Network Working Group
INTERNET DRAFT
Category

David Meyer (Editor)
Bill Fenner (Editor)
Standards Track
August, 2001

Multicast Source Discovery Protocol (MSDP)
<draft-ietf-msdp-spec-11.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.

Note that MSDP may be used with protocols other than PIM-SM, but such usage is not specified in this memo.

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 have a MSDP peering relationship with MSDP peers in another domain. The peering relationship is made up of a TCP connection in which control information is exchanged. Each domain has one or more connections to this virtual topology.

The purpose of this topology is to allow domains to 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

When an RP in a PIM-SM domain first learns of a new sender, e.g. via PIM register messages, it constructs a "Source-Active" (SA) message and sends it to its MSDP peers. The SA message contains the following fields:

- o Source address of the data source.
- o Group address the data source sends to.
- o 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 Multicast RPF Routing Information Base (MRIB) is examined to determine which peer towards the originating RP of the SA message is selected. Such a peer is called an "RPF peer". See section 14 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).

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, 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. Caching

A MSDP speaker MUST cache SA messages. Caching allows pacing of MSDP messages as well as reducing join latency for new receivers of a group G at an originating RP which has existing MSDP (S,G) state. 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, 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 required to keep announcements alive in caches, and so that new receivers who join after a source has been active can get data quickly via a non-caching RP. 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 since MSDP is a periodic protocol, an implemenation SHOULD send all cached SA messages when a connection is established. Finally, the timer is deleted when the MSDP process is deconfigured.

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 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

When an SA message is received which creates (S,G) state, the (S,G)-SA message will be forwarded if the peer-RPF check succeeds. If the peer-RPF check succeeds and the (S,G)-SA message is not already in the SA cache, then the (S,G)-SA-Hold-Down timer is set to [SA-Hold-Down-Period] seconds. When an (S,G)-SA message is received and an (S,G) entry already exists, the message is forwarded only if the (S,G)-SA-Hold-Down timer is not running. [SA-Hold-Down-Period] SHOULD be set to 30 seconds.

8.5. Peer Hold Timer

If a system has not received any MSDP 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. [HoldTime-Period] MUST be at least three seconds. The recommended value for [HoldTime-Period] is 90 seconds.

The Hold Timer is initialized to [HoldTime-Period] when the peer's transport connection is established, and is reset to [HoldTime-Period] when any MSDP message is received. Finally, the timer is deleted when the peer's transport connection is closed.

8.6. KeepAlive Timer

Once an MSDP transport connection is established, each side of the connection sends a KeepAlive message and sets a KeepAlive timer. If the KeepAlive timer expires, the local system sends a KeepAlive message and restarts its KeepAlive timer.

The KeepAlive timer is set to [KeepAlive-Period] when the peer comes up. The timer is reset to [KeepAlive-Period] each time an MSDP message is sent to the peer, and reset when the timer expires. Finally, the KeepAlive timer is deleted when the peer's transport

connection is closed.

[KeepAlive-Period] MUST be less than [HoldTime-Period], and MUST be at least one second. The recommended value for [KeepAlive-Period] is 75 seconds.

8.7. 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 speaker attempts to actively open a TCP connection to its peer (see section 15, event E2, action A2). 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.

9. Intermediate MSDP Peers

Intermediate MSDP speakers 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, and ONLY RPs may originate SA messages.

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 to 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

A MSDP speaker MAY accept SA-Requests from other MSDP peers. When an MSDP speaker 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 in its SA cache sending to the group 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

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. Definitions

The following definitions are used in the description of the Peer-RPF Forwarding Rules:

14.1.1. Multicast RPF Routing Information Base (MRIB)

The MRIB is the multicast topology table. It is typically derived from the unicast routing table or from other routing protocols such as multi-protocol BGP [RFC2283].

14.1.2. RPF Route

The RPF route is the route that the MRIB chooses for a given address. The RPF route for a SA's originating RP is used to select the peer from which the SA is accepted.

14.2. Peer-RPF Forwarding Rules

An SA message originated by R and received by X from N is accepted if N is the peer-RPF neighbor for X, and is discarded otherwise.

```
        MPP(R,N)                MP(N,X)
R ---------....-------> N ------------------> X
        SA(S,G,R)               SA(S,G,R)
```

Where MPP(R,N) is an MSDP peering path (zero or more MSDP peers) between R and N. SA(S,G,R) is an SA message for source S on group G originated by an RP R. MP(N,X) is an MSDP peering between N and X.

The peer-RPF neighbor is chosen deterministically, using the first of the following rules that matches. In particular, N is the RPF neighbor of X with respect to R if

(i). N == R (X has an MSDP peering with R).

(ii). N is the BGP NEXT_HOP of the active RPF route for R.

(iii). The active RPF route for R is learned through a distance-vector or path-vector routing protocol (e.g. BGP, RIP, DVMRP) and N is the neighbor that advertised the active RPF route for R.

(iv). N resides in an AS that is in the AS_PATH of the active RPF route for R, and N has the highest IP address among the MSDP peers that reside in ASs in that AS_PATH.

(v). N is configured as the static RPF-peer for R.

14.3. MSDP static RPF-peer semantics

If none of the rules (i) - (iv) are able to determine an RPF peer for R, a longest-match lookup is performed in the static RPF peer table. This table MUST be able to contain a default entry, and SHOULD be able to contain prefix or per-host (RP) entries. This table statically maps RP addresses to peers, and allows configuration of topology that is e.g. unknown to the MRIB.

The result of the longest-match lookup of an RP address R in the static RPF peer table is an MSDP peer, which is the RPF neighbor for R.

14.4. MSDP mesh-group semantics

A MSDP mesh-group is a operational mechanism for reducing SA flooding, typically in an intra-domain setting. In particular, when some subset of a domain's MSDP speakers are fully meshed, then can be configured into a mesh-group.

Note that mesh-groups assume that a member doesn't have to forward an SA to other members of the mesh-group because the originator will forward to all members. To be able for the originator to forward to all members (and to have each member also be a potential originator), the mesh-group must be a full mesh of MSDP peering among all members.

The semantics of the mesh-group are as follows:

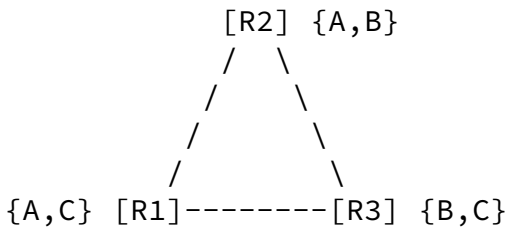
(i). If a member R of a mesh-group M receives a SA message from an MSDP peer that is also a member of mesh-group M, R accepts the SA message and forwards it to all of its peers that are not part of any mesh-group. R MUST NOT forward the SA message to other members of mesh-group M.

(ii). If a member R of a mesh-group M receives a SA message from an MSDP peer that is not a member of mesh-group M, and the SA message passes the peer-RPF check, then R forwards the SA message to all members of mesh-group M.

(iii). Cross mesh-group forwarding

If a member R of a mesh-groups M and N receives an SA message from an MSDP peer in mesh-group M, R forwards the SA to its MSDP peers in mesh-group N if it receives that SA message from a peer that is in the same mesh-group as its peer-RPF neighbor for that SA.

For example, consider the case in which three routers (R1, R2, and R3) and three mesh-groups (A, B, and C) are arranged in a triangle, e.g.,



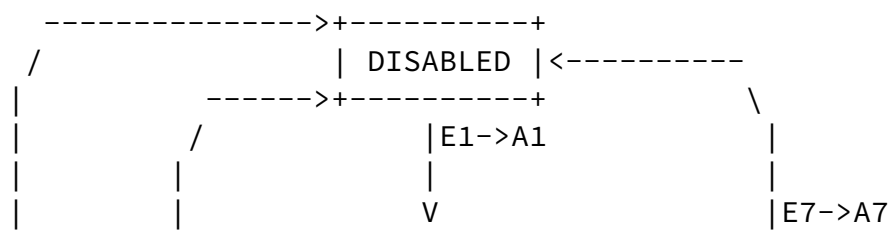
Now, when R1 receives an SA message from R2 and R1's peer-RPF neighbor for this SA lies in mesh-group A, R1 forwards the SA message its peers in other mesh-groups (in particular, R3 in mesh-group C). Similarly, if R3's peer-RPF neighbor lies in mesh-group B, R3 will forward an SA message from R2. In this case, both R1 and R3 will send SA messages to each other (because they share common mesh-group C), but neither of them will forward any further the SA messages received from each other (as their peer-RPF neighbors do not lie in mesh-group C).

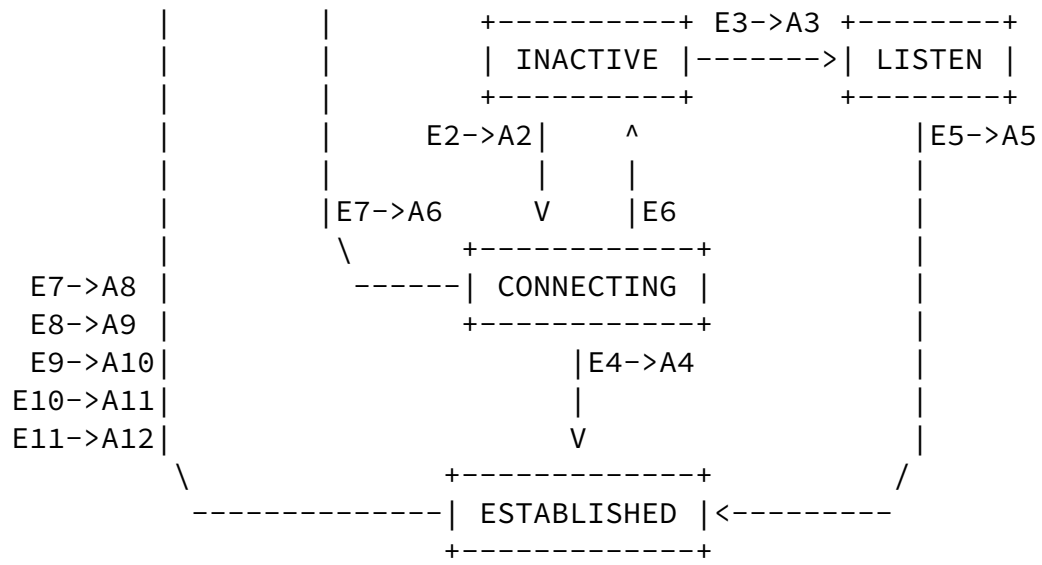
Note that since mesh-groups suspend peer-RPF checking of SAs received from a mesh-group member ((i). above), they allow for mis-configuration to cause SA looping.

15. MSDP Connection State Machine

MSDP uses TCP as its transport protocol. In a peering relationship, one MSDP peer listens for new TCP connections on the well-known port 639. The other side makes an active connect to this port. The peer with the higher IP address will 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 side.

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





15.1. Events

E1) Enable MSDP peering with P
E2) Own IP address < P's IP address
E3) Own IP address > P's IP address
E4) TCP established (active side)
E5) TCP established (passive side)
E6) ConnectRetry timer expired
E7) Disable MSDP peering with P
An example of when to do this is when one's own address is changed)
E8) Hold Timer expired
E9) Authorization failure
E10) Notification TLV received
E11) Error detected

15.2. Actions

A1) Allocate resources for peering with P
    Compare one's own and peer's IP addresses
A2) TCP active OPEN
    Set ConnectRetry timer to [ConnectRetry-Period]
A3) TCP passive OPEN (listen)
A4) Delete ConnectRetry timer
    Send KeepAlive TLV
    Set KeepAlive timer to [KeepAlive-Period]
    Set Hold Timer to [HoldTime-Period]
A5) Send KeepAlive TLV
    Set KeepAlive timer to [KeepAlive-Period]
    Set Hold Timer to [HoldTime-Period]
A6) Abort TCP active OPEN attempt
    Release resources allocated for peering with P
A7) Abort TCP passive OPEN attempt
    Release resources allocated for peering with P

In action sets 8)-12), the action "Close peering session" includes the following steps:
   Close TCP connection
   Delete KeepAlive timer
   Delete Hold Timer
   Release resources allocated for peering with P

A8) Send Notification TLV with Error Code "Cease"
    Close peering session
A9) Send Notification TLV with Error Code "Hold Timer Expired"
    Close peering session

A10) Notify management system unless this has already been done by the security mechanism
     Close peering session
A11) Notify management system
     If the received Notification TLV's O-bit was cleared, close peering session. Otherwise, remain in ESTABLISHED state.
A12) Send Notification TLV with appropriate Error Code
     Notify management system
     If the sent Notification TLV's O-bit was cleared, close peering session. Otherwise, remain in ESTABLISHED state.

15.3. Peer-specific Events

The following peer-specific events can occur in the ESTABLISHED state, they do not cause a state transition. Appropriate actions are listed for each event.

*) KeepAlive timer expired:
   -> Send KeepAlive TLV
   -> Set KeepAlive timer to [KeepAlive-Period]
*) KeepAlive TLV received:
   -> Set Hold Timer to [HoldTime-Period]
*) Source-Active TLV received:
   -> Set Hold Timer to [HoldTime-Period]
   -> Run Peer-RPF Forwarding algorithm (if caching, consider SA-Hold-Down Timer and SA-State Timer)
   -> Set KeepAlive timer to [KeepAlive-Period] for those peers the Source-Active TLV is forwarded to
   -> Send information to PIM-SM
   -> If caching, store information
*) Source-Active Request TLV received:
   -> Set Hold Timer to [HoldTime-Period]
   -> If SA-Requests are accepted, send Source-Active Response TLV and set KeepAlive timer to [KeepAlive-Period]
*) Source-Active Response TLV received:
   -> Set Hold Timer to [HoldTime-Period]
   -> If a corresponding SA-Request were previously sent, send information to PIM-SM. If not, an error has occured (event 11 above)
   -> If caching, store information

15.4. Peer-independent Events

There are also a number of events that affect more than one peering session, but still require actions to be performed on a per-peer

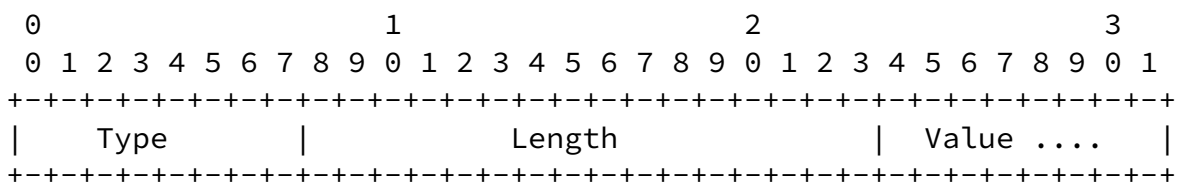
basis. If the MSDP speaker does not cache SA messages, ignore all events and actions pertaining to caching.

*) SA-Advertisement-Timer expired:
   -> Start periodic transmission of Source-Active TLV(s)
   -> Set KeepAlive timer to [KeepAlive-Period] each time a Source-Active TLV is sent
*) MSDP learns of a new active internal source (e.g. PIM-SM register received for a new source):
   -> Send Source-Active TLV
   -> Set KeepAlive timer to [KeepAlive-Period]
*) Source-Active Request triggered (event not specified here):
   -> Send Source-Active Request TLV
   -> Set KeepAlive timer to [KeepAlive-Period]
*) SA-State-Timer expired (one timer per cache entry):
   -> Implementation specific, typically mark the cache entry for deletion

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. The maximum TLV length is 1400.

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.

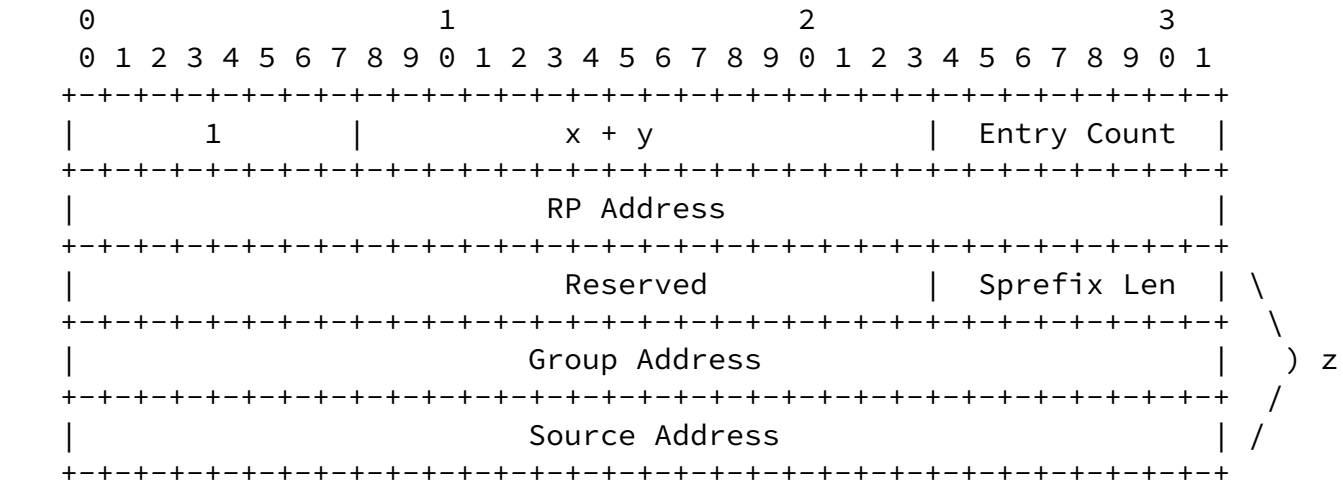
In addition, the following TLV Types are assigned but not described

in this memo:

Code	Type
6	MSDP traceroute in progress
7	MSDP traceroute reply

16.2.1. IPv4 Source-Active TLV

The maximum size SA message that can be sent is 9192 octets. The 9192 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 MUST be 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.

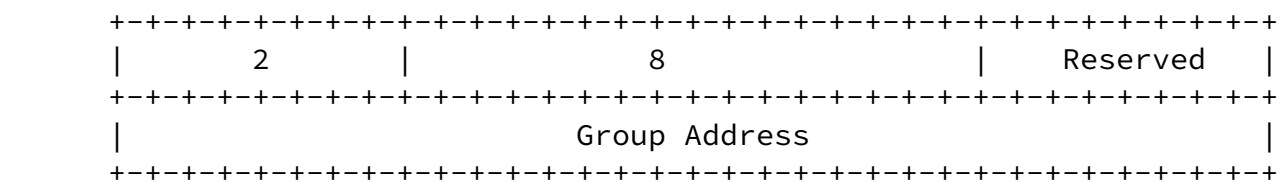
Source Address
The IP address of the active source.

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 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, it may send an SA-Request message for the group.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```



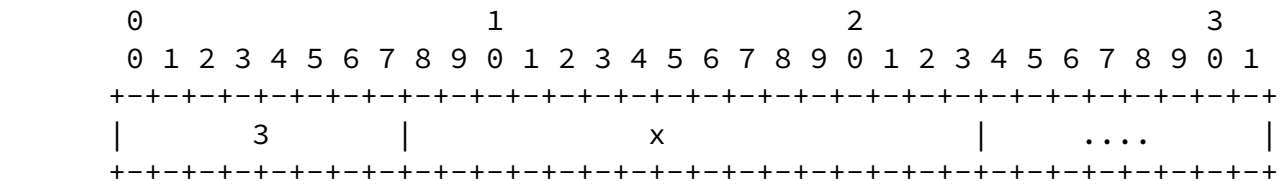
Type
IPv4 Source-Active Request TLV is type 2.

Reserved
Must be transmitted as zero and ignored on receipt.

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.



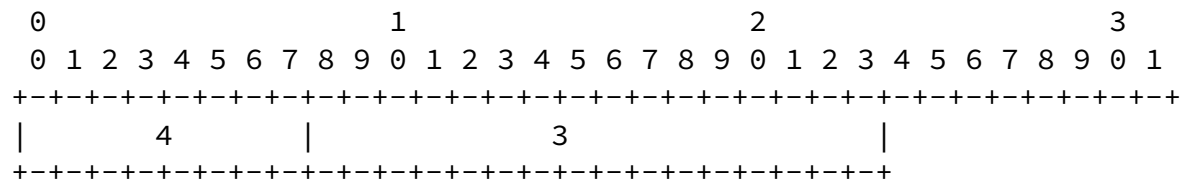
Type
IPv4 Source-Active Response TLV is type 3.

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

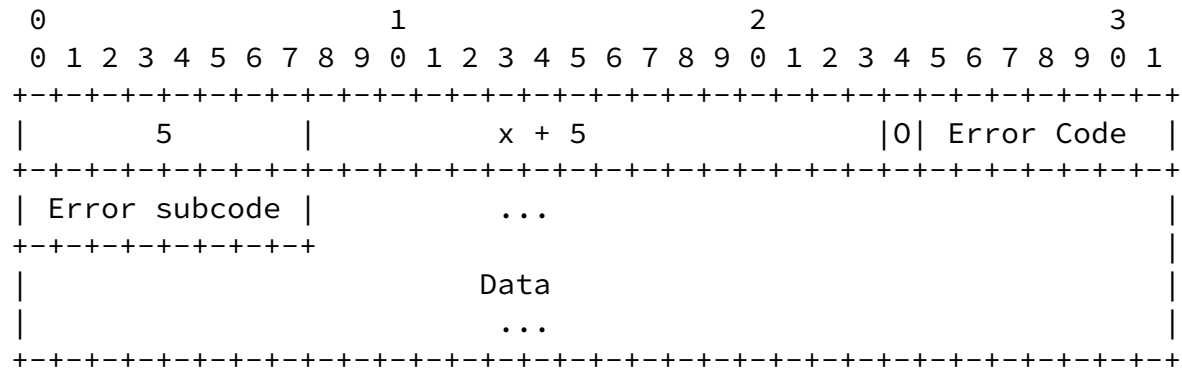
MSDP messages sent to the peer within [KeepAlive-Period] seconds. This message is necessary to keep the MSDP connection alive.



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 MAY be cleared.

Message Header Error subcodes:

0 - Unspecific	(MC)
2 - Bad Message Length	(MC)
3 - Bad Message Type	(CC)

SA-Request Error subcodes (the O-bit is always clear):

0 - Unspecific	(MC)

```
         1 - 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 (the O-bit is always clear):

```
         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 MAY 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).

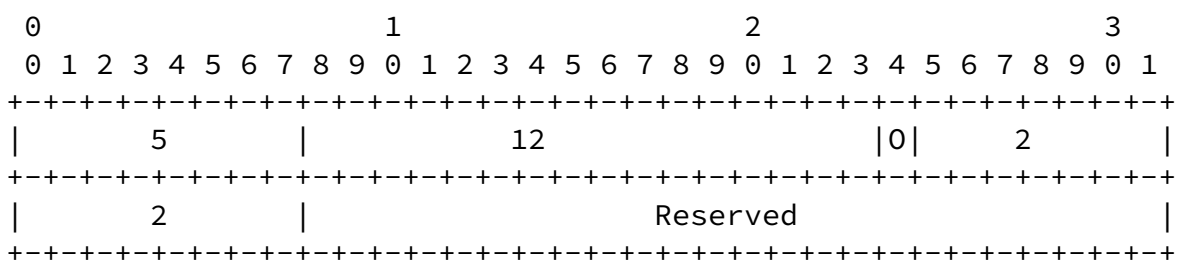
If the Length field of the message header is less than 4 or greater than 1400, or the length of a KeepAlive message is not equal to 3, then the Error Subcode is set to Bad Message Length.

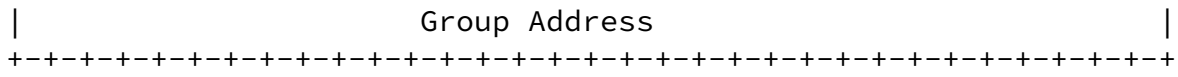
If the Type field of the message header is not recognized, then the Error Subcode is set to Bad Message Type.

17.2. SA-Request Error Handling

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

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

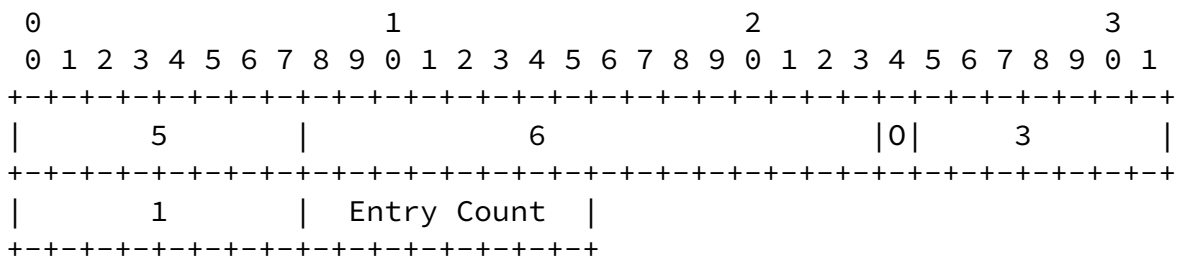




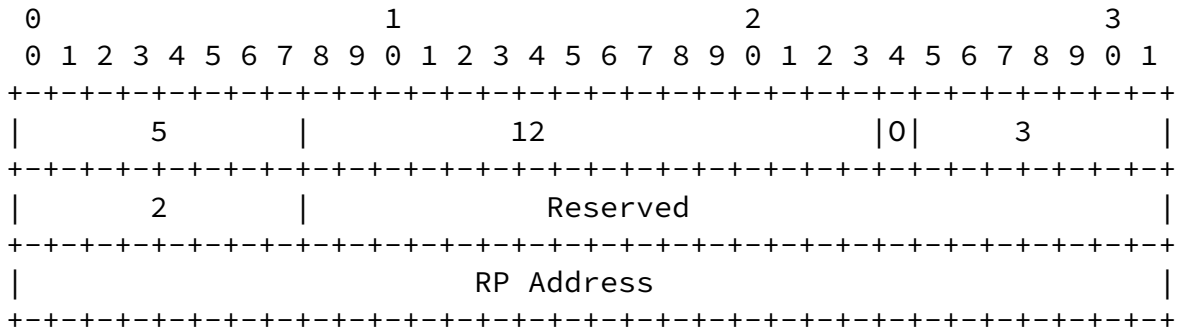
17.3. SA-Message/SA-Response Error Handling

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:

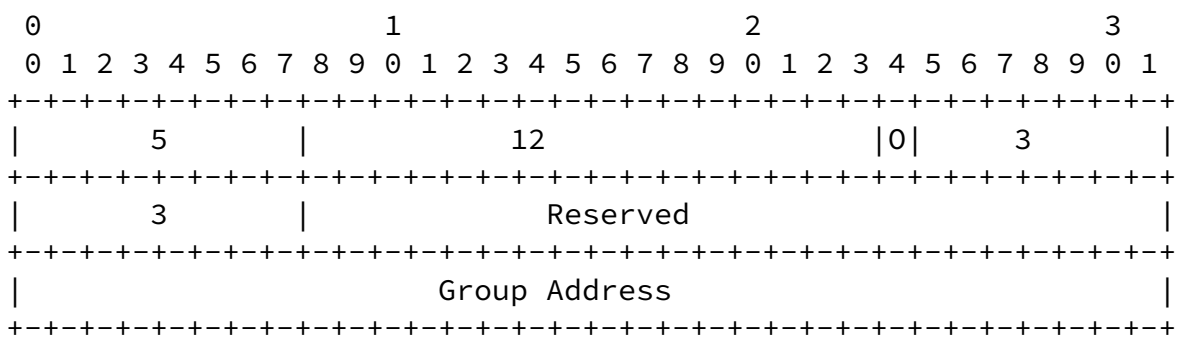
17.3.1. Invalid Entry Count (IEC)



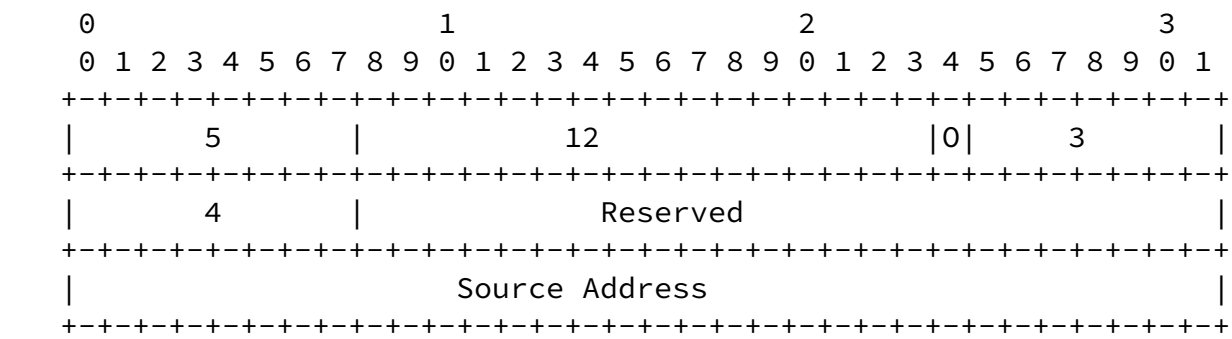
17.3.2. Invalid RP Address



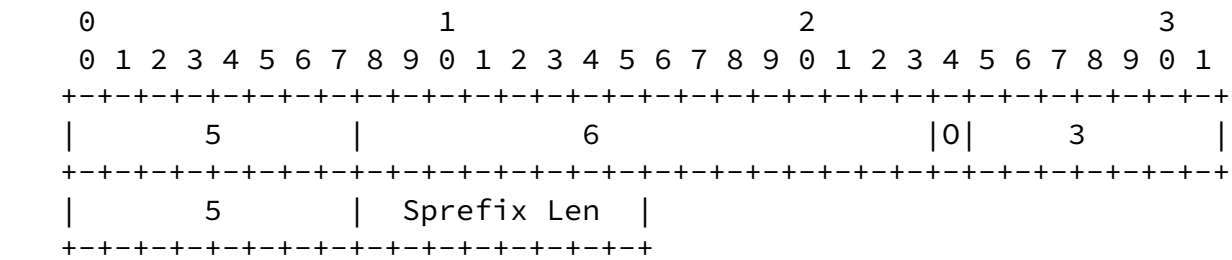
17.3.3. Invalid Group Address



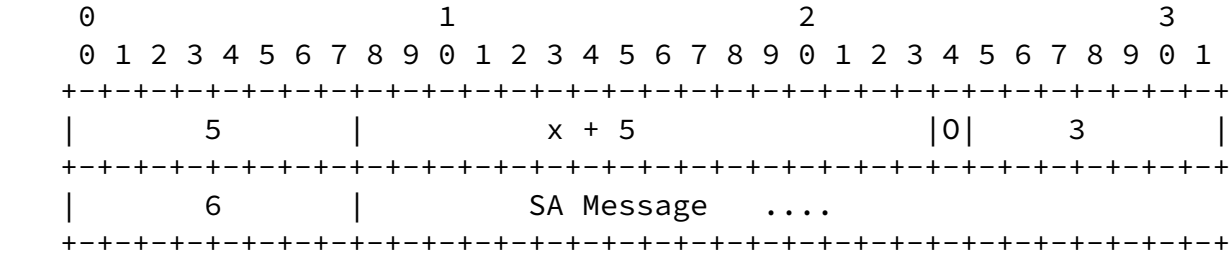
17.3.4. Invalid Source Address



17.3.5. Invalid Sprefix Length (ISL)



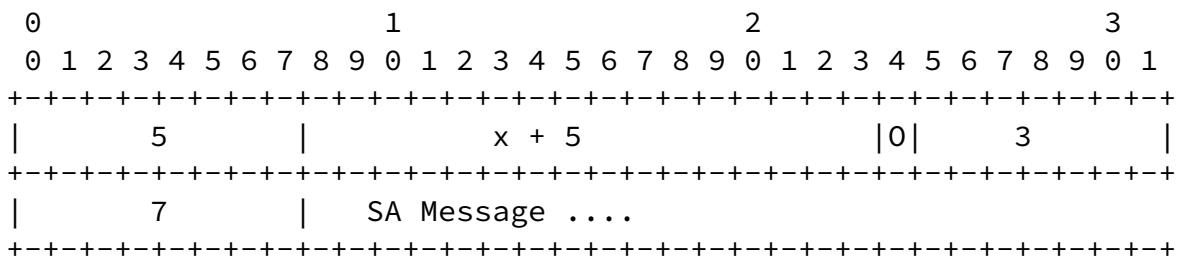
17.3.6. Looping SAs (Self is RP in received SA)



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

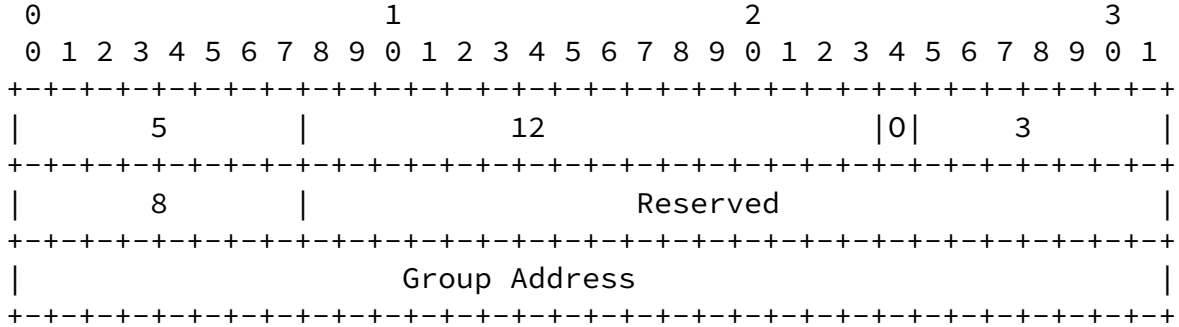
This notification is sent on receipt of SA data that is encapsulated in an unknown encapsulation type. See section 18 for known encapsulations.



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 has not received any MSDP message 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

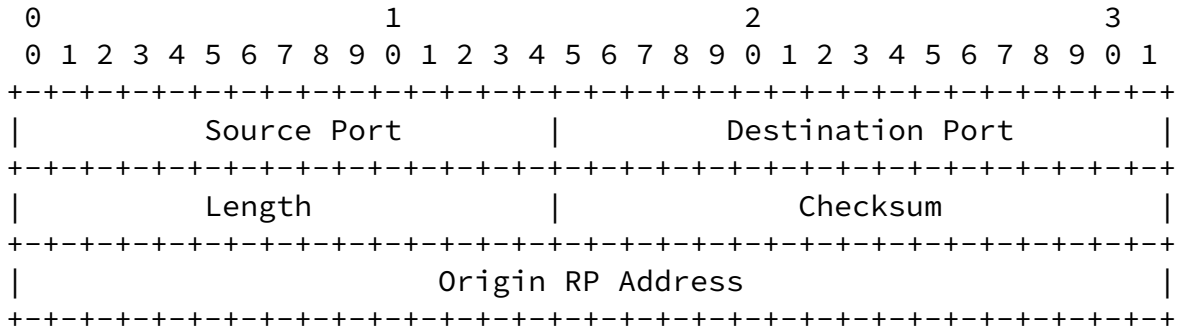
In absence of any fatal errors (that are indicated in this section), an MSDP node may choose at any given time to close its MSDP connection by sending the Notification message with Error Code Cease. However, the Cease Notification message MUST NOT be used when a fatal error indicated by this section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

18. SA Data Encapsulation

This section describes UDP, GRE, and TCP encapsulation of data packets to be included with SA messages. 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:



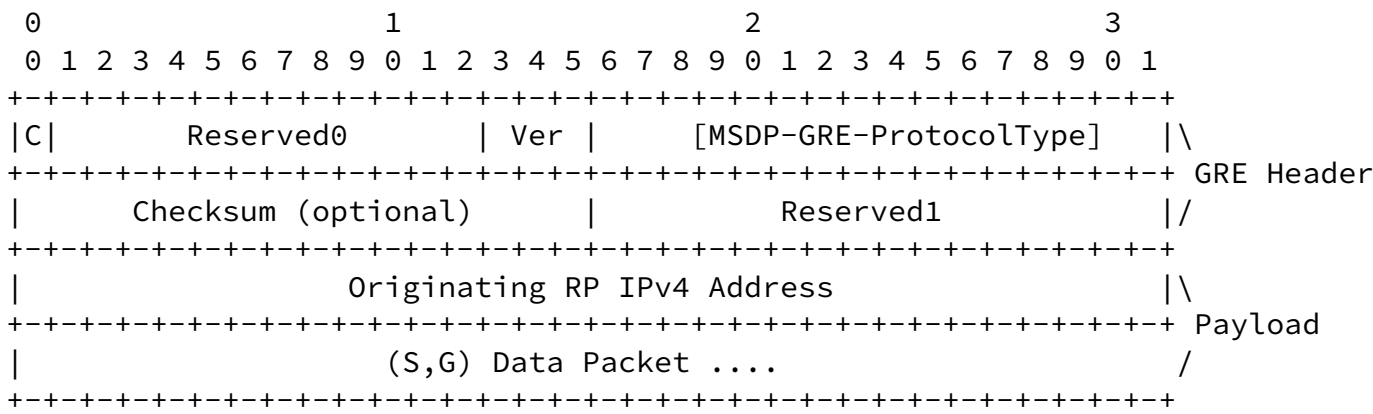
The Source port, Destination Port, Length, and Checksum are used according to RFC 768. Source and Destination ports are known via an implementation-specific method (e.g. per-peer configuration).

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 [RFC2784]. 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. IANA Considerations

The IANA should assign 0x0009 from the IANA SNAP Protocol IDs [IANA] to MSDP-GRE-ProtocolType.

20. Security Considerations

An MSDP implementation MUST use IPsec [RFC2401] to secure control messages. In particular, the TCP connection between MSDP peers MUST be secured using IPsec. When encapsulating data packets 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.

---

21. Acknowledgments

The editors would like to thank the original authors, Dino Farinacci, Yakov Rehkter, Peter Lothberg, Hank Kilmer, and Jermey Hall for their orginal contribution to the MSDP specification. In addition, Bill Nickless, John Meylor, Liming Wei, Manoj Leelanivas, Mark Turner, John Zwiebel, Cristina Radulescu-Banu, Brian Edwards, Selina

Priestley and IJsbrand Wijnands provided useful and productive design feedback and comments. In addition to many other contributions, Tom Pusateri, Kristofer Warell, Henning Eriksson, and Thomas Eriksson helped to clarify the connection state machine, Dave Thaler helped to clarify the Notification message types. Ravi Shekhar helped clarify the semantics of mesh-groups, and countless others helped to clarify the Peer-RPF rules.

22. Editors' Address:

David Meyer
Sprint
12502 Sunrise Valley Drive
Reston VA, 20191
Email: dmm@sprint.net

Bill Fenner
AT&T Labs -- Research
75 Willow Road
Menlo Park, CA 94025
Email: fenner@research.att.com

23. REFERENCES

[IANA] http://www.iana.org

[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.

[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.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2784] Farinacci, D., et al., "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

24. Full Copyright Statement

kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of

---

developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.