T. Pusateri
Juniper Networks

INTERNET DRAFT
Obsoletes: RFC 1075
draft-ietf-idmr-dvmrp-v3-11
October 22, 2003
Expires: April 22, 2004

Distance Vector Multicast Routing Protocol

Status of this Memo

Abstract

DVMRP is an Internet routing protocol that provides an efficient mechanism for connection-less datagram delivery to a group of hosts across an internetwork. It is a distributed protocol that dynamically generates IP Multicast delivery trees using a technique called Reverse Path Multicasting (RPM) [Deer90]. This document is an update to Version 1 of the protocol specified in RFC 1075 [Wait88].

1. Introduction

DVMRP uses a distance vector distributed routing algorithm in order to build per-source-group multicast delivery trees. A good introduction to distance vector routing can be found in [Perl92]. The application of distance vector routing to multicast tree formulation is described in [Deer91].

1.1. Requirements Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC-2119].

1.2. Reverse Path Multicasting

Datagrams follow multicast delivery trees from a source to all members of a multicast group [Deer89], 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 DVMRP routing table of known source networks. If the datagram arrives on an interface that would be used to transmit 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 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.

1.3. Tunnel Encapsulation

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.

In previous implementations, DVMRP protocol messages were sent un-encapsulated to the unicast tunnel endpoint address. While this was more direct, it increased the complexity of firewall configuration. The most noticeable change in this specification regarding tunnels is that all DVMRP protocol messages should be sent encapsulated across the tunnel. Previously, protocol messages were sent un-encapsulated directly to the tunnel endpoint. See Appendix C for backward compatibility issues.

Note: All protocol messages sent on point-to-point links (including tunnels) should use a destination address of All-DVMRP-Routers. This change will allow the protocol messages to be forwarded across multicast-only tunnels without making encapsulation and decapsulation difficult.

In practice, tunnels typically use either IP-IP [Perk96] or Generic Routing Encapsulation (GRE) [Han94a,Han94b], although, other encapsulation methods are acceptable.

1.4. 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 inter-operable implementations. Included in Appendix A, is a summary of the DVMRP parameters. A section on DVMRP support for tracing and troubleshooting is the topic of Appendix B. Finally, a short DVMRP version compatibility section is provided in Appendix C to assist with backward compatibility issues.

2. Protocol Overview

DVMRP can be summarized as a "broadcast & prune" multicast routing protocol. It builds per-source broadcast trees based upon routing exchanges, then dynamically creates per-source-group multicast delivery trees by pruning (removing branches from) the source's truncated broadcast tree. It performs Reverse Path Forwarding checks to determine when multicast traffic should be forwarded to downstream

interfaces. In this way, source-rooted shortest path trees can be formed to reach all group members from each source network of multicast traffic.

2.1. Neighbor Discovery

Neighbor DVMRP routers are discovered dynamically by sending Neighbor Probe Messages on local multicast capable network interfaces and tunnel pseudo interfaces. These messages are sent periodically to the All-DVMRP-Routers [Reyn94] IP Multicast group address. (See Appendix C for backwards compatibility issues.) The IP TTL of these messages MUST be set to 1.

Each Neighbor Probe message contains the list of Neighbor DVMRP routers for which Neighbor Probe messages have been received on that interface. In this way, Neighbor DVMRP routers can ensure that they are seen by each other.

Once you have received a Probe from a neighbor that contains your address in the neighbor list, you have established a two-way neighbor adjacency with this router.

2.2. Source Location

When an IP Multicast datagram is received by a router running DVMRP, it first looks up the source network in the DVMRP routing table. The interface on which the best route to the source of the datagram was received is called the upstream (also called RPF) 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 path back to a source, a 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 requires an interface metric to be configured on all physical and tunnel interfaces. When a route is received, the metric of the interface over which the datagram was received must be added to the metric of the route being advertised in the route report message. This adjusted metric should be used when comparing metrics

to determine the best upstream neighbor.

Although there is certainly additional overhead associated with propagating a separate DVMRP routing table, it does provide two nice features. First, since all DVMRP routers are exchanging the same routes, 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 black holes due to disagreement between neighbor routers on the upstream interface.

Second, by propagating its own routing table, DVMRP makes it convenient to have separate paths for unicast versus 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.

2.3. Dependent Downstream Routers

In addition to providing a consistent view of source networks, the exchange of routes in DVMRP provides one other important feature. DVMRP uses the 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 technique called "Poison Reverse". If a downstream router selects an upstream router as the best next hop to a particular source network, this is indicated by echoing back the route on the upstream interface 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 twice 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 IP source specific multicast trees.

2.4. Designated Forwarder

When two or more multicast routers are connected to a multi-access network, it could be possible for duplicate packets to be forwarded on the network (one copy from each router). DVMRP prevents this possibility by electing a forwarder for each source as a side effect of its route exchange. When two routers on a multi-access network exchange source networks, each of the routers will know the others metric back to each source network. Therefore, of all the DVMRP routers on a shared network, the router with the lowest metric to a source network is responsible for forwarding data on to the shared network. If two or more routers have an equally low metric, the router with the lowest IP address becomes the designated forwarder for the network. In this way, DVMRP does an implicit designated forwarder election for each source network on each downstream interface.

2.5. Building Multicast Trees

As previously mentioned, when an IP multicast datagram arrives, the upstream interface is determined by looking up the interface on which the best route to the source of the datagram was received. If the upstream interface is correct, then a DVMRP router will forward the datagram to a list of downstream interfaces.

2.5.1. Adding Local Group Members

The IGMP local group database is maintained by all IP multicast routers on each physical, multicast capable network [Cain02]. If the destination group address is listed in the local group database, and the router is the designated forwarder for the source, then the interface is included in the list of downstream interfaces. If there are no group members on the interface, then the interface is removed from the outgoing interface list.

2.5.2. Adding Interfaces with Neighbors

Initially, all interfaces with downstream dependent neighbors should be included in the downstream interface list when a forwarding cache entry is first created. This allows the downstream routers to be aware of traffic destined for a particular (source network, group)

pair. The downstream routers will then have the option to send prunes and subsequent grafts for this (source network, group) pair as requirements change from their respective downstream routers and local group members.

2.6. Pruning Multicast Trees

As mentioned above, routers at the edges will remove their interfaces that have no group members associated with an IP multicast datagram. If a router removes all of its downstream interfaces, it notifies the upstream router that it no longer wants traffic destined for a particular (source network, 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 DVMRP 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 remove this interface from its downstream interface list. If the upstream router is able to remove all of its downstream interfaces in this way, it can then send a DVMRP Prune message to its upstream router. This continues until the unneeded branches are removed from the delivery tree.

In order to remove old prune state information for (source network, group) pairs that are no longer active, it is necessary to limit the life of a prune and periodically resume the broadcasting procedure. The prune message contains a prune lifetime, indicating the length of time that the prune should remain in effect. When the prune lifetime expires, the interface is joined back onto the multicast delivery tree. If unwanted multicast datagrams continue to arrive, the prune mechanism will be re-initiated and the cycle will continue. If all of the downstream interfaces are removed from a multicast delivery tree causing a DVMRP Prune message to be sent upstream, the lifetime of the prune sent must be equal to the minimum of the remaining lifetimes of the received prunes.

2.7. Grafting Multicast Trees

Once a tree branch has been pruned from a multicast delivery tree, packets from the corresponding (source network, group) pair will no longer be forwarded. However, since IP multicast supports dynamic group membership, hosts may join a multicast group at any time. In this case, DVMRP routers use Grafts to cancel the prunes that are in place from the host back on to the multicast delivery tree. A router will send a Graft message to its upstream neighbor if a group join occurs for a group that the router has previously sent a prune. Separate Graft messages must be sent to the appropriate upstream neighbor for each source network 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 using binary exponential back-off between retransmissions. Duplicate Graft Ack messages should simply be ignored. The purpose of the Graft Ack message is to simply acknowledge the receipt of a Graft message. It does not imply that any action was taken as a result of receiving the Graft message. Therefore, all Graft messages received from a neighbor with whom a two-way neighbor relationship has been formed should be acknowledged whether or not they cause an action on the receiving router.

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 [Reyn94]. All fields are transmitted in Network Byte Order. DVMRP packets use a common protocol header that specifies the IGMP [Cain02] Packet Type as hexadecimal 0x13 (DVMRP). DVMRP protocol packets should be sent with the Precedence field in the IP header set to Internetwork Control (hexadecimal 0xc0 for the Type of Service Octet) [Post81]. A diagram of the common protocol header follows:

```
 0         8          16               31
+---------+---------+--------------------+
| Type    |  Code   |      Checksum      |
|(0x13)   |         |                    |
+---------+---------+----------+---------+
|     Reserved      |  Minor   | Major   |
|                   | Version  |Version  |
+-------------------+----------+---------+
```

Figure 1 - Common Protocol Header

A Major Version of 3 and a Minor Version of 0xFF should be used to indicate compliance with this specification. 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:

Code	Packet Type	Description
1	DVMRP Probe	for neighbor discovery
2	DVMRP Report	for 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

Table 1 - Standard Protocol Packet Types

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 MUST be calculated upon transmission and MUST be validated on reception of a packet. The checksum of the DVMRP message is calculated with the checksum field set to zero. See [Brad88] for more information.

3.2. Probe Messages

When a DVMRP router is configured to run on an interface (physical or tunnel), it multicasts DVMRP Probe packets to inform other DVMRP routers that it is operational. Effectively, they serve three purposes.

1. Probes provide a mechanism for DVMRP routers to locate each other. DVMRP sends on each interface, a Probe Message containing the list of the neighbors detected for that specific interface. If no DVMRP neighbors are found, the network is considered to be a leaf network.

2. Probes 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. These flags were first introduced to allow optional protocol features. This specification now mandates the use of Generation Id's and pruning and, therefore, provides no optional capabilities. Other capability flags were used for tracing and troubleshooting and are no longer a part of the actual protocol.

3. Probes provide a keep-alive function in order to quickly detect neighbor loss. Probes sent on each multicast capable interface configured for DVMRP SHOULD use an interval of 10 seconds. The neighbor time-out interval SHOULD be set at 35 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 inter-operate with these older implementations that don't formally specify their capabilities and are not compliant with this specification. 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 an well known implementation (Mrouted version 3.5) in an attempt

to take the guess work out which features are supported by a neighbor. Many of these flags are no longer necessary since they are now a required part of the protocol, however, special consideration is necessary to not confuse older implementations that expect these flags to be set. Appendix C was written to assist with these and other backward compatibility issues.

Three of the flags were used for actual protocol operation. The other two assigned flags were used for troubleshooting purposes which should be documented in a separate specification. All of the bits marked "U" in the Figure below are now unused. They may be defined in the future and MUST be set to 0 on transmission and ignored on reception. Bit position 0 is the LEAF bit which is a current research topic. It MUST be set to 0 and ignored on reception. Bit positions 1, 2, and 3 MUST be set to 1 for backward compatibility. They were used to specify the PRUNE, GENID, and MTRACE bits. The first two, PRUNE and GENID, are now required features. The MTRACE bit must be set so existing implementations will not assume this neighbor does not support multicast trace-route [Fenn00]. However, since this bit is now reserved and set to 1, newer implementations should not use this bit in the Probe message to determine if multicast trace-route is supported by a neighbor. Instead, the M bit should only be used in a Neighbors2 message as described in Appendix B. The bit marked S stands for SNMP capable. This bit is used by troubleshooting applications and should only be tested in the Neighbors2 message.

The N bit (which stands for Netmask) is defined by this specification. It is used to indicate the neighbor will accept network masks appended to the Prune, Graft, and Graft Ack messages. This bit only indicates that the neighbor understands the netmask. It DOES NOT mean that Prune, Graft, and Graft Ack messages sent to this neighbor must include a netmask. Refer to the sections on Prune, Graft, and Graft Ack messages for more details.

Each time a Probe message is received from a neighbor, the capabilities bits should be compared to the previous version for that neighbor in order to detect changes in neighbor capabilities.

```
 7   6   5    4   3   2   1    0
 U   U   N    S   M   G   P    L
+---+---+---+---+----+---+---+---+
|0  |0  |X  | 0 | 1  |1  |1  | 0 |
+---+---+---+---+----+---+---+---+
```

Figure 2 - Probe Capability Flags

3.2.2. Generation ID

If a DVMRP router is restarted, it will not be aware of any previous prunes that it had sent or received. In order for the neighbor to detect that the router has restarted, a non-decreasing number is placed in the periodic probe message called the generation ID. When a change in the generation ID is detected, any prune information received from the router is no longer valid and should be flushed. If this prune state has caused prune information to be sent upstream, a graft will need to be sent upstream just as though a new member has joined below. Once data begins to be delivered downstream, if the downstream router again decides to be pruned from the delivery tree, a new prune can be sent upstream at that time.

In addition, the effects of a restart can be minimized if the router can learn all of the routes known by its neighbors without having to wait for an entire report interval to pass. When a router detects a change in the generation ID of a neighbor, it should send a unicast copy of its entire routing table to the neighbor.

In addition to restarting, a router may also miss prune information while an interface has transitioned to a down state. Therefore, a change in the generation ID is necessary when an interface transitions to the up state. In order to prevent all prune state from being flushed on a router when a single interface transitions, a DVMRP router should keep separate generation ID numbers per interface.

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.

In order to minimize one-way neighbor relationships, a router MUST delay sending poison route reports in response to routes advertised by a neighbor until the neighbor includes the routers address in its probe messages. On point-to-point interfaces and tunnel pseudo-interfaces, this means that no packets should be forwarded onto these interfaces until two-way neighbor relationships have formed.

Implementations written before this specification will not wait before sending reports nor will they ignore reports sent. Therefore, reports from these implementations SHOULD be accepted whether or not a probe with the routers address has been received.

3.2.4. Neighbor Expiry

When a neighbor expires, the following steps should be taken:

1. All routes learned from this neighbor should be immediately placed in hold-down. All downstream dependencies ON this neighbor should be removed.

2. If this neighbor is considered to be the designated forwarder for any of the routes it is advertising, a new designated forwarder for each source network should be selected.

3. Any forwarding cache entries based on this upstream neighbor should be flushed.

4. Any outstanding Grafts awaiting acknowledgments from this router should be flushed.

5. All downstream dependencies received FROM this neighbor should be removed. Forwarding cache entries should be checked to see if this is the last downstream dependent neighbor on the interface. If so, and this router isn't the designated forwarder (with local group members present), the interface should be removed.

   It is possible as an optimization to send a prune upstream if this causes the last downstream interface to be removed. However, this prune could be unnecessary if no more traffic is arriving. It is also acceptable to simply wait for traffic to arrive before sending the prune upstream.

3.2.5. Probe Packet Format

The Probe packet is variable in length depending on the number of neighbor IP addresses included. The length of the IP packet can be

used to determine the number of neighbors in the Probe message. The current Major Version is 3.

```
                      7             15        23         31
              +---------+--------------+--------------------+
              |  Type   |     Code     |      Checksum      |
              | (0x13)  |    (0x1)     |                    |
              +---------+--------------+----------+---------+
              |         |              |          |         |
              |Reserved | Capabilities |  Minor   | Major   |
              +---------+--------------+----------+---------+
              |                                             |
              |               Generation ID                 |
              +---------------------------------------------+
              |                                             |
              |           Neighbor IP Address 1             |
              +---------------------------------------------+
              |                                             |
              |           Neighbor IP Address 2             |
              +---------------------------------------------+
              |                                             |
              |                     ...                     |
              +---------------------------------------------+
              |                                             |
              |           Neighbor IP Address N             |
              +---------------------------------------------+
```

Figure 3 - DVMRP Probe Packet Format

Generation ID
This field contains a non-decreasing number used to detect a change in neighbor state.

Neighbor IP Address N
This is a list of Neighbor IP addresses whom the sending router has received Probe messages from.

3.2.6. IGMP Designated Querier Election

Since it is wasteful to have more than a single router sending IGMP Host Membership Queries on a given physical network, a single router on each physical network is elected as the Designated Querier. This election was formerly a part of DVMRP. However, this is now specified as a part of the IGMP version 2 protocol. See Appendix C for details on backwards compatibility.

Even though only one router will act as the IGMP designated querier, all DVMRP routers must use IGMP to learn local group memberships.

3.3. Multicast Forwarding

DVMRP can forward multicast packets by building the downstream interface list for each packet as it arrives. However, to reduce per packet processing time, the result of the first lookup MAY be cached in a forwarding table. Then, as routes, downstream dependent neighbors, or group membership change, the cache forwarding table entries MUST be updated to reflect these changes.

3.3.1. Designated Forwarder

Initially, a DVMRP router should assume it is the designated forwarder for all source networks on all downstream interfaces. As it receives route reports, it can determine if other routers on multi-access networks have better routes back to a particular source network. A route is considered better if the adjusted received metric is less than the metric that it will advertise for the source network on the received interface or if the metrics are the same but the IP address of the neighbor is lower.

If this neighbor becomes unreachable or starts advertising a worse metric, then the router should become the designated forwarder for this source network again on the downstream interface until it hears from a better candidate.

If the upstream RPF interface changes, then the router should become the designated forwarder on the previous upstream interface (which is now a potential downstream interface) until it hears from a better candidate.

3.3.2. Determining the upstream interface

When a multicast packet arrives, a DVMRP router will use the DVMRP routing table to determine which interface leads back to the source. If the packet did not arrive on that interface, it MUST 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 routing table.

3.3.3. Determining the downstream interface list

The downstream interface list is built by starting with the list of non-leaf interfaces. The upstream interface MUST be removed from this list. Then any interfaces on the list where all of the downstream dependents have sent prunes upstream MUST be removed. Next, any interfaces for which the router is the designated forwarder and local group members are present MUST be added to the list.

3.4. Route Exchange

The routing information propagated by DVMRP is used for determining the reverse path neighbor back to the source of the multicast traffic. The interface used to reach this neighbor is called the upstream interface. Tunnel pseudo-interfaces are considered to be distinct from the physical interface on which the packet is actually transmitted for the purpose of determining upstream and downstream interfaces.

The routing information that is propagated by DVMRP contains a list of source networks and an appropriate metric. The metric used is a hop count which is incremented by the cost of the incoming interface metric. 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 to create symmetric routing and provide for easier problem determination although the protocol does not strictly enforce this.

3.4.1. Source Network Aggregation

Implementations may wish to provide a mechanism to aggregate source networks to reduce the size of the routing table. All implementations should be able to accept reports for aggregated source networks in accordance with Classless Inter-Domain Routing (CIDR) as described in [Rekh93] and [Full93].

There are two places where aggregation is particularly useful.

1. At organizational boundaries to limit the number of source networks advertised out of the organization.

2. Within an organization to summarize non-local routing information by using a default (0/0) route.

If an implementation wishes to support source aggregation, it MUST transmit Prune and Graft messages according to the following rules:

A. If a Prune is received on a downstream interface for which the source network advertised to that neighbor is an aggregate, then if a prune is sent upstream, it should only be sent for the contributing route based on the source address in the received prune.

   If additional data is received for sources within the range of the aggregate, then this SHOULD trigger additional prunes to be sent upstream for these sources.

   There may be active forwarding cache entries for other contributing routes to the aggregate. Prunes should not be sent upstream to the contributing routes that have no forwarding state.

B. If a Graft is received on a downstream interface for which the source network advertised to that neighbor is an aggregate generated by the receiving router, then Graft messages MUST be sent upstream (if necessary) for each route that contributed to the aggregate that had been previously pruned.

3.4.2. 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. There was an earlier requirement that Route Reports MUST contain source network/mask pairs sorted first by increasing network mask and then by increasing source network. This restriction has been lifted. Implementations conforming to this specification MUST be able to receive Route Reports containing any mixture of network masks and source networks.

In order to pack more source networks into a route report, source networks are often represented by less than 4 octets. The number of non-zero 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. This method of specifying source network masks is compatible with techniques described in [Rekh93] and [Full93] to group traditional Class C networks into super-nets and to allow different subnets of the same Class A network to be discontinuous. It does not, however, allow grouping class A networks into super-nets since the first octet of the netmask is always assumed to be 255.

In this notation, the default route is represented as the least three significant octets of the netmask [00 00 00], followed by one octet for the network number [00]. This special case MUST be interpreted as 0.0.0.0/0.0.0.0 and NOT 0.0.0.0/255.0.0.0.

3.4.3. Route Metrics

For each source network reported, a route metric is associated with the route being reported. The metric is the sum of the interface metrics between the router originating the report and the source network. For the purposes of DVMRP, 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, a "Last" bit is defined as the high order bit of the octet following

the network address. This bit is set to signify when the last route is being reported for a particular mask value. When the "Last" bit is set and 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.

3.4.4. 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 broadcasted 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 removed when the router has received Prune messages from each of the dependent routers on that interface. Each downstream router uses Poison Reverse to indicate for which source networks it is dependent upon 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 x Infinity - 1) or 1 and 63. Values between 1 and 31 indicate reachable source networks. The value Infinity (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.

3.4.5. Sending Route Reports

All of the active routes MUST be advertised over all interfaces with neighbors present each Route Report Interval. In addition, flash updates MAY be sent as needed but flash updates MUST NOT happen more often than the Minimum Flash Update Interval (5 seconds). Flash updates 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.

When a router sees its own address in a neighbor probe packet for the first time, it should send a unicast copy of its entire routing table to the neighbor to reduce start-up time.

Reports should not be sent to a neighbor until a router has seen its own address in the neighbors Probe router list. See Appendix C for exceptions.

3.4.6. Receiving Route Reports

After receiving a route report, a check should be made to verify it is from a known neighbor. Two-way neighbor relationships are essential for proper DVMRP operation. Therefore, route reports from unknown neighbors MUST be discarded.

In the following discussion, "Metric" refers to the metric of the route as received in the route report. "Adjusted Metric" refers to the metric of the route after the incoming interface metric has been added.

If the metric received is less than infinity but the Adjusted Metric is greater than or equal to infinity, the Adjusted Metric should be set to infinity.

If the metric is greater than or equal to infinity, then no adjustment of the metric should be made.

Each route in the report is then parsed and processed according to the following rules:

A. If the route is new and the Adjusted Metric is less than infinity, the route should be added.

B. If the route already exists, several checks must be performed.

1. Received Metric < infinity

   If the neighbor was considered a downstream dependent neighbor, the dependency is canceled.

   In the following cases, the designated forwarder on one of the downstream interfaces should be updated:

   - If the Metric received would cause the router to advertise a better metric on a downstream interface than the existing designated forwarder for the source network on that interface (or advertised metric would be the same but the router's IP address is lower than the existing designated forwarder on that interface). Then the receiving router

becomes the new designated forwarder for that source network on that interface. If this router had sent a prune upstream that is still active, it will need to send a graft.

- If the metric being advertised by the current designated forwarder is worse than the receiving routers metric that it would advertise on the receiving interface (from learning the same route from a neighbor on another interface) or the metric is the same but the receiving router has a lower IP address, then the receiving router becomes the new designated forwarder on that interface. This may trigger a graft to be sent upstream.

- If the metric received for the source network is better than the metric of the existing designated forwarder, save the new designated forwarder and the metric it is advertising. It is necessary to maintain knowledge of the current designated forwarder for each source network in case the time-out value for this neighbor is reached. If the time-out is reached, then the designated forwarder responsibility for the source network should be assumed.

A route is considered better when either the received Metric is lower than the existing metric or the received Metric is the same but the advertising router's IP address is lower.

a. Adjusted Metric > existing metric

If the Adjusted Metric is greater than the existing metric, then check to see if the same neighbor is reporting the route. If so, update the route metric and schedule a flash update containing the route. Otherwise, skip to the next route in the report.

b. Adjusted Metric < existing metric

Update the metric for the route and if the neighbor reporting the route is different, update the upstream neighbor in the routing table. Schedule a flash update containing the route to downstream neighbors and a flash poison update containing the route should be sent upstream indicating a change in downstream dependency (even if its on the same upstream interface).

c. Adjusted metric = existing metric

If the neighbor reporting the route is the same as the existing upstream neighbor, then simply refresh the route.

If the neighbor is the same and the route is in hold-down, it is permissible to prematurely take the route out of hold-down and begin advertising it with a non-infinity metric. If the route is taken out of hold-down, a flash update containing the route should be scheduled on all DVMRP interfaces except the one over which it was received.

If the neighbor reporting the route has a lower IP address than the existing upstream neighbor, then switch to this neighbor as the best route. Schedule a flash update containing the route to downstream neighbors and a flash poison update containing the route should be sent upstream indicating a change in downstream dependency (even if its on the same upstream interface).

2. Received Metric = infinity

   If the neighbor was considered to be the designated forwarder, the receiving router should now become the designated forwarder for the source network on this interface.

   a. Next hop = existing next hop

      If the existing metric was less than infinity, the route is now unreachable. Delete the route and schedule a flash update containing the route to all interfaces for which you are the designated forwarder or have downstream dependents.

   b. Next hop != existing next hop

      The route can be ignored since the existing next hop has a metric better than or equal to this next hop.

      If the neighbor was considered a downstream dependent neighbor, this should be canceled.

3. infinity < Received Metric < 2 x infinity

   The neighbor considers the receiving router to be upstream for the route and is indicating it is dependent on the receiving router.

   If the neighbor was considered to be the designated forwarder, the receiving router should now become the designated forwarder for the source network on this interface.

a. Neighbor on downstream interface

If the sending neighbor is considered to be on a downstream interface for that route then the neighbor is to be registered as a downstream dependent router for that route.

If this is the first time the neighbor has indicated downstream dependence for this source and one or more prunes have been sent upstream containing this source network, then Graft messages MUST be sent upstream in the direction of the source network for each group with existing prune state.

b. Neighbor on upstream interface

If the receiving router thinks the neighbor is on the upstream interface, then the route should be treated as if an infinity metric was received (See Received Metric = infinity).

4. 2 x infinity <= Received Metric

If the Received Metric is greater than or equal to 2 x infinity, the Metric is illegal and the route should be ignored.

3.4.7. Route Expiration

A route expires if it has not been refreshed within the Route Expiration time. This should be set to 2 x Route Report Interval + 20 (or 140) seconds. Due to flash updates, routes will typically not expire but instead be removed in response to receiving an infinity metric for the route. However, since not all existing implementations implement flash updates, route expiration is necessary.

3.4.8. Route Hold-down

When a route is deleted (because it expires, the neighbor it was learned from goes away, or an infinity metric is received for the route) a router may be able to reach the source network described by the route through an alternate gateway. However, in the presence of complex topologies, often, the alternate gateway may only be echoing back the same route learned via a different path. If this occurs, the

route will continue to be propagated long after it is no longer valid.

In order to prevent this, it is common in distance vector protocols to continue to advertise a route that has been deleted with a metric of infinity for one or more report intervals. This is called Hold-down. A route MUST only be advertised with an infinity metric while it is in hold-down. The hold-down period is 2 Report Intervals.

When a route goes into hold-down, all forwarding cache entries based on the route should be flushed.

During the hold-down period, the route may be learned via a different gateway or the same gateway with a different metric. The router MAY use this new route for delivering to local group members. Also, installing a new route during the hold-down period allows the route to be advertised with a non-infinity metric more quickly once the hold-down period is over.

In order to minimize outages caused by flapping routes, it is permissible to prematurely take a route out of hold-down ONLY if the route is re-learned from the SAME router with the SAME metric.

Route hold-down is not effective if only some routers implement it. Therefore, it is now a REQUIRED part of the protocol.

3.4.9. Graceful Shutdown

During a graceful shutdown, an implementation MAY want to inform neighbor routers that it is terminating. Routes that have been advertised with a metric less than infinity should now be advertised with a metric equal to infinity. This will allow neighbor routers to switch more quickly to an alternate path for a source network if one exists.

Routes that have been advertised with a metric between infinity and 2 x infinity (indicating downstream neighbor dependence) should now be advertised with a metric equal to infinity (canceling the downstream dependence).

3.4.10. Route Report Packet Format

The format of a sample Route Report Packet is shown in Figure 4 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.

```
                    7           15           23           31
          +-----------+------------+-------------------------+
          |   Type    |    Code    |        Checksum         |
          |  (0x13)   |   (0x2)    |                         |
          +-----------+------------+------------+------------+
          |       Reserved         |   Minor    |   Major    |
          |                        |  Version   |  Version   |
          +-----------+------------+------------+------------+
          |  Mask1    |   Mask1    |   Mask1    |    Src     |
          |  Octet2   |   Octet3   |   Octet4   |   Net11    |
          +-----------+------------+------------+------------+
          |  SrcNet11(cont.)...    |  Metric11  |    Src     |
          |                        |            |   Net12    |
          +------------------------+------------+------------+
          |  SrcNet12(cont.)...    |  Metric12  |   Mask2    |
          |                        |            |   Octet2   |
          +-----------+------------+------------+------------+
          |  Mask2    |   Mask2    |        SrcNet21         |
          |  Octet3   |   Octet4   |                         |
          +-----------+------------+------------+------------+
          |  SrcNet21(cont.)...    |  Metric21  |   Mask3    |
          |                        |            |   Octet2   |
          +-----------+------------+------------+------------+
          |  Mask3    |   Mask3    |           ...           |
          |  Octet3   |   Octet4   |                         |
          +-----------+------------+-------------------------+
```

Figure 4 - Example Route Report Packet Format

3.5. Pruning

DVMRP is described as a broadcast and prune multicast routing protocol since datagrams are initially sent out all dependent downstream interfaces forming a tree rooted at the source of the data. As the routers at the leaves of the tree begin to receive unwanted multicast traffic, they send prune messages upstream toward the source. This results in the multicast tree for a given source

network and a given set of receivers.

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 dependent routers. This detection mechanism is covered above in section 3.2 titled Probe Messages. If there are no group members present for a particular multicast datagram received, the leaf routers will start the pruning process by removing their downstream interfaces and sending a prune to the upstream router for that source.

3.5.2. Source Networks

By default, prunes are meant to be applied to a group and source network. However, it is possible to include a Netmask in the Prune message to alter this behavior. If no Netmask is included, a prune sent upstream triggered by traffic received from a particular source applies to all sources on that network. If a Netmask is included, it MUST first be validated. If the Netmask is a host mask, only that source address should be pruned. Otherwise, the Netmask MUST match the mask sent to the downstream router for that source. If it does not match the mask that the upstream router expected, the upstream router MUST ignore the prune and should log an error. When a aggregate source network is advertised downstream, the Netmask in the prune will match the mask of the aggregate route that was advertised.

If the Prune message only contains the host address of the source (and not the corresponding Netmask), the source network can be determined easily by a best-match lookup using the routing table distributed as a part of DVMRP.

3.5.3. Receiving a Prune

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

1. If the neighbor is unknown, discard the received prune.

2. Ensure the prune message contains at least the correct amount of data.

3. Copy the source address, group address, and prune time-out value. If it is available in the packet, copy the Netmask value. Determine route to which prune applies.

4. If there is no active source information for the (source network, group) pair, then ignore the prune.

5. Verify that the prune was received from a dependent neighbor for the source network. If not, discard the prune.

6. Determine if a prune is currently active from the same dependent neighbor for this (source network, group) pair.

7. 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.

8. Determine if all dependent downstream routers on the interface from which the prune was received have now sent prunes.

9. If so, then determine if there are group members active on the interface and if this router is the designated forwarder for the network.

10. If not, then remove the interface from all forwarding cache entries for this group instantiated using the route to which the prune applies.

3.5.4. Sending a Prune

When a forwarding cache is being used, there is a trade-off that should be considered when deciding when Prune messages should be sent upstream. In all cases, when a data packet arrives and the downstream interface list is empty, a prune is sent upstream. However, when a forwarding cache entry transitions to an empty downstream interface list it is

possible as an optimization to send a prune at this time as well. This prune will possibly stop unwanted traffic sooner at the expense of sending extra prune traffic for sources that are no longer sending.

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 default prune lifetime (randomized to prevent synchronization) and the remaining prune lifetimes of the downstream neighbors.

5. Form and transmit the packet to the upstream neighbor for the source.

3.5.5. Retransmitting a Prune

By increasing the prune lifetime to ~2 hours, the effect of a lost prune message becomes more apparent. Therefore, an implementation SHOULD retransmit prunes messages using binary exponential back-off during the lifetime of the prune if traffic is still arriving on the upstream interface.

One way to implement this would be to send a prune, install a negative cache entry for 3 seconds while waiting for the prune to take effect. Then remove the negative cache entry. If traffic continues to arrive, a new forwarding cache request will be generated. The prune can be resent with the remaining prune lifetime and a negative cache entry can be installed for 6 seconds. After this, the negative cache entry is removed. This procedure is repeated while each time doubling the length of time the negative cache entry is installed.

In addition to using binary exponential back-off, the interval between subsequent retransmissions should also be randomized to prevent synchronization.

On multi-access networks, even if a prune is received by the upstream

router, data may still be received due to other receivers (i.e. group members or other downstream dependent routers) on the network.

3.5.6. Prune Packet Format

A Prune Packet contains three required fields: the source host IP address, the destination group IP address, and the Prune Lifetime in seconds. An optional source network mask may also be appended to the Prune message. This mask applied to the Source Host Address will indicate the route that the prune applies to. A Source Network Mask field should only be sent in a Prune message to a neighbor if that neighbor has advertised the ability to process it by setting the Netmask capabilities bit. The length of the Prune message will indicate if the Source Network Mask has been included or not.

The Prune Lifetime is a derived value calculated as the minimum of the default prune lifetime (2 hours) and the remaining lifetimes of any downstream prunes received for this source network and group. A router with no downstream dependent neighbors would use the the default prune lifetime (randomized to prevent synchronization).

```
                      7           15            23            31
            +-----------+------------+-------------------------+
            |   Type    |    Code    |        Checksum         |
            |  (0x13)   |   (0x7)    |                         |
            +-----------+------------+------------+------------+
            |       Reserved         |   Minor    |   Major    |
            +------------------------+------------+------------+
            |               Source Host Address                |
            +--------------------------------------------------+
            |                  Group Address                   |
            +--------------------------------------------------+
            |                 Prune Lifetime                   |
            +--------------------------------------------------+
            |               Source Network Mask                |
            +--------------------------------------------------+
```

Figure 5 - Prune Packet Format

Source Host Address
   The source host IP address of the datagram that triggered the prune.

Group Address
The destination group IP address of the datagram that triggered the prune.

Prune Lifetime
The number of seconds for which the upstream neighbor should keep this prune active.

Source Network Mask
The (optional) netmask of the route this prune applies to.

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 hop-by-hop from the new receiver's first-hop router until a point on the multicast tree is reached. Since there is no way to tell whether a graft message was lost or the source stopped sending, 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.

One or more Graft messages should be sent under the following conditions:

1. A new local member joins a group that has been pruned upstream and this router is the designated forwarder for the source.

2. A new dependent downstream router appears on a pruned branch.

3. A dependent downstream router on a pruned branch restarts (new Generation ID).

4. A Graft Retransmission Timer expires before a Graft-Ack is received.

3.6.1. Sending a Graft

Recall that by default, Prunes are source network specific and are sent up different trees for each source network. Grafts are sent in response to various conditions which are not necessarily source

specific. Therefore, it may be necessary to send separate Graft messages to the appropriate upstream routers to counteract each previous source network specific prune that was sent.

As mentioned above, a Graft message sent to the upstream DVMRP router should be acknowledged hop by hop guaranteeing end-to-end delivery. In order to send a Graft message, the following steps should be taken:

1. Verify a prune exists for the source network and group.

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.

If a Graft Acknowledgment is not received within the Graft Retransmission Time-out period, the Graft should be resent to the upstream router. The initial retransmission period is 5 seconds. A binary exponential back-off policy is used on subsequent retransmissions.

3.6.2. Receiving a Graft

1. If the neighbor is unknown, discard the received graft.

2. Ensure the graft message contains at least the correct amount of data.

3. Send back a Graft Ack to the sender.

4. If the sender was a downstream dependent neighbor from which a prune had previously been received, then remove the prune state for this neighbor. If necessary, any forwarding cache entries

based on this (source, group) pair should be updated to include this downstream interface.

5. If a prune had been sent upstream, this may trigger a graft to now be sent to the upstream router.

3.6.3. Graft Packet Format

The format of a Graft packet is show below:

```
                   7           15           23           31
         +-----------+------------+-------------------------+
         |   Type    |    Code    |        Checksum         |
         |  (0x13)   |   (0x8)    |                         |
         +-----------+------------+------------+------------+
         |       Reserved         |   Minor    |   Major    |
         +------------------------+------------+------------+
         |               Source Host Address                |
         +--------------------------------------------------+
         |                  Group Address                   |
         +--------------------------------------------------+
         |               Source Network Mask                |
         +--------------------------------------------------+
```

Figure 6 - Graft Packet Format

Source Host Address
The source host IP address indicating which source host or source network to Graft.

Group Address
The destination group IP address to be grafted.

Source Network Mask
The (optional) netmask of the route this graft applies to.

3.6.4. Sending a Graft Acknowledgment

A Graft Acknowledgment packet is sent to a downstream neighbor in response to receiving a Graft message. All Graft messages MUST 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.

3.6.5. Receiving a Graft Acknowledgment

When a Graft Acknowledgment is received, ensure the message contains at least the correct amount of data. The (source address, 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.

3.6.6. Graft Acknowledgment Packet Format

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

```
                      7           15           23           31
            +-----------+------------+-------------------------+
            |   Type    |    Code    |        Checksum         |
            |  (0x13)   |   (0x9)    |                         |
            +-----------+------------+------------+------------+
            |       Reserved         |   Minor    |   Major    |
            +------------------------+------------+------------+
            |               Source Host Address                |
            +--------------------------------------------------+
            |                  Group Address                   |
            +--------------------------------------------------+
            |               Source Network Mask                |
            +--------------------------------------------------+
```

Figure 7 - Graft Ack Packet Format

Source Host Address
The source host IP address that was received in the Graft message.

Group Address
The destination group IP address that was received in the Graft message.

Source Network Mask
The (optional) netmask of the route this Graft Ack applies to.

3.7. Interfaces

Interfaces running DVMRP will either be multicast capable physical interfaces or encapsulated tunnel pseudo-interfaces. Physical interfaces may either be multi-access networks or point-to-point networks. Tunnel interfaces are used when there are non-multicast capable routers between DVMRP neighbors. Protocol messages and multicast data traffic are sent between tunnel endpoints using a standard encapsulation method [Perk96,Han94a,Han94b]. The unicast IP addresses of the tunnel endpoints are used as the source and destination IP addresses in the outer IP header. The inner IP header remains unchanged from the original packet.

Protocol messages on point-to-point links should always use a destination IP address of All-DVMRP-Routers for ALL message types. While Prune, Graft, and Graft-Ack messages are only intended for a

single recipient, the use of a multicast destination address is necessary for un-numbered links and encapsulated interfaces.

When multiple addresses are configured on a single interface, it is necessary that all routers on the interface know about the same set of network addresses. In this way, each router will make the same choice for the designated forwarder for each source. In addition, a router configured with multiple addresses on an interface should consistently use the same address when sending DVMRP control messages.

The maximum packet length of any DVMRP message should be the maximum packet size required to be forwarded without fragmenting. The use of Path MTU Discovery [Mogu90] is encouraged to determine this size. In the absence of Path MTU, the Requirements for Internet Hosts [Brad89] specifies this number as 576 octets. Be sure to consider the size of the encapsulated IP header as well when calculating the maximum size of a DVMRP protocol message.

3.7.1. Interface transitions

When an interface transitions to the up state, the generation ID of that interface should be updated so that DVMRP neighbors know to resend prune information.

When an interface transitions to the down state, all neighbors on that interface should be expired. All actions associated with an expired neighbor should be taken as specified in the Neighbor Expiry section.

4. IANA Considerations

The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols. DVMRP uses IGMP [Cain02] IP protocol messages to communicate between routers. The IGMP Type field is hexadecimal 0x13.

On IP multicast capable networks, DVMRP uses the All-DVMRP-Routers local multicast group. This group address is 224.0.0.4.

5. Network Management Considerations

DVMRP provides several methods for network management monitoring and troubleshooting. Appendix B describes a request/response mechanism to directly query DVMRP neighbor information. In addition, a Management Information Base for DVMRP is defined in [Thal97].

A Management Information Base for the multicast forwarding cache is defined in [McCl00].

Also, a protocol independent multicast trace-route facility is defined in [Fenn00].

6. Security Considerations

Security for DVMRP follows the general security architecture provided for the Internet Protocol [Ken98a]. The IPsec authentication header [Ken98b] MAY be used to provide data integrity protection and groupwise data origin authentication of DVMRP protocol messages.

Currently, the IPsec anti-replay option does not handle the case of a Security Association identified by a multicast destination address. Thus, the anti-replay option currently must be disabled on these Security Associations. The anti-replay option SHOULD be enabled on all security associations having a unicast destination address.

There are only two DVMRP protocol message types sent to a multicast destination address. The effects of replaying these messages are outlined below:

DVMRP Probes

The Probe message contains two important state mechanisms. The first is the Generation ID. This is a non-decreasing number that allows the neighbors to detect if the router has been restarted. If an old Probe message is replayed, the Generation ID will either be the same or smaller than the current Generation ID. If it is smaller, then the replayed Probe will be ignored. If it is the same, then the message will continue to be processed.

The second state mechanism is the list of neighbors a router has learned. If a neighbor no longer appears in this list, then any existing prune information learned from this neighbor will be removed. This may cause multicast data to once again be flooded onto networks where it is not needed.

In addition, downstream dependent neighbors are based on the

neighbor list in the Probe message. If a neighbor no longer appears in this list, it will be removed from the downstream dependent list for each prefix that it expects to receive data from. Therefore, it is possible to stop data from being forwarded downstream by replaying an older Probe message that doesn't contain the neighbor address.

However, because the Probe messages are periodic, the replayed message would have to be continuously sent after each periodic Probe message that contains a valid neighbor list.

DVMRP Reports

DVMRP Report messages are sent with both unicast and multicast destination addresses. Report messages that have multicast destinations are periodic Reports containing prefixes learned by the router. If these Reports were to be replayed at a later time, they could disrupt a routers ability to correctly determine the upstream interface of a source and therefore, stop forwarding of multicast data.

Periodic Route Reports would continue containing correct information which could result in route flapping or holddown preventing multicast data from being forwarded for sources falling within the prefix ranges in the replayed Reports.

DVMRP Prune, Graft, and Graft-Ack messages use unicast destination addresses. Security Associations between neighbors sending and receiving these protocol message types can make full use of the anti-replay protection provided by the IP security protocols.

7. References

[Brad88] Braden, R., Borman, D., Partridge, C., "Computing the Internet Checksum", RFC 1071, September 1988.

[Brad89] Braden, R., "Requirements for Internet Hosts -- Communication Layers", RFC 1122, October 1989.

[Cain02] Cain, B., Deering, S., Kouvelas, I., Fenner, W., Thyagarajan, A., "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[Deer89] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, August 1989.

[Deer90] Deering, S., Cheriton, D., "Multicast Routing in Datagram Internetworks and Extended LANs", ACM Transactions on Computer Systems, Vol. 8, No. 2, May 1990, pp. 85-110.

[Deer91] Deering, S., "Multicast Routing in a Datagram Internetwork", PhD thesis, Electric Engineering Dept., Stanford University, December 1991.

[Fenn00] Fenner, W., Casner, S., "A "traceroute" facility for IP Multicast", Work In Progress, July 2000.

[Full93] Fuller, V., T. Li, J. Yu, and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519, September 1993.

[Han94a] Hanks, S., Li, T, Farinacci, D., and P. Traina, "Generic Routing Encapsulation", RFC 1701, NetSmiths, Ltd., and cisco Systems, October 1994.

[Han94b] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation over IPv4 networks", RFC 1702, NetSmiths, Ltd., cisco Systems, October 1994.

[Ken98a] Kent, S., Atkinson, R. "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[Ken98b] Kent, S., Atkinson, R., "IP Authentication Header", RFC 2402, November 1998.

[McCl00] McCloghrie, K., Farinacci, D., Thaler, D., "IPv4 Multicast Routing MIB", RFC 2932, October 2000.

[Mogu90] Mogul, J., Deering, S., "Path MTU Discovery", RFC 1191, November 1990.

[Perk96] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

[Perl92] Perlman, R., "Interconnections: Bridges and Routers", Addison-Wesley, May 1992, pp. 205-211.

[Post81] Postel, J., "Internet Protocol", RFC 791, September, 1981.

[Rekh93] Rekhter, Y., and T. Li, "An Architecture for IP Address Allocation with CIDR", RFC 1518, September 1993.

[Reyn94] Reynolds, J., Postel, J., "Assigned Numbers", STD 0002, October 1994.

[Thal97] Thaler, D., "Distance-Vector Multicast Routing Protocol MIB", Work In Progress, April 1997.

[Wait88] Waitzman, D., Partridge, C., Deering, S., "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988.

8. Author's Address

Thomas Pusateri
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: (919) 807-0023
EMail: pusateri@juniper.net

9. Acknowledgments

The author would like to acknowledge the original designers of the protocol, Steve Deering, Craig Partridge, and David Waitzman. Version 3 of the protocol would not have been possible without the original work of Ajit Thyagarajan and the ongoing (and seemingly endless) work of Bill Fenner. Credit also goes to Danny Mitzel and Dave Thaler for the careful review of this document and Nitin Jain, Dave LeRoy, Charles Mumford, Ravi Shekhar, and Shuching Shieh for their helpful comments.

10. Appendix A - Constants & Configurable Parameters

The following table provides a summary of the DVMRP timing parameters:

Parameter	Value (seconds)
Probe Interval	10
Neighbor Time-out Interval	35
Minimum Flash Update Interval	5
Route Report Interval	60
Route Expiration Time	140
Route Hold-down	2 x Route Report Interval
Prune Lifetime	variable (< 2 hours)
Prune Retransmission Time	3 with exp. back-off
Graft Retransmission Time	5 with exp. back-off

Table 2 - Parameter Summary

11. Appendix B - Tracing and Troubleshooting support

There are several packet types used to gather DVMRP specific information. They are generally used for diagnosing problems or gathering topology information. The first two messages are now obsoleted and should not be used. The remaining two messages provide a request/response mechanism to determine the versions and capabilities of a particular DVMRP router.

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

Table 3 - Debugging Packet Types

11.1. DVMRP Ask Neighbors2

The Ask Neighbors2 packet is a unicast request packet directed at a DVMRP router. The destination should respond with a unicast Neighbors2 message back to the sender of the Ask Neighbors2 message.

```
 0         8          16               31
+---------+---------+--------------------+
| Type    |  Code   |      Checksum      |
|(0x13)   | (0x5)   |                    |
+---------+---------+----------+---------+
|     Reserved      |  Minor   | Major   |
|                   | Version  |Version  |
+-------------------+----------+---------+
```

Figure 8 - Ask Neighbors 2 Packet Format

11.2. DVMRP Neighbors2

The format of a Neighbors2 response packet is shown below. This is sent as a unicast message back to the sender of an Ask Neighbors2 message. There is a common header at the top followed by the routers capabilities. One or more sections follow that contain an entry for each logical interface. The interface parameters are listed along with a variable list of neighbors learned on each interface.

If the interface is down or disabled, list a single neighbor with an address of 0.0.0.0 for physical interfaces or the remote tunnel endpoint address for tunnel pseudo-interfaces.

```
           0            8              16                    31
          +-----------+--------------+--------------------------+
          |   Type    |     Code     |         Checksum         |
          |  (0x13)   |    (0x6)     |                          |
          +-----------+--------------+------------+-------------+
          | Reserved  | Capabilities |   Minor    |    Major    |
          |           |              |  Version   |   Version   |
          +-----------+--------------+------------+-------------+
          |                                                     |
          |                    Local Addr 1                     |
          +-----------+--------------+------------+-------------+
          |           |              |            |             |
          | Metric 1  | Threshold 1  |  Flags 1   | Nbr Count 1 |
          +-----------+--------------+------------+-------------+
          |                                                     |
          |                       Nbr 1                         |
          +-----------------------------------------------------+
          |                                                     |
          |                         ...                         |
          +-----------------------------------------------------+
          |                                                     |
          |                       Nbr m                         |
          +-----------------------------------------------------+
          |                                                     |
          |                    Local Addr N                     |
          +-----------+--------------+------------+-------------+
          |           |              |            |             |
          | Metric N  | Threshold N  |  Flags N   | Nbr Count N |
          +-----------+--------------+------------+-------------+
          |                                                     |
          |                       Nbr 1                         |
          +-----------------------------------------------------+
          |                                                     |
          |                         ...                         |
          +-----------------------------------------------------+
          |                                                     |
          |                       Nbr k                         |
          +-----------------------------------------------------+
```

Figure 9 - Neighbors 2 Packet Format

The capabilities of the local router are defined as follows:

Bit	Flag	Description
0	Leaf	This is a leaf router
1	Prune	This router understands pruning
2	GenID	This router sends Generation Id's
3	Mtrace	This router handles Mtrace requests
4	Snmp	This router supports the DVMRP MIB

Table 4 - DVMRP Router Capabilities

The flags associated with a particular interface are:

Bit	Flag	Description
0	Tunnel	Neighbor reached via tunnel
1	Source Route	Tunnel uses IP source routing
2	Reserved	No longer used
3	Reserved	No longer used
4	Down	Operational status down
5	Disabled	Administrative status down
6	Querier	Querier for interface
7	Leaf	No downstream neighbors on interface

Table 5 - DVMRP Interface flags

12. Appendix C - Version Compatibility

There have been two previous major versions of DVMRP with implementations still in circulation. If the receipt of a Probe message reveals a major version of 1 or 2, then it can be assumed that this neighbor does not support pruning or the use of the Generation ID in the Probe message. However, since these older implementations are known to safely ignore the Generation ID and neighbor information in the Probe packet, it is not necessary to send specially formatted Probe packets to these neighbors.

There were three minor versions (0, 1, and 2) of major version 3 that did support pruning but did not support the Generation ID or capability flags. These special cases will have to be accounted for.

Any other minor versions of major version 3 closely compare to this specification.

In addition, cisco Systems is known to use their software major and minor release number as the DVMRP major and minor version number. These will typically be 10 or 11 for the major version number. Pruning was introduced in Version 11.

Implementations prior to this specification may not wait to send route reports until probe messages have been received with the routers address listed. Reports SHOULD be sent to these neighbors without first requiring a received probe with the routers address in it as well as reports from these neighbors SHOULD be accepted. Although, this allows one-way neighbor relationships to occur, it does maintain backward compatibility.

It may be necessary to form neighbor relationships based solely on Route Report messages. Neighbor time-out values may need to be configured to a value greater than the Route Report Interval for these neighbors.

Implementations that do not monitor Generation ID changes can create more noticeable black holes when using long prune lifetimes such as ~2 hours. This happens when a long prune is sent upstream and then the router that sent the long prune restarts. If the upstream router ignores the new Generation ID, the prune received by the upstream router will not be flushed and the downstream router will have no knowledge of the upstream prune. For this reason, prunes sent upstream to routers that are known to ignore Generation ID changes should have short lifetimes.

If the router must run IGMP version 1 on an interface for backwards

compatibility, DVMRP must elect the DVMRP router with the highest IP address as the IGMP querier.

Some implementations of tools that send DVMRP Ask Neighbors2 requests and receive Neighbors2 response messages require a neighbor address of 0.0.0.0 when no neighbors are listed in the response packet. (Mrinfo)

When DVMRP protocol packets are sent to tunnel endpoints, some implementations do not accept packets addressed to the All-DVMRP-Routers address and then encapsulated with the tunnel endpoint address. Mrouted versions 3.9beta2 and earlier are known to have this problem.

13. Intellectual Property Rights Notice

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

14. Full Copyright Statement

Table of Contents

Attachment Converted: "c:\program
files\qualcomm\eudora\imap\dominant\ids\attach\Untitled"