) pair, then it must acknowledge the received graft and send a subsequent graft to its upstream router.

If the receiving router has some pruned downstream interfaces but has not sent a prune upstream, then the receiving interface can simply be added to the list of downstream interfaces in the forwarding cache. A Graft Acknowledgment must also be sent back to the source of the Graft message.

If the receiving router has no state at all for the (source, group) pair, then datagrams arriving for the (source, group) pair should automatically be flooded when they arrive. A Graft Acknowledgment must be sent to the source of the Graft message.

If a Graft message is received from an unknown neighbor, it should be discarded.

3.6.4 Graft Packet Format

The format of a Graft packet is show below:

Table 7 - Graft Packet Format

0 31

Source Address
Group Address

3.6.5 Sending a Graft Acknowledgment

A Graft Acknowledgment packet is sent to a downstream neighbor in response to receiving a Graft message. Grafts received from unknown neighbors should be discarded but all other correctly formatted Graft messages should be acknowledged. This is true even if no other action is taken in response to receiving the Graft to prevent the source from continually re-transmitting the Graft message.

The Graft Acknowledgment packet is identical to the Graft packet except that the DVMRP code in the common header is set to Graft Ack. This allows the receiver of the Graft Ack message to correctly identify which Graft was acknowledged and stop the appropriate retransmission timer.

3.6.6 Receiving a Graft Acknowledgment

When a Graft Acknowledgment is received, the (source, group) pair in the packet can be used to determine if a Graft was sent to this particular upstream router. If no Graft was sent, the Graft Ack can simply be ignored.

If a Graft was sent, and the acknowledgment has come from the correct upstream router, then it has been successfully received and the retransmission timer for the Graft can be stopped.

3.6.7 Graft Acknowledgment Packet Format

The format of a Graft Ack packet (which is identical to that of a Graft packet is show below:

Table 8 - Graft Ack Packet Format

0 31

Source Address
Group Address

3.7 Loop Detection and Suppression

3.8 Interfaces

3.8.1 IP Tunnels

3.8.2 Parameters

3.8.3 Metric

3.8.4 Threshold

3.8.5 Scope Control

4. Security Considerations

Security considerations will be discussed in an upcoming revision of this document.

5. References

[1]. Deering, S., Cheriton, D., "Multicast Routing in Datagram Internetworks and Extended LANs", ACM Transactions on Computer Systems, Vol. 8, No. 2, May 1990, Pages 85-110.

[2]. Waitzman, D., Partridge, C., Deering, S., "Distance Vector Multicast Routing Protocol", RFC 1075, Internet Architecture Board, November 1988.

[3]. Deering, S., "Host Extensions for IP Multicasting", RFC 1112, Internet Architecture Board, August 1989.

[4]. Deering, S., "Host Extensions for IP Multicasting - Appendix 1 (IGMP)", RFC 1112, Internet Architecture Board, August 1989.

6. Author's Address

Thomas Pusateri
NetEdge Systems, Inc.
PO Box 14993
Research Triangle Park, NC 27709

Phone: 919-991-9028
Fax: 919-991-9060
EMail: pusateri@NetEdge.COM

7. Appendix A - Interaction with IGMP (V1 & V2)

8. Appendix B - Constants & Configurable Parameters

9. Appendix C - Tracing and Troubleshooting support

Table 9 - Tracing and Debugging Packet Types

Code	Packet Type	Description
3	DVMRP Ask Neighbors	Obsolete
4	DVMRP Neighbors	Obsolete
5	DVMRP Ask Neighbors 2	Request Neighbor List
6	DVMRP Neighbors 2	Respond with Neighbor List
13	DVMRP Info	General Information Request/Reply

9.1 DVMRP Ask Neighbors

9.2 DVMRP Neighbors

9.3 DVMRP Ask Neighbors2

9.4 DVMRP Neighbors2

9.5 DVMRP Info message

10. Appendix D - Version History