Network Working Group
INTERNET DRAFT

Dino Farinacci
Procket Networks
Yakov Rekhter
David Meyer
Cisco Systems
Peter Lothberg
Sprint
Hank Kilmer
Jeremy Hall
UUnet

Category

Standards Track
February, 2000

Multicast Source Discovery Protocol (MSDP)
<draft-ietf-msdp-spec-05.txt>

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

2. Abstract

The Multicast Source Discovery Protocol, MSDP, describes a mechanism to connect multiple PIM-SM domains together. Each PIM-SM domain uses its own independent RP(s) and does not have to depend on RPs in other domains.

3. Copyright Notice

4. Introduction

The Multicast Source Discovery Protocol, MSDP, describes a mechanism to connect multiple PIM-SM domains together. Each PIM-SM domain uses its own independent RP(s) and does not have to depend on RPs in other domains. Advantages of this approach include:

- o No Third-party resource dependencies on RP

  PIM-SM domains can rely on their own RPs only.

- o Receiver only Domains

  Domains with only receivers get data without globally advertising group membership.

- o Global Source State

  Global source state is not required, since a router need not cache Source Active (SA) messages (see below). MSDP is a periodic protocol.

The keywords MUST, MUST NOT, MAY, OPTIONAL, REQUIRED, RECOMMENDED, SHALL, SHALL NOT, SHOULD, SHOULD NOT are to be interpreted as defined in RFC 2119 [RFC2119].

5. Overview

MSDP-speaking routers in a PIM-SM [RFC2362] domain will have a MSDP peering relationship with MSDP peers in another domain. The peering relationship will be made up of a TCP connection in which control information is exchanged. Each domain will have one or more connections to this virtual topology.

The purpose of this topology is to allow domains discover multicast sources from other domains. If the multicast sources are of interest to a domain which has receivers, the normal source-tree building mechanism in PIM-SM will be used to deliver multicast data over an inter-domain distribution tree.

We envision this virtual topology will essentially be congruent to the existing BGP topology used in the unicast-based Internet today. That is, the TCP connections between MSDP peers are likely to be congruent to the connections in the BGP routing system.

6. Procedure

A source in a PIM-SM domain originates traffic to a multicast group. The PIM DR which is directly connected to the source sends the data encapsulated in a PIM Register message to the RP in the domain.

The RP will construct a "Source-Active" (SA) message and send it to its MSDP peers. The SA message contains the following fields:

- Source address of the data source.
- Group address the data source sends to.
- IP address of the RP.

Each MSDP peer receives and forwards the message away from the RP address in a "peer-RPF flooding" fashion. The notion of peer-RPF

flooding is with respect to forwarding SA messages. The BGP routing table is examined to determine which peer is the NEXT_HOP towards the originating RP of the SA message. Such a peer is called an "RPF peer". See section 14 below for the details of peer-RPF forwarding.

If the MSDP peer receives the SA from a non-RPF peer towards the originating RP, it will drop the message. Otherwise, it forwards the message to all its MSDP peers (except the one from which it received the SA message).

The flooding can be further constrained to children of the peer by interrogating BGP reachability information. That is, if a BGP peer

advertises a route (back to you) and you are the next to last AS in the AS_PATH, the peer is using you as the NEXT_HOP. This is known in other circles as Split-Horizon with Poison Reverse. An implementation SHOULD NOT forward SA messages (which were originated from the RP address covered by a route) to peers which have not Poison Reversed that route.

When an MSDP peer which is also an RP for its own domain receives a new SA message, it determines if it has any group members interested in the group which the SA message describes. That is, the RP checks for a (*,G) entry with a non-empty outgoing interface list; this implies that the domain is interested in the group. In this case, the RP triggers a (S,G) join event towards the data source as if a Join/Prune message was received addressed to the RP itself. This sets up a branch of the source-tree to this domain. Subsequent data packets arrive at the RP which are forwarded down the shared-tree inside the domain. If leaf routers choose to join the source-tree they have the option to do so according to existing PIM-SM conventions. Finally, if an RP in a domain receives a PIM Join message for a new group G, and it is caching SAs, then the RP should trigger a (S,G) join event for each SA for that group in its cache.

This procedure has been affectionately named flood-and-join because if any RP is not interested in the group, they can ignore the SA message. Otherwise, they join a distribution tree.

7. Controlling State

While RPs which receive SA messages are not required to keep MSDP (S,G) state, an RP SHOULD cache SA messages by default. One of the main advantages of caching is that since the RP has MSDP (S,G) state, join latency is greatly reduced for new receivers of G. In addition, caching greatly aids in diagnosis and debugging of various problems.

8. Timers

The main timers for MSDP are: SA-Advertisement-Timer, SA-Hold-Down-Timer, SA Cache Entry timer, KeepAlive timer, and ConnectRetry and Peer Hold Timer. Each is considered below.

8.1. SA-Advertisement-Timer

RPs which originate SA messages do it periodically as long as there is data being sent by the source. There is one SA-Advertisement-Timer covering the sources that an RP may advertise. [SA-Advertisement-Period] MUST be 60 seconds. An RP MUST not send more than one periodic SA message for a given (S,G) within an SA Advertisement interval. Originating periodic SA messages is important so that new receivers who join after a source has been active can get data quickly via the receiver's own RP when it is not caching SA state. Finally, an originating RP SHOULD trigger the transmission of an SA message as soon as it receives data from an internal source for the first time.

8.2. SA-Advertisement-Timer Processing

An RP MUST spread the generation of periodic SA messages over its reporting interval (i.e. SA-Advertisement-Period). An RP starts the SA-Advertisement-Timer when the MSDP process is configured. When the timer expires, an RP resets the timer to [SA-Advertisement-Period]

seconds, and begins the advertisement of its active sources. Active sources are advertised in the following manner: An RP packs its active sources into an SA message until the largest MSDP packet that can be sent is built or there are no more sources, and then sends the message. This process is repeated periodically within the SA-Advertisement-Period in such a way that all of the RP's sources are advertised. Note that the largest MSDP packet that can be sent has size that is the minimum of MTU of outgoing link minus size of TCP and IP headers, and 1400 (largest MSDP packet). Finally, the timer is deleted when the MSDP process is deconfigured. Note that a caching implementation may also wish to check the SA-Cache on this timer event.

8.3. SA Cache Timeout (SA-State-Timer)

Each entry in an SA Cache has an associated SA-State-Timer. A (S,G)-SA-State-Timer is started when an (S,G)-SA message is initially received by a caching MSDP peer. The timer is reset to [SA-State-Period] if another (S,G)-SA message is received before the (S,G)-SA-State-Timer expires. [SA-State-Period] MUST NOT be less than 90 seconds.

8.4. SA-Hold-Down-Timer

A caching MSDP peer SHOULD NOT forward an SA message it has received in during the previous [SA-Hold-Down-Period] seconds. [SA-Hold-Down-Period] SHOULD be set to 30 seconds. The per-SA message timer is set to [SA-Hold-Down-Period] when forwarding an (S,G)-SA message, and a (S,G)-SA message MUST only be forwarded when it's associated timer is not running. Finally, the timer is deleted when the (S,G)-SA cache entry is deleted.

8.5. KeepAlive Timer

The KeepAlive timer is used by the active connect side of the MSDP connection to track the state of the passive-connect side of the

connection. In particular, the KeepAlive timer is used to reset the TCP connection when the passive-connect side of the connection goes down. The KeepAlive timer is set to [KeepAlive-Period] when the passive-connect peer comes up. [KeepAlive-Period] SHOULD NOT be less that 75 seconds. The timer is reset to [KeepAlive-Period] upon receipt of an MSDP message from peer, and deleted when the timer expires or the passive-connect peer closes the connection.

8.6. ConnectRetry Timer

The ConnectRetry timer is used by an MSDP peer to transition from INACTIVE to CONNECTING states. There is one timer per peer, and the [ConnectRetry-Period] SHOULD be set to 30 seconds. The timer is initialized to [ConnectRetry-Period] when an MSDP peer's active connect attempt fails. When the timer expires, the peer retries the connection and the timer is reset to [ConnectRetry-Period]. It is deleted if either the connection transitions into ESTABLISHED state or the peer is deconfigured.

8.7. Peer Hold Timer

If a system does not receive successive KeepAlive messages (or any SA message) within the period specified by the Hold Timer, then a Notification message with Hold Timer Expired Error Code MUST be sent and the MSDP connection MUST be closed. [Hold-Time-Period] MUST be at least three seconds. A suggested value for [Hold-Time-Period] is 90 seconds.

The Hold Timer is initialized to [Hold-Time-Period] when the peer's transport connection is established, and is reset to [Hold-Time-

Period] when any MSDP message is received.

9. Intermediate MSDP Peers

Intermediate RPs do not originate periodic SA messages on behalf of sources in other domains. In general, an RP MUST only originate an SA for a source which would register to it.

10. SA Filtering and Policy

As the number of (S,G) pairs increases in the Internet, an RP may want to filter which sources it describes in SA messages. Also, filtering may be used as a matter of policy which at the same time can reduce state. Only the RP co-located in the same domain as the source can restrict SA messages. Note, however, that MSDP peers in transit domains should not filter SA messages or the flood-and-join model can not guarantee that sources will be known throughout the Internet (i.e., SA filtering by transit domains can cause undesired lack of connectivity). In general, policy should be expressed using MBGP [RFC2283]. This will cause MSDP messages will flow in the desired direction and peer-RPF fail otherwise. An exception occurs at an administrative scope [RFC2365] boundary. In particular, a SA message for a (S,G) MUST NOT be sent to peers which are on the other side of an administrative scope boundary for G.

11. SA Requests

If an MSDP peer decides to cache SA state, it MAY accept SA-Requests from other MSDP peers. When an MSDP peer receives an SA-Request for a group range, it will respond to the peer with a set of SA entries, in an SA-Response message, for all active sources sending to the group range requested in the SA-Request message. The peer that sends the request will not flood the responding SA-Response message to other peers. See section 17 for discussion of error handling relating to SA requests and responses.

12. Encapsulated Data Packets

For bursty sources, the RP may encapsulate multicast data from the source. An interested RP may decapsulate the packet, which SHOULD be forwarded as if a PIM register encapsulated packet was received. That is, if packets are already arriving over the interface toward the source, then the packet is dropped. Otherwise, if the outgoing interface list is non-null, the packet is forwarded appropriately. Note that when doing data encapsulation, an implementation MUST bound the time during which packets are encapsulated.

This allows for small bursts to be received before the multicast tree is built back toward the source's domain. For example, an implementation SHOULD encapsulate at least the first packet to provide service to bursty sources.

13. Other Scenarios

MSDP is not limited to deployment across different routing domains. It can be used within a routing domain when it is desired to deploy multiple RPs for the same group ranges. As long as all RPs have a interconnected MSDP topology, each can learn about active sources as well as RPs in other domains.

14. MSDP Peer-RPF Forwarding

The MSDP Peer-RPF Forwarding rules are used for forwarding SA messages throughout an MSDP enabled internet. Unlike the RPF check used when forwarding data packets, the Peer-RPF check is against the RP address carried in the SA message.

14.1. Peer-RPF Forwarding Rules

An SA message originated by an MSDP originator R and received by a MSDP router from MSDP peer N is accepted if N is the appropriate RPF neighbor for originator R (the RP in the SA message), and discarded otherwise.

The RPF neighbor is chosen using the first of the following rules that matches:

(i). R is the RPF neighbor if we have an MSDP peering with R.

(ii). The external MBGP neighbor towards which we are

poison-reversing the MBGP route towards R is the RPF neighbor if we have an MSDP peering with it.

(iii). If we have any MSDP peerings with neighbors in the first AS along the AS_PATH (the AS from which we learned this route), but no external MBGP peerings with them, the neighbor with the highest IP address is the RPF neighbor.

(vi). The internal MBGP advertiser of the router towards R is the RPF neighbor if we have an MSDP peering with it.

(v). If none of the above match, and we have an MSDP default-peer configured, the MSDP default-peer is the RPF neighbor.

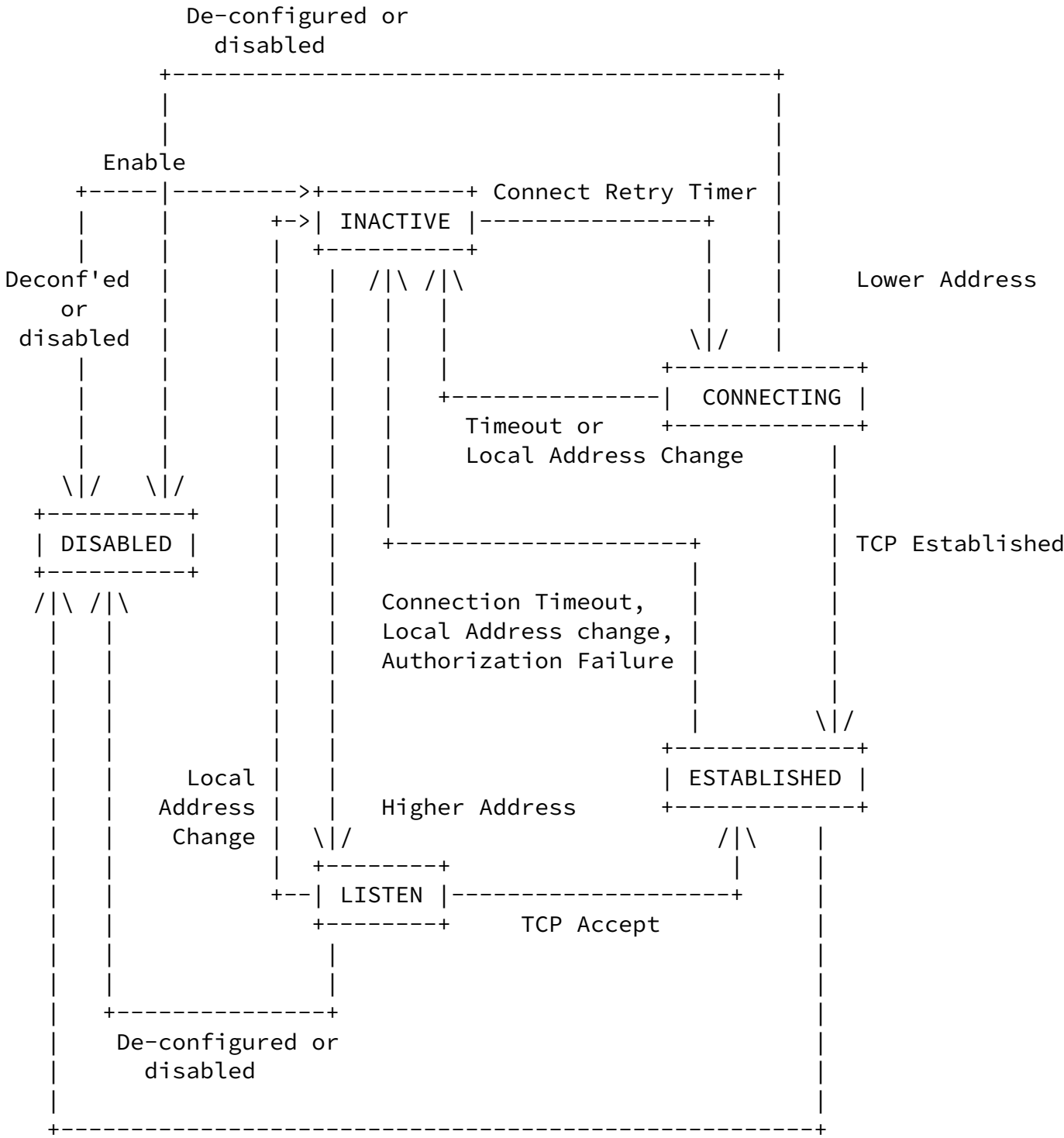
14.2. MSDP default-peer semantics

An MSDP default-peer is much like a default route. It is intended to be used in those cases where a stub network isn't running BGP or MBGP. An MSDP peer configured with a default-peer accepts all SA messages from the default-peer. Note that a router running BGP or MBGP SHOULD NOT allow configuration of default peers, since this allows the possibility for SA looping or black-holes to occur.

15. MSDP Connection Establishment

MSDP messages will be encapsulated in a TCP connection. An MSDP peer listens for new TCP connections on port 639. One side of the MSDP peering relationship will listen on the well-known port and the other side will do an active connect to the well-known port. The side with the higher peer IP address will do the listen. This connection establishment algorithm avoids call collision. Therefore, there is no need for a call collision procedure. It should be noted, however, that the disadvantage of this approach is that it may result in longer startup times at the passive end.

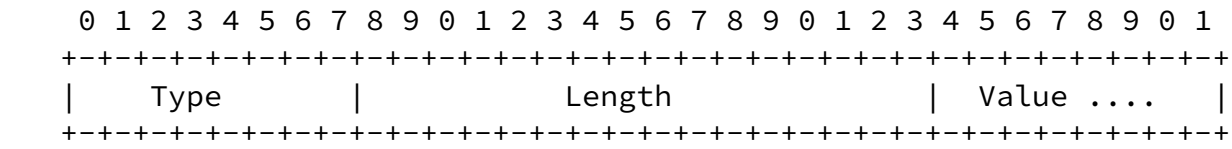
An MSDP peer starts in the INACTIVE state. MSDP peers establish peering sessions according to the following state machine:


De-configured or
disabled
Enable
Connect Retry Timer
INACTIVE
Deconf'ed
or
disabled
Lower Address
CONNECTING
Timeout or
Local Address Change
DISABLED
TCP Established
Connection Timeout,
Local Address change,
Authorization Failure
ESTABLISHED
Local
Address
Change
Higher Address
LISTEN
TCP Accept
De-configured or
disabled

16. Packet Formats

MSDP messages will be encoded in TLV format. If an implementation receives a TLV that has length that is longer than expected, the TLV SHOULD be accepted. Any additional data SHOULD be ignored.

16.1. MSDP TLV format:



Type (8 bits)
Describes the format of the Value field.

Length (16 bits)
Length of Type, Length, and Value fields in octets.
minimum length required is 4 octets, except for
Keepalive messages.

Value (variable length)
Format is based on the Type value. See below. The length of the value field is Length field minus 3. All reserved fields in the Value field MUST be transmitted as zeros and ignored on receipt.

16.2. Defined TLVs

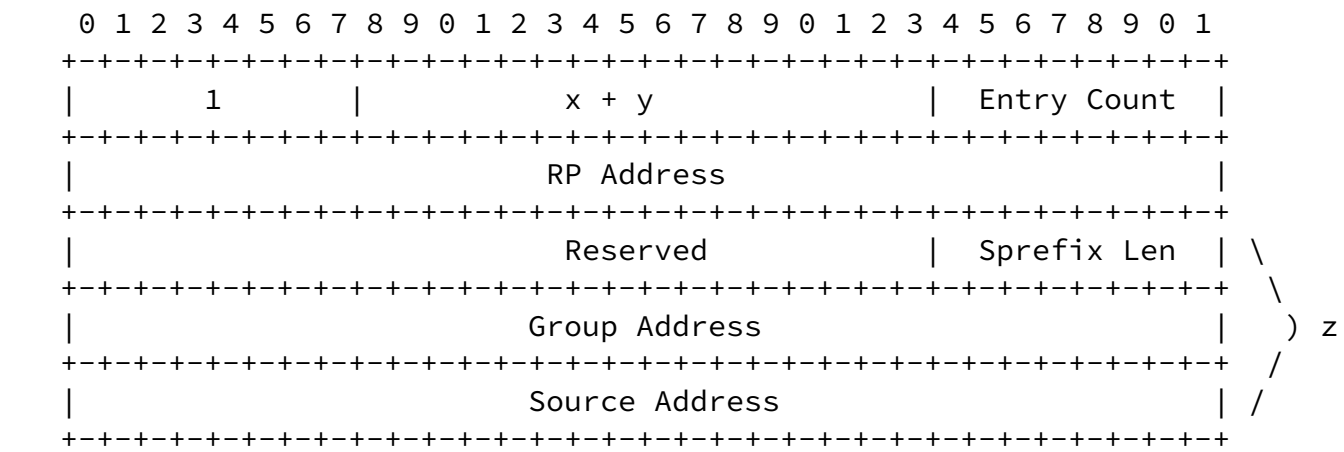
The following TLV Types are defined:

Code	Type
1	IPv4 Source-Active
2	IPv4 Source-Active Request
3	IPv4 Source-Active Response
4	KeepAlive
5	Notification

Each TLV is described below.

16.2.1. IPv4 Source-Active TLV

The maximum size SA message that can be sent is 1400 octets. If an MSDP peer needs to originate a message with information greater than 1400 octets, it sends successive 1400 octet or smaller messages. The 1400 octet size does not include the TCP, IP, layer-2 headers.



Type
IPv4 Source-Active TLV is type 1.

Length x
Is the length of the control information in the message. x is

8 octets (for the first two 32-bit quantities) plus 12 times Entry Count octets.

Length y
If 0, then there is no data encapsulated. Otherwise an IPv4 packet follows and y is the length of the total length field of the IPv4 header encapsulated. If there are multiple SA TLVs in a message, and data is also included, y must be 0 in all SA TLVs except the last one and the last SA TLV must reflect the source and destination addresses in the IP header of the encapsulated data.

Entry Count
Is the count of z entries (note above) which follow the RP address field. This is so multiple (S,G)s from the same domain can be encoded efficiently for the same RP address.

RP Address
The address of the RP in the domain the source has become active in.

Reserved
The Reserved field MUST be transmitted as zeros and ignored

by a receiver.

Sprefix Len
The route prefix length associated with source address. This field MUST be transmitted as 32 (/32). An Invalid Sprefix Len Notification SHOULD be sent upon receipt of any other value.

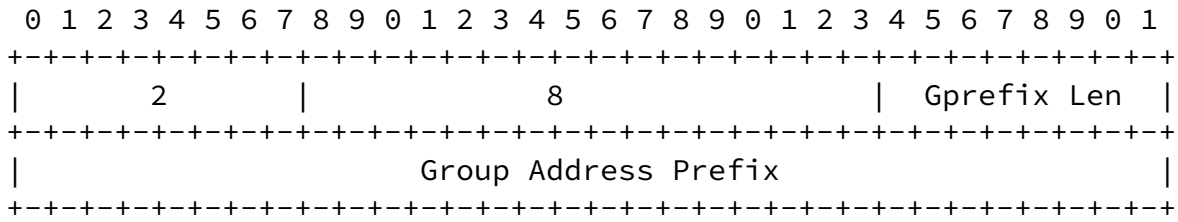
Group Address
The group address the active source has sent data to.

Source Address
The IP address of the active source.

Multiple SA TLVs MAY appear in the same message and can be batched for efficiency at the expense of data latency. This would typically occur on intermediate forwarding of SA messages.

16.2.2. IPv4 Source-Active Request TLV

The Source-Active Request is used to request SA-state from a caching MSDP peer. If an RP in a domain receives a PIM Join message for a group, creates (*,G) state and wants to know all active sources for group G, and it has been configured to peer with an SA-state caching peer, it may send an SA-Request message for the group.



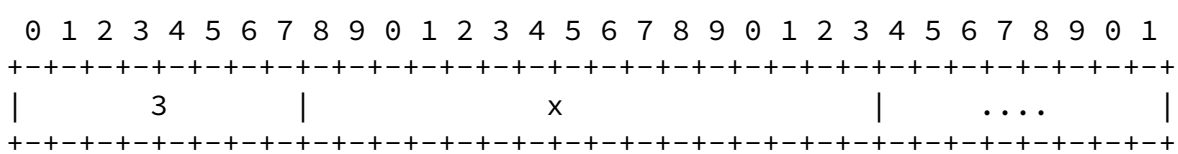
Type
IPv4 Source-Active Request TLV is type 2.

Gprefix Len
The route prefix length associated with the group address prefix.

Group Address
The group address the MSDP peer is requesting.

16.2.3. IPv4 Source-Active Response TLV

The Source-Active Response is sent in response to a Source-Active Request message. The Source-Active Response message has the same format as a Source-Active message but does not allow encapsulation of multicast data.

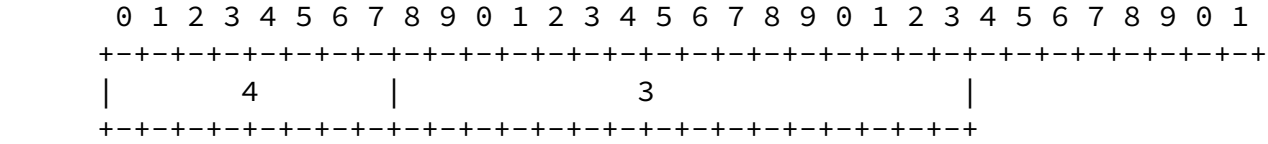


Type
IPv4 Source-Active Response TLV is type 3.

Length x
Is the length of the control information in the message. x is 8 octets (for the first two 32-bit quantities) plus 12 times Entry Count octets.

16.2.4. KeepAlive TLV

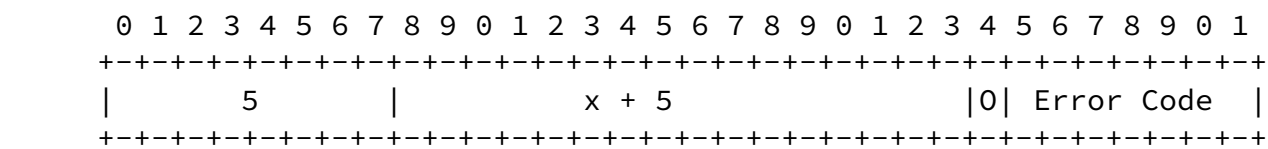
A KeepAlive TLV is sent to an MSDP peer if and only if there were no MSDP messages sent to the peer after a period of time. This message is necessary for the active connect side of the MSDP connection. The passive connect side of the connection knows that the connection will be reestablished when a TCP SYN packet is sent from the active connect side. However, the active connect side will not know when the passive connect side goes down. Therefore, the KeepAlive timeout will be used to reset the TCP connection.

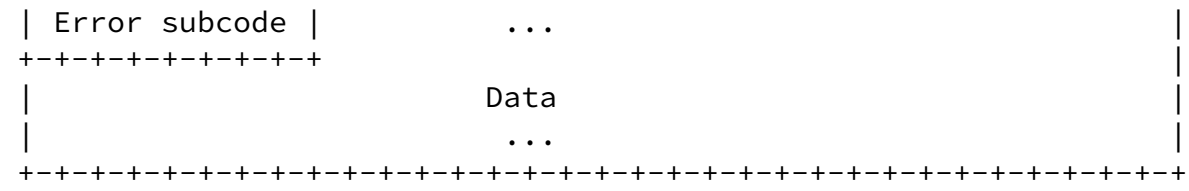


The length of the message is 3 octets which encompasses the one octet Type field and the two octet Length field.

16.2.5. Notification TLV

A Notification message is sent when an error condition is detected, and has the following form:





Type
The Notification TLV is type 5.

Length
Length is a two octet field with value x + 5, where x is the length of the notification data field.

O-bit
Open-bit. If clear, the connection will be closed.

Error code
This 7-bit unsigned integer indicates the type of Notification. The following Error Codes have been defined:

Error Code	Symbolic Name	Reference
1	Message Header Error	Section 17.1
2	SA-Request Error	Section 17.2
3	SA-Message/SA-Response Error	Section 17.3
4	Hold Timer Expired	Section 17.4
5	Finite State Machine Error	Section 17.5
6	Notification	Section 17.6
7	Cease	Section 17.7

Error subcode:
This one-octet unsigned integer provides more specific information about the reported error. Each Error Code may have one or more Error Subcodes associated with it. If no appropriate Error Subcode is defined, then a zero (Unspecific) value is used for the Error Subcode field, and the O-bit must be cleared (i.e. the connection will be closed). The used notation in the error description below is: MC = Must Close connection = O-bit clear; CC = Can Close connection = O-bit might be cleared.

Message Header Error subcodes:

0 - Unspecific	(MC)
2 - Bad Message Length	(MC)

3 - Bad Message Type	(CC)

SA-Request Error subcodes:

0 - Unspecific	(MC)
1 - Does not cache SA	(MC)
2 - Invalid Group	(MC)

SA-Message/SA-Response Error subcodes

0 - Unspecific	(MC)
1 - Invalid Entry Count	(CC)
2 - Invalid RP Address	(MC)
3 - Invalid Group Address	(MC)
4 - Invalid Source Address	(MC)
5 - Invalid Sprefix Length	(MC)
6 - Looping SA (Self is RP)	(MC)
7 - Unknown Encapsulation	(MC)
8 - Administrative Scope Boundary Violated	(MC)

Hold Timer Expired subcodes (the O-bit is always clear):

0 - Unspecific	(MC)

Finite State Machine Error subcodes:

0 - Unspecific	(MC)
1 - Unexpected Message Type FSM Error	(MC)

Notification subcodes (the O-bit is always clear):

0 - Unspecific	(MC)

Cease subcodes (the O-bit is always clear):

0 - Unspecific	(MC)

17. MSDP Error Handling

This section describes actions to be taken when errors are detected while processing MSDP messages. MSDP Error Handling is similar to that of BGP [RFC1771].

When any of the conditions described here are detected, a Notification message with the indicated Error Code, Error Subcode, and Data fields is sent. In addition, the MSDP connection might be closed. If no Error Subcode is specified, then a zero (Unspecific) must be used.

The phrase "the MSDP connection is closed" means that the transport protocol connection has been closed and that all resources for that MSDP connection have been deallocated.

17.1. Message Header Error Handling

All errors detected while processing the Message Header are indicated by sending the Notification message with Error Code Message Header Error. The Error Subcode describes the specific nature of the error. The Data field contains the erroneous Message (including the message header).

If the Length field of the message header is less than 4 or greater than 1400, or the length of a KeepAlive message is not equal to 3, then the Error Subcode is set to Bad Message Length.

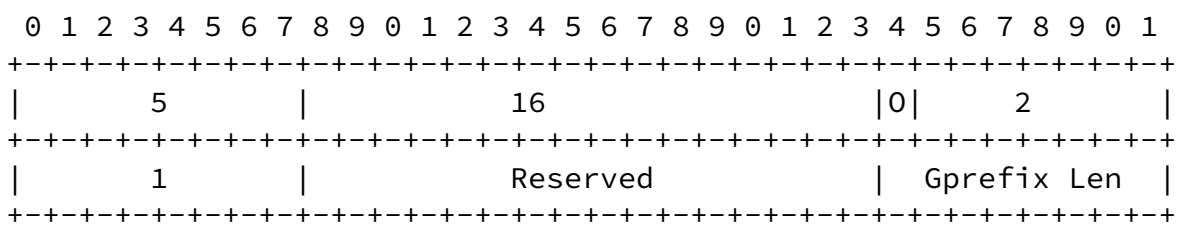
If the Type field of the message header is not recognized, then the Error Subcode is set to Bad Message Type.

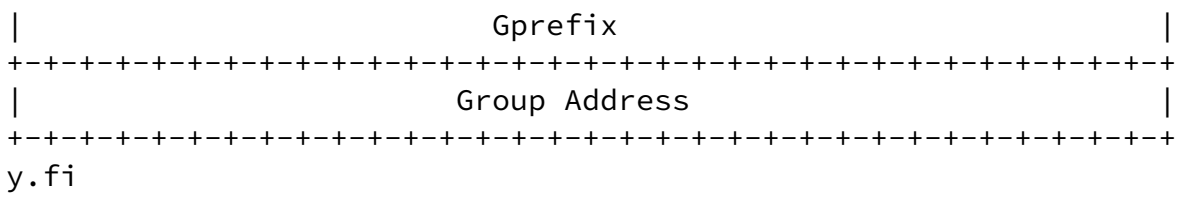
17.2. SA-Request Error Handling

The SA-Request Error code is used to signal the receipt of a SA request at a non-caching MSDP peer, or at a caching MSDP peer when an invalid group address requested.

When a non-caching MSDP peer receives an SA-Request, it returns the

following notification:





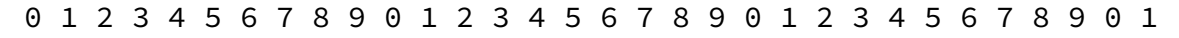
y.fi

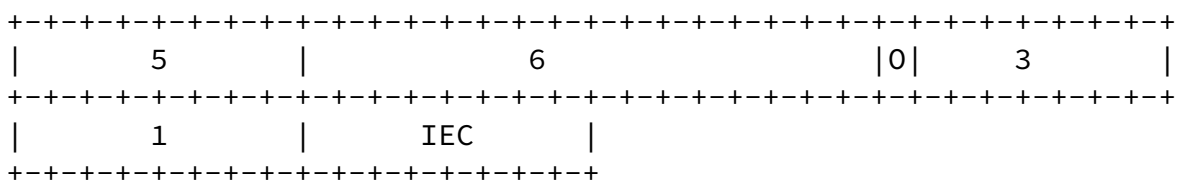
If a caching MSDP peer receives a request for an invalid group, it returns the following notification:

17.3. SA-Message/SA-Response Error Handling

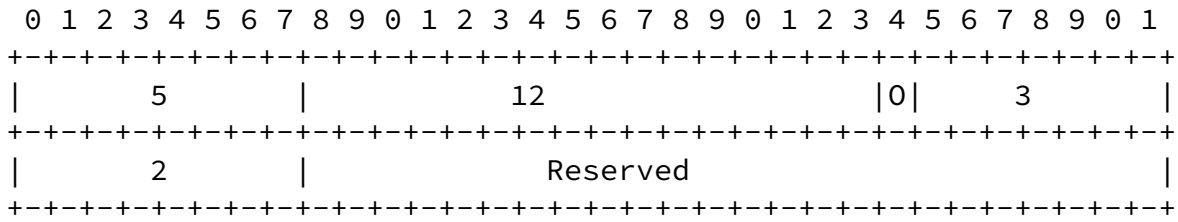
The SA-Message/SA-Response Error code is used to signal the receipt of a erroneous SA Message at an MSDP peer, or the receipt of an SA-Response Message by a peer that did not issue a SA-Request. It has the following form:

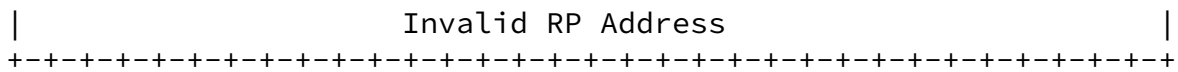
17.3.1. Invalid Entry Count (IEC)



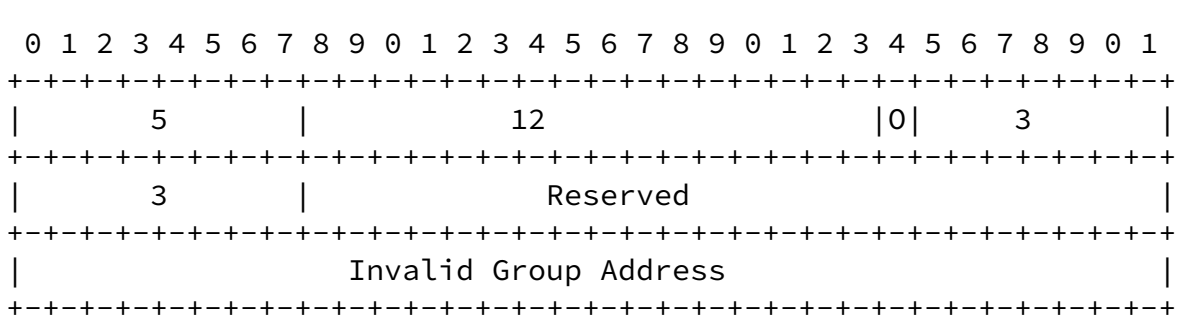


17.3.2. Invalid RP Address

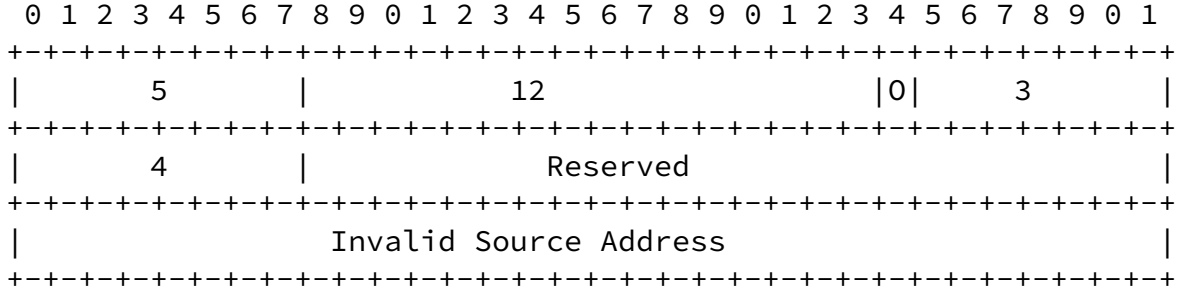




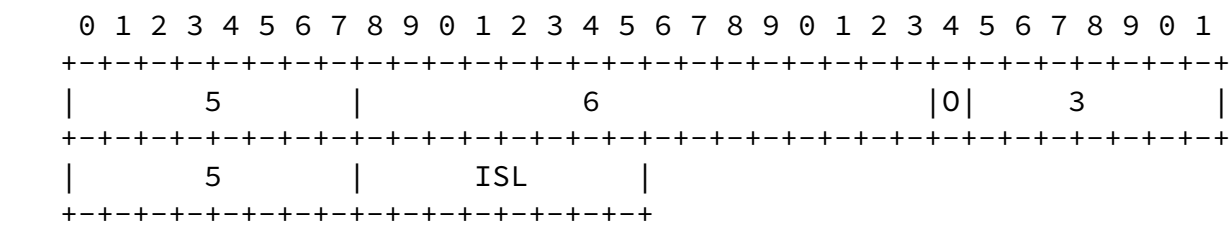
17.3.3. Invalid Group Address



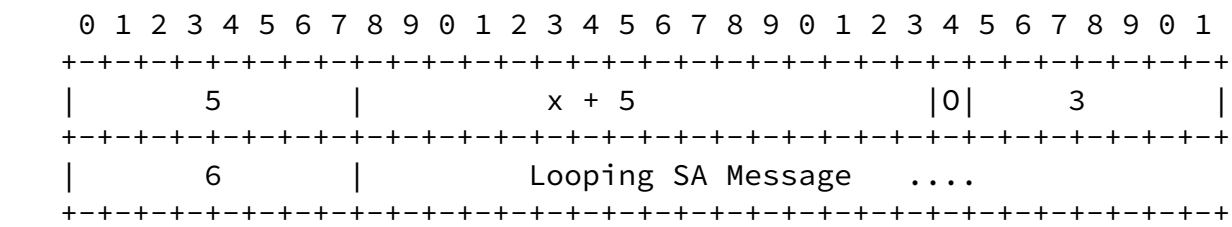
17.3.4. Invalid Source Address



17.3.5. Invalid Sprefix Length (ISL)



17.3.6. Looping SAs (Self is RP in received SA)

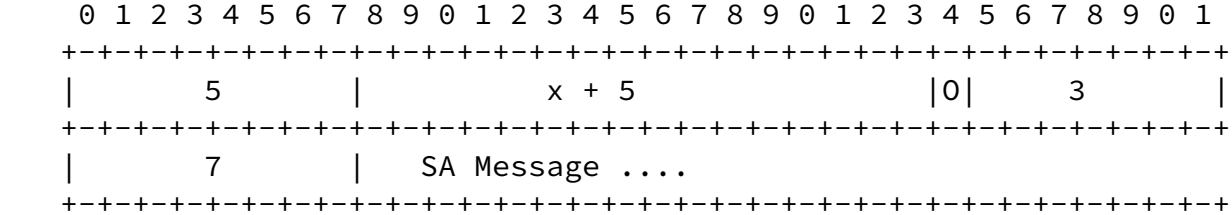


Length x

x is the length of the looping SA message contained in the data field of the Notification message.

17.3.7. Unknown Encapsulation

This notification is sent on receipt of SA data that is encapsulated in an unknown encapsulation type. See section 18 for known encapsulations.

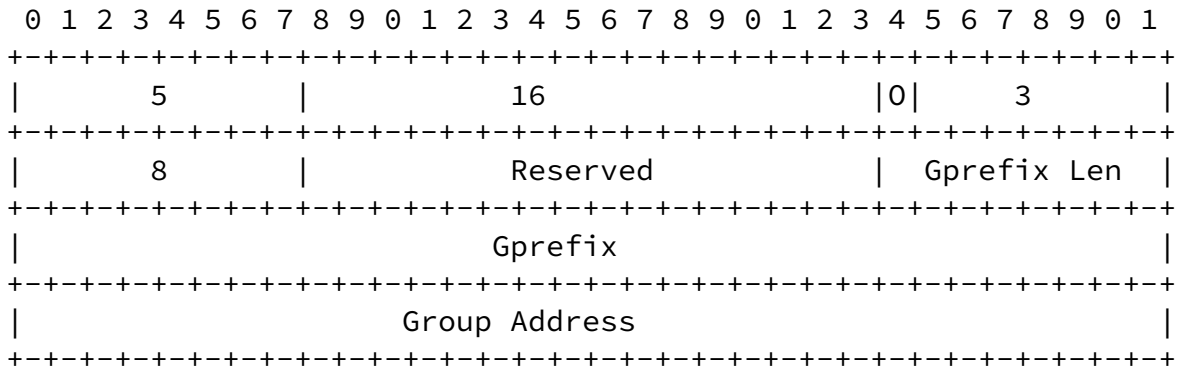


Length x
x is the length of the SA message (which contained data which

was encapsulated in some unknown way) that is contained in the data field of the Notification message.

17.3.8. Administrative Scope Boundary Violated

This notification is used when an SA message is received for a group G from a peer which is across an administrative scope boundary for G.



17.4. Hold Time Expired

If a system does not receive successive KeepAlive or any SA Message and/or Notification messages within the period specified in the Hold Timer, the notification message with Hold Timer Expired Error Code and no additional data MUST be sent and the MSDP connection closed.

17.5. Finite State Machine Error Handling

Any error detected by the MSDP Finite State Machine (e.g., receipt of an unexpected event) is indicated by sending the Notification message with Error Code Finite State Machine Error.

17.6. Notification Message Error Handling

If a node sends a Notification message, and there is an error in that message, and the O-bit of that message is not clear, a Notification

with O-bit clear, Error Code of Notification Error, and subcode Unspecific must be sent. In addition, the Data field must include the Notification message that triggered the error. However, if the erroneous Notification message had the O-bit clear, then any error, such as an unrecognized Error Code or Error Subcode, should be noticed, logged locally, and brought to the attention of the administrator of the remote node.

17.7. Cease

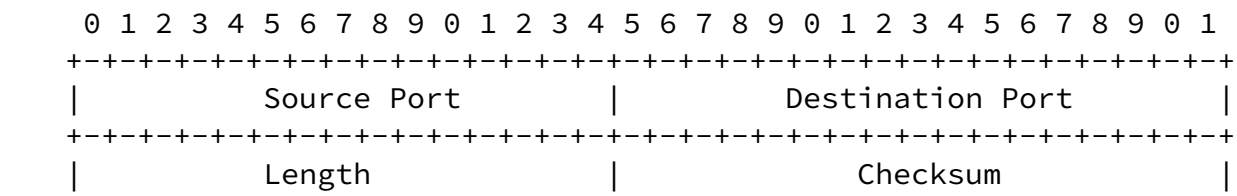
In absence of any fatal errors (that are indicated in this section), an MSDP node may choose at any given time to close its MSDP connection by sending the Notification message with Error Code Cease. However, the Cease Notification message MUST NOT be used when a fatal error indicated by this section does exist.

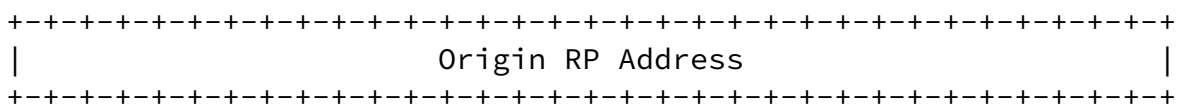
18. SA Data Encapsulation

This section describes UDP, GRE, and TCP encapsulation of SA data. Encapsulation type is a configuration option.

18.1. UDP Data Encapsulation

Data packets MAY be encapsulated in UDP. In this case, the UDP pseudo-header has the following form:





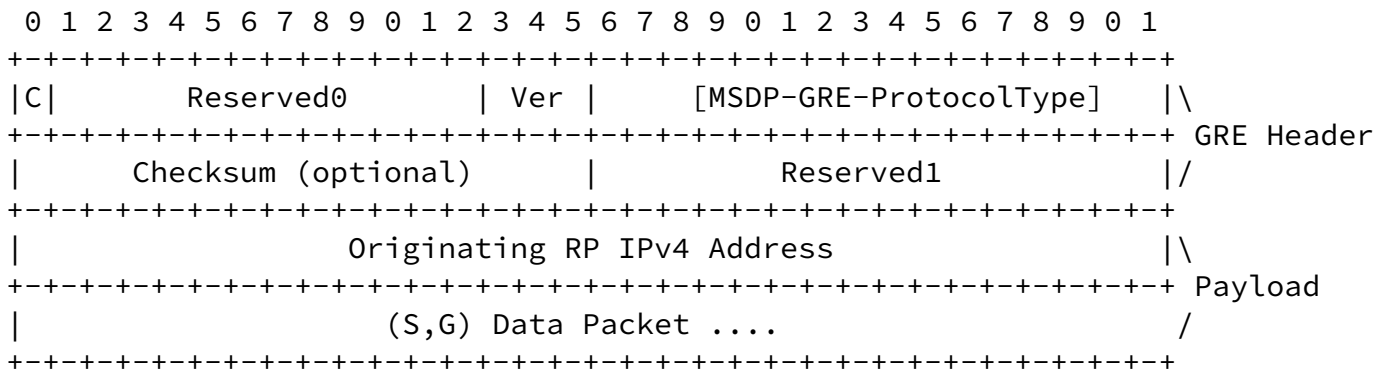
The Source port, Destination Port, Length, and Checksum are used according to RFC 768. Source and Destination ports are known via an implementation-specific method (e.g. per-peer configuration).

Checksum
The checksum is computed according to RFC 768 [RFC768].

Originating RP Address
The Originating RP Address is the address of the RP sending the encapsulated data.

18.2. GRE Encapsulation

MSDP SA-data MAY be encapsulated in GRE using protocol type [MSDP-GRE-ProtocolType]. The GRE header and payload packet have the following form:



18.2.1. Encapsulation and Path MTU Discovery [RFC1191]

Existing implementations of GRE, when using IPv4 as the Delivery Header, do not implement Path MTU discovery and do not set the Don't Fragment bit in the Delivery Header. This can cause large packets to become fragmented within the tunnel and reassembled at the tunnel exit (independent of whether the payload packet is using PMTU). If a

tunnel entry point were to use Path MTU discovery, however, that tunnel entry point would also need to relay ICMP unreachable error messages (in particular the "fragmentation needed and DF set" code) back to the originator of the packet, which is not required by the GRE specification [GRE]. Failure to properly relay Path MTU information to an originator can result in the following behavior: the originator sets the don't fragment bit, the packet gets dropped within the tunnel, but since the originator doesn't receive proper feedback, it retransmits with the same PMTU, causing subsequently transmitted packets to be dropped.

18.3. TCP Data Encapsulation

As discussed earlier, encapsulation of data in SA messages MAY be supported for backwards compatibility with legacy MSDP peers.

19. Security Considerations

An MSDP implementation MAY use IPsec [RFC1825] or keyed MD5 [RFC1828] to secure control messages. When encapsulating SA data in GRE, security should be relatively similar to security in a normal IPv4 network, as routing using GRE follows the same routing that IPv4 uses natively. Route filtering will remain unchanged. However packet filtering at a firewall requires either that a firewall look inside the GRE packet or that the filtering is done on the GRE tunnel endpoints. In those environments in which this is considered to be a security issue it may be desirable to terminate the tunnel at the firewall.

20. Acknowledgments

The authors would like to thank Bill Nickless, John Meylor, Liming Wei, Manoj Leelanivas, Mark Turner, John Zwiebel, and Cristina Radulescu-Banu for their design feedback and comments. In addition to many other contributions, Tom Pusateri helped to clarify the connection state machine, Dave Thaler helped to clarify the Notification message types, and Bill Fenner helped to clarify the Peer-RPF rules.

21. Author's Address:

Dino Farinacci
Procket Networks
3850 No. First St., Ste. C
San Jose, CA 95134
Email: dino@procket.com

Yakov Rehkter
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134
Email: yakov@cisco.com

Peter Lothberg
Sprint
VARESA0104
12502 Sunrise Valley Drive
Reston VA, 20196
Email: roll@sprint.net

Hank Kilmer
Email: hank@rem.com

Jeremy Hall
UUnet Technologies
3060 Williams Drive
Fairfax, VA 22031
Email: jhall@uu.net

David Meyer
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134
Email: dmm@cisco.com

22. REFERENCES

[GRE] Farinacci, D., et al., "Generic Routing Encapsulation (GRE)", draft-meyer-gre-update-03.txt, January, 2000. Work in Progress.

[RFC768] Postel, J. "User Datagram Protocol", RFC 768, August, 1980.

[RFC1191] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191, November 1990.

[RFC1771] Rekhter, Y., and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[RFC1825] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 1825, August, 1995.

[RFC1828] P. Metzger and W. Simpson, "IP Authentication using Keyed MD5", RFC 1828, August, 1995.

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March, 1997.

[RFC2283] Bates, T., Chandra, R., Katz, D., and Y. Rekhter., "Multiprotocol Extensions for BGP-4", RFC 2283, February 1998.

[RFC2362] Estrin D., et al., "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, June 1998.

[RFC2365] Meyer, D. "Administratively Scoped IP Multicast", RFC 2365, July, 1998.