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
January, 2000

Multicast Source Discovery Protocol (MSDP)
<draft-ietf-msdp-spec-02.txt>

1. Status of this Memo

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

An RP (or other MSDP SA originator) in a PIM-SM [RFC2362] domain will have a MSDP peering relationship with a MSDP speaker in another domain. The peering relationship will be made up of a TCP connection in which control information exchanged. Each domain will have one or more connections to this virtual topology.

The purpose of this topology is to have 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 speakers can be realized by the underlying 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:

- 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 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 10 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.

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 (See [RFC2362] Section 3.2.2). 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. The advantage of caching is that newly formed MSDP peers can get MSDP (S,G) state sooner and therefore reduce join latency for new joiners. In addition, caching greatly aids in diagnosis and debugging of various problems.

7.1. Timers

The main timers for MSDP are: SA-Advertisement-Timer, SA-Hold-Down-Timer, SA Cache entry timers, and KeepAlive timer.

7.1.1. 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-Timer] MUST be 60 seconds. An RP will not send more than one 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.

7.1.1.1. SA-Advertisement-Timer Processing

When an RP is processing a PIM register message, it encapsulates the data (if any) in an SA message and sends the SA message it to each of its peers. The RP starts the SA Advertisement-Timer for the (S,G) at this time. When the timer expires, and there is (S,G) state for a source within the RP's domain, an (S,G)-SA message is sent to each peer and the timer is reset to [SA-Advertisement-Timer] seconds. If no (S,G) state exists, the timer is deleted.

The following table summarizes (S,G)-SA-Advertisement-Timer processing:

Set to [SA-Advertisement-Timer]	When created off Register packet	Applies to (S,G)
Reset to [SA-Advertisement-Timer]	When Timer expires and (S,G) state exists and was created by a register	Applies to (S,G)
Deleted [SA-Advertisement-Timer]	When Timer expires and (S,G) state has expired	Applies to (S,G)

Note that a caching implementation may also wish to check the SA-Cache on this timer event.

7.1.2. 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 is started when an (S,G)-SA message is initially received by a caching MSDP speaker. The timer is reset to [SA-State-Timer] if another (S,G)-SA message is received before the (S,G)-SA-State-Timer expires. [SA-State-Timer] MUST NOT be less than 90 seconds. The following table summarizes SA-State-Timer processing:

Set to [SA-State-Timer]	When creating (S,G)-SA cache entry (on receipt of a (S,G)-SA message)	Applies to (S,G)-SA Cache Entry

Reset to [SA-State-Timer]	When On receipt of (S,G)-SA message	Applies to (S,G)-SA Cache Entry
Deleted (S,G) SA Cache entry	When Timer expires	Applies to (S,G)-SA Cache Entry

7.1.3. SA-Hold-Down-Timer

A caching MSDP speaker SHOULD NOT forward an SA message it has received in the last SA-Hold-Down interval. [SA-Hold-Down-Timer] SHOULD be set to 30 seconds. The following table summarizes SA-Hold-Down-Timer processing:

Set to [SA-Hold-Down-Timer]	When Upon receipt of (S,G)-SA message	Applies to (S,G)-SA Cache En
Reset to [SA-Hold-Down-Timer]	When When forwarding (S,G)-SA message	Applies to (S,G)-SA Cache En
Deleted (S,G)-SA-Hold-Down-Timer]	When (S,G)-SA entry is deleted	Applies to (S,G)-SA Cache En

7.1.4. KeepAlive Timer

Set to [KeepAliver-Timer]	When passive-connect peer comes up	Applies to each peer
Reset to [KeepAliver-Timer]	When Receipt of data from peer	Applies to each peer

Deleted	When	Applies to
KeepAliver-Timer	Timer expires	each peer
	or passive-connect peer	
	closes connection	

7.2. Intermediate MSDP Speakers

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 its own sources.

7.3. 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.

7.4. 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 12 for discussion of error handling relating to SA requests and responses.

8. 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 the source which 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.

9. 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 different 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. Another example is the Anycast RP mechanism [ANYCASTRP].

10. 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.

10.1. 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, 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, pick one via a deterministic rule.

(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.

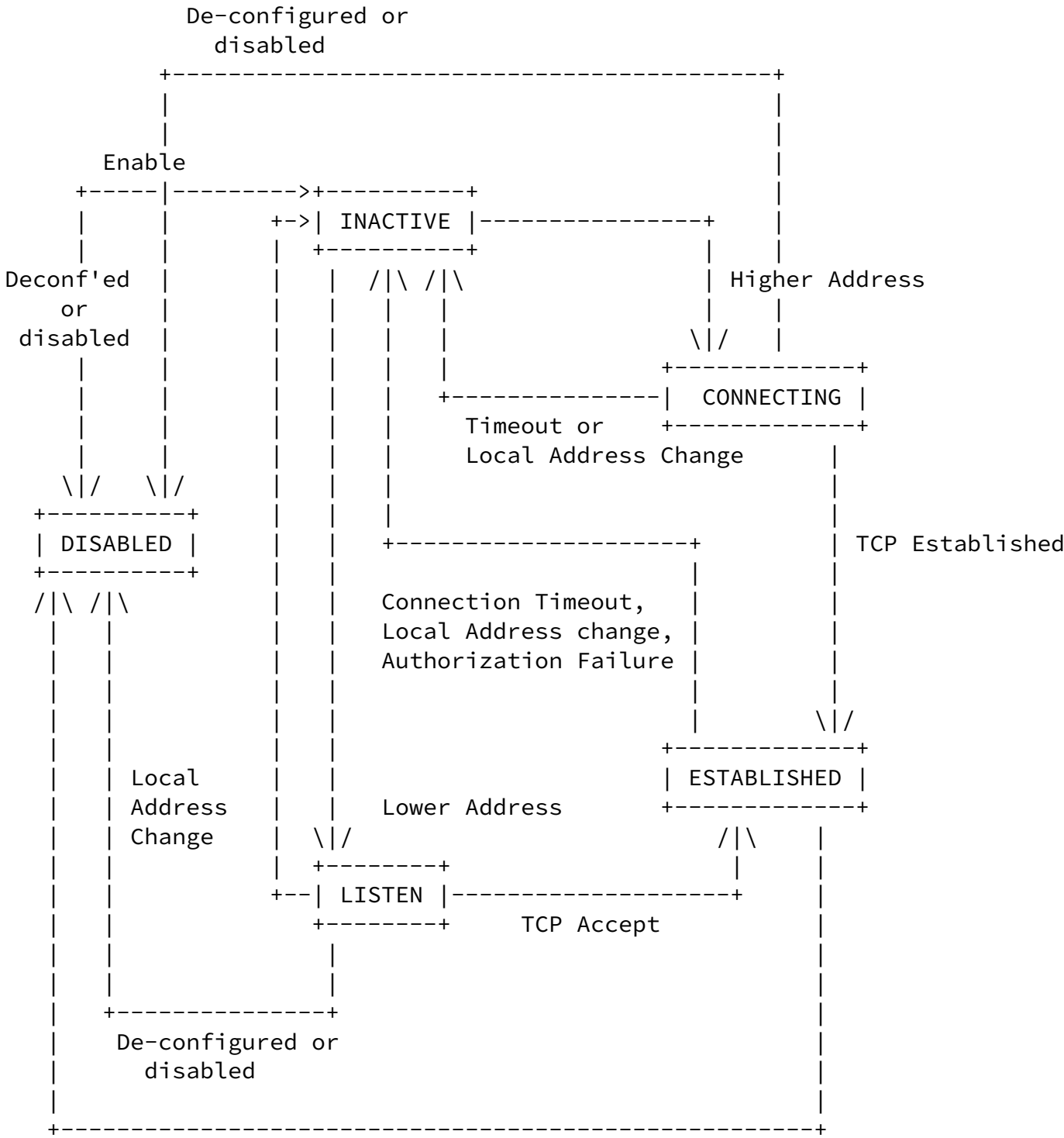
10.2. MSDP default-peer semantics

A 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. A MSDP speaker 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 to occur.

11. MSDP Connection Establishment

MSDP messages will be encapsulated in a TCP connection using well-known 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 on 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 speaker starts in the INACTIVE state. MSDP speakers establish peering sessions according to the following state machine:

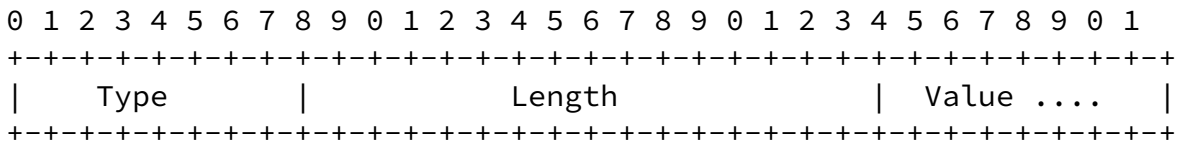

De-configured or
disabled
Enable
Deconf'ed
or
disabled
INACTIVE
Higher Address
CONNECTING
Timeout or
Local Address Change
DISABLED
TCP Established
Connection Timeout,
Local Address change,
Authorization Failure
Local
Address
Change
Lower Address
ESTABLISHED
LISTEN
TCP Accept
De-configured or
disabled

De-configured or
disabled

12. 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.

12.1. 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. The minimum length required is 3 octets.

Value (variable length)
Format is based on the Type value. See below. The length of the value field is Length field minus 3.

12.2. Defined TLVs

The following TLV Types are defined:

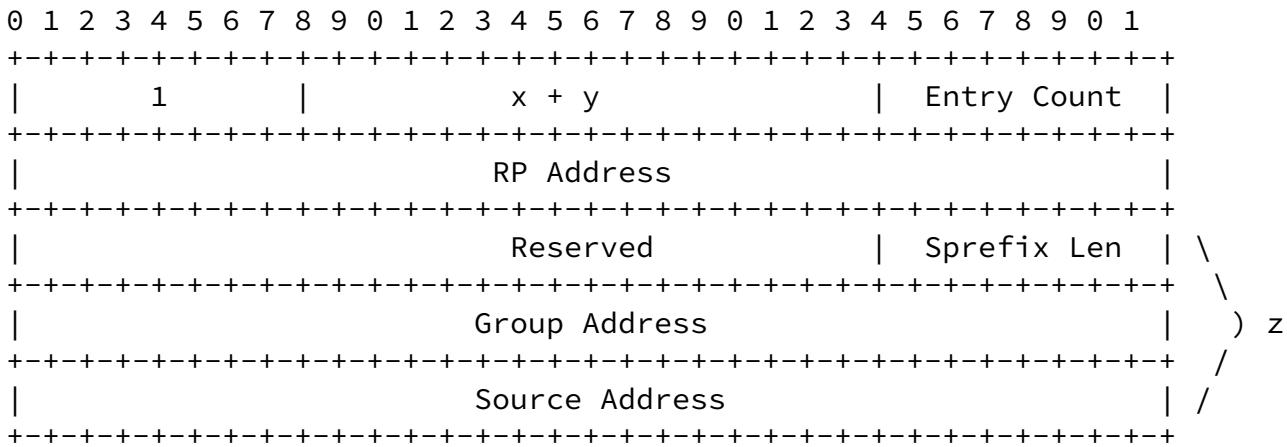
Code	Type

```
============================================================
 1                  IPv4 Source-Active
 2                  IPv4 Source-Active Request
 3                  IPv4 Source-Active Response
 4                  KeepAlive
 5                  Notification
```

Each TLV is described below.

12.2.1. 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 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.

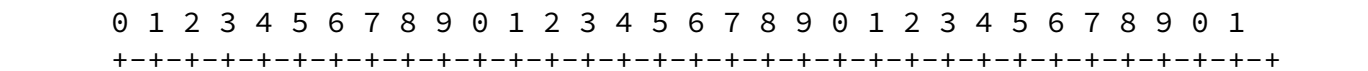
Group Address
The group address the active source has sent data to.

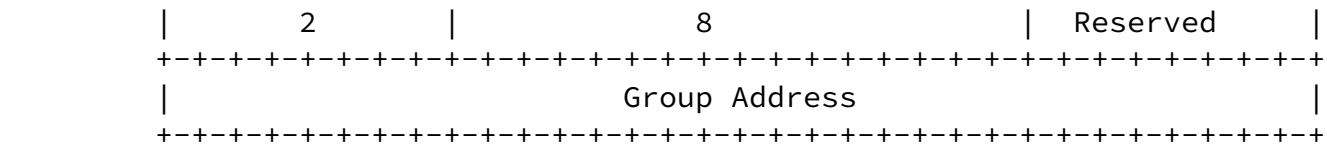
Source Address
The IP address of the active source.

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.

12.2.2. 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.





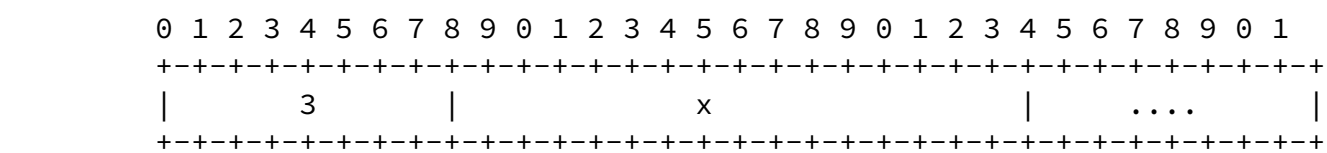
Type
IPv4 Source-Active Request TLV is type 2.

Reserved
The Reserved field MUST be transmitted as zeros and ignored by a receiver.

Group Address
The group address the MSDP peer is requesting.

12.2.3. 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.



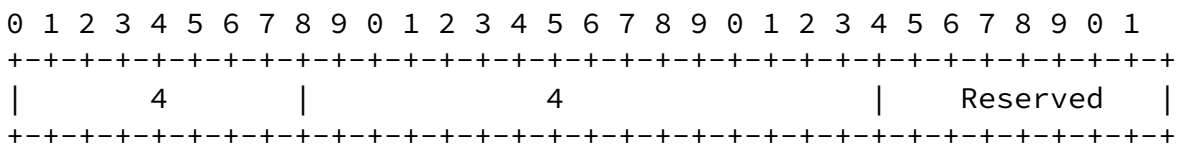
Type
IPv4 Source-Active Response TLV is type 3.

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.

12.2.4. 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

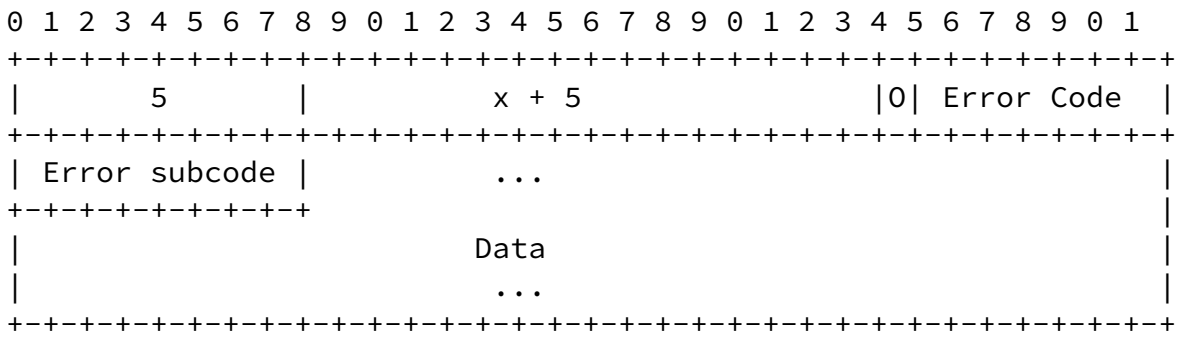
passive connect side of the connection knows that the connection will be reestablished when a TCP SYN packet is sent from the active connect side. However, the active connect side will not know when the passive connect side goes down. Therefore, the KeepAlive timeout will be used to reset the TCP connection.



The length of the message is 4 octets which encompasses the 1-octet Type field and the 2-octet Length field, plus the Reserved field. The Reserved field MUST be transmitted as zeros and ignored by a receiver.

12.2.5. Notification TLV

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



Type
The Notification TLV is type 7.

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 reset, the connection will be closed [MASC].

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 12.3
2	Finite State Machine Error	Section 12.4
3	Notification Message Error	Section 12.5
4	SA-Request Error	Section 12.6
5	SA-Response Error	Section 12.7
6	SA-Message Error	Section 12.8

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 reset (i.e. the connection will be
 closed). The used notation in the error description below is: MC =
 Must Close connection = O-bit reset; CC = Can Close connection =
 O-bit might be reset [MASC].

Message Header Error subcodes:

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

Finite State Machine Error subcodes:

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

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

```
          0 - Unspecific                          (CC)

   SA-Request Error subcodes:

          0 - Not caching                         (MC)
          0 - Invalid Group Address prefix        (CC)

   SA-Reponse Error subcodes:

          0 - Didn't send Request                 (MC)

   SA-Message Error subcodes

          0 - Invalid Entry Count                 (CC)
          1 - Invalid RP Address                  (CC)
          2 - Invalid Group Address               (CC)
          3 - Invalid Source Address              (CC)
          4 - Invalid Sprefix Length              (CC)
          5 - Looping SA (Self is RP)             (CC)
          6 - Unknown Encapsulation               (MC)
```

12.3. 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 4, 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.

12.4. 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.

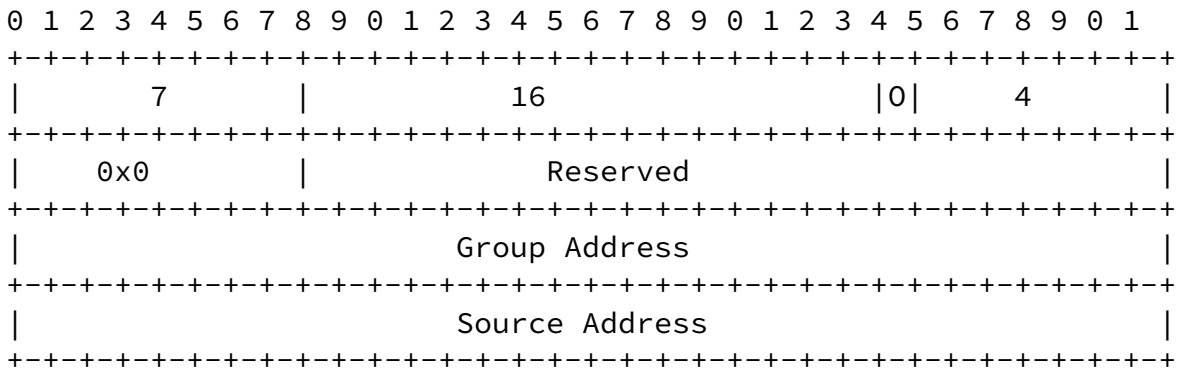
12.5. Notification Message Error Handling

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

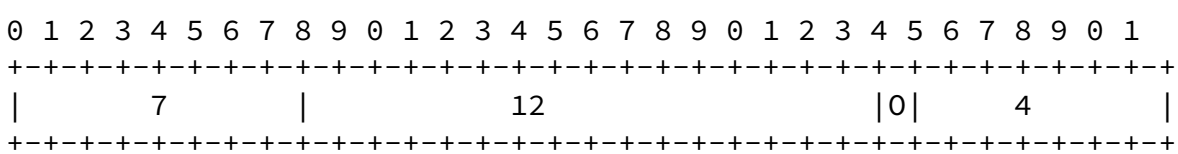
12.6. SA-Request Error Handling

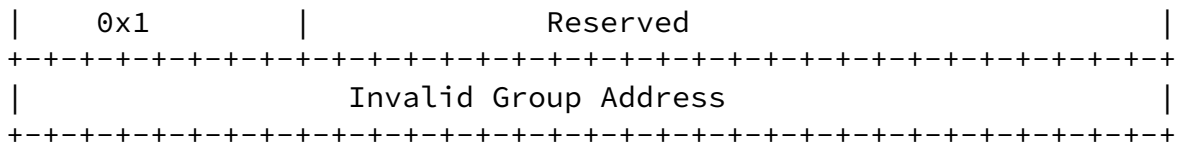
The SA-Request Error code is used to signal the receipt of a SA request at a non-caching MSDP speaker, or at a caching MSDP speaker when an invalid group address requested.

When a non-caching MSDP speaker receives an SA-Request, it returns the following notification and closes the connection:



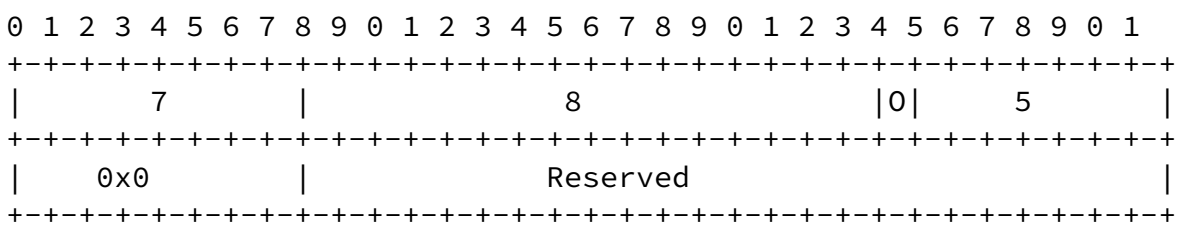
If a caching MSDP speaker receives a request for an invalid group, it returns the following notification and closes the connection:





12.7. SA-Response Error Handling

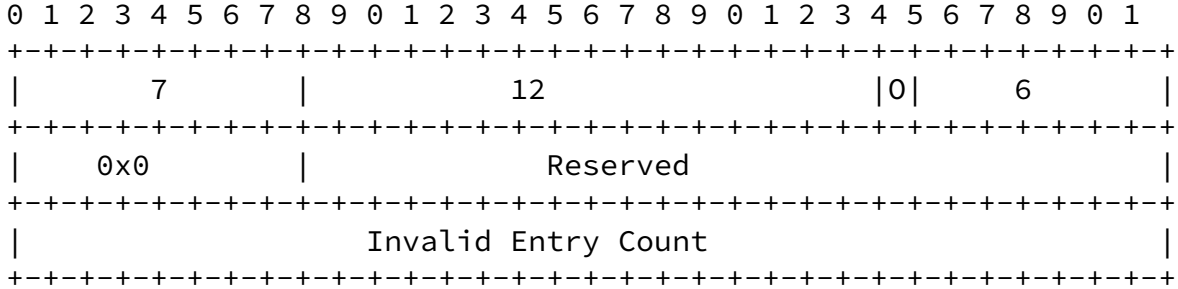
The SA-Response Error code is used to signal the receipt of a SA Response at MSDP speaker which did not issue a SA-Request to the peer. It has the following form:



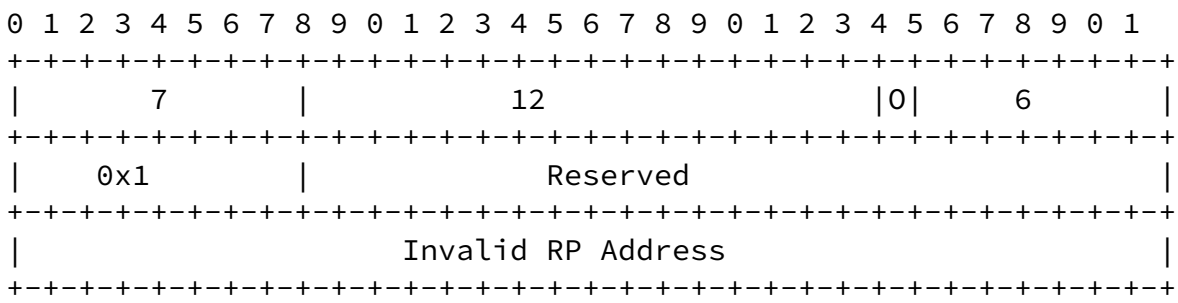
12.8. SA-Message Error Handling

The SA-Message Error code is used to signal the receipt of an SA message that contains invalid data.

12.8.1. Invalid Entry Count



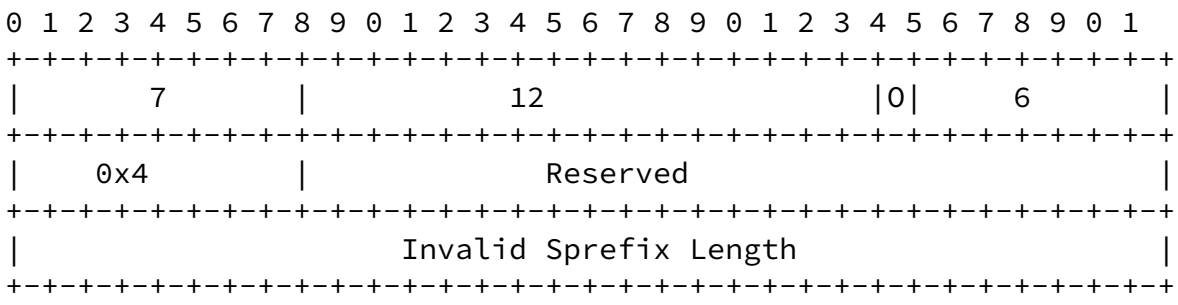
12.8.2. Invalid RP Address



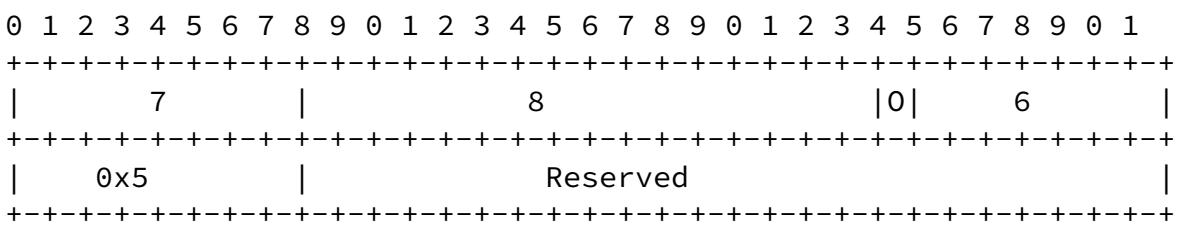
12.8.3. Invalid Group Address

12.8.4. Invalid Source Address

12.9. Invalid Sprefix Length

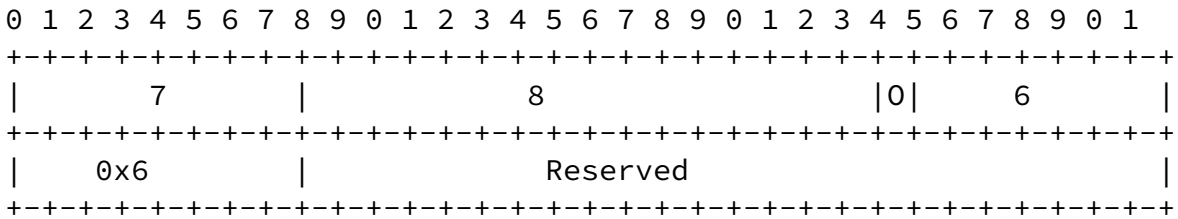


12.10. Looping SAs (Self is RP in received SA)



12.11. Unknown Encapsulation

This notification is sent on receipt of SA data that is encapsulated in an unknown encapsulation type. See section 12.12 for known encapsulations.

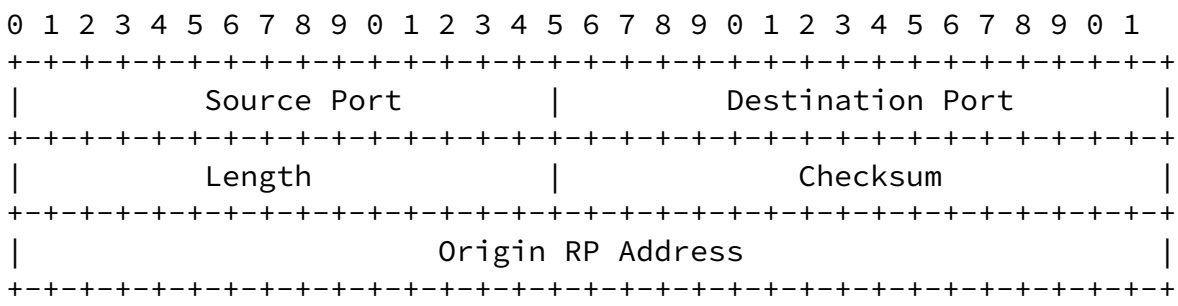


12.12. SA Data Encapsulation

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

12.12.1. UDP Data Encapsulation

MSDP SA-data MAY be encapsulated in UDP. In this case, the UDP psuedo-header has the following form:



Source Port
Port to be used by the remote end, and is known via configuration.

Destination Port
The Destination Port is set to the remote endpoint's Source port, and is known via configuration.

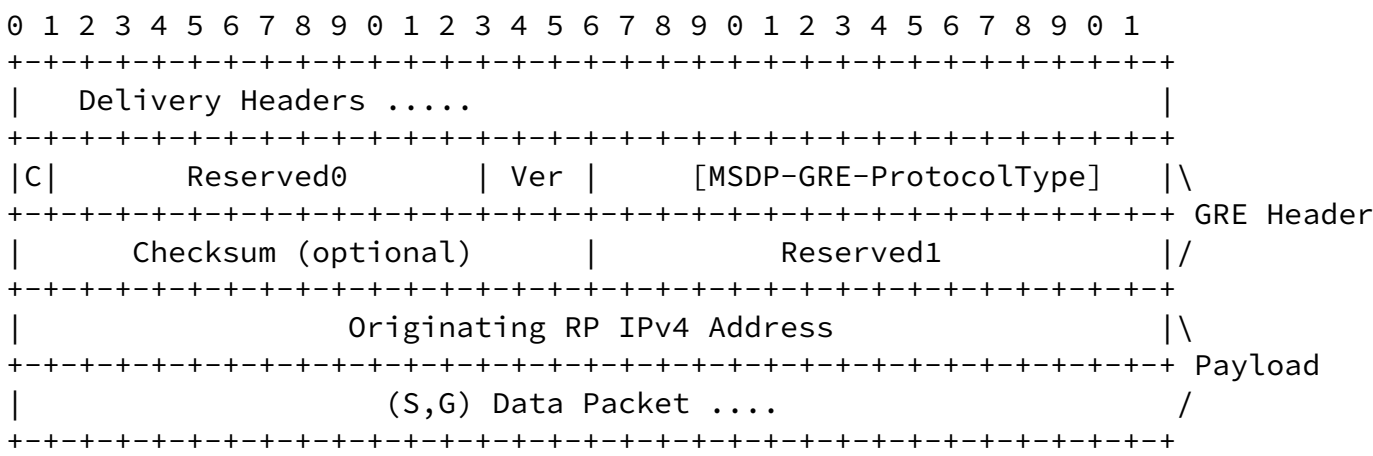
Length
Length is the length in octets of this user datagram including this header and the data. The minimum value of the length is twelve.

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.

12.12.2. GRE Encapsulation

MSDP SA-data MAY be encapsulated in GRE using protocol type [MSDP-GRE-ProtocolType].



12.12.2.1. GRE Encapsulation and PMTU 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.

13. 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.

14. Acknowledgments

The authors would like to thank Dave Thaler, Bill Nickless, John Meylor, Liming Wei, Manoj Leelanivas, Mark Turner, John Zwiebel, and Cristina Radulescu-Banu for their design feedback and comments. Bill Fenner also made many contributions, including clarification of the Peer-RPF rules.

15. 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

16. REFERENCES

[ANYCASTRP] Meyer, et. al, "Anycast RP mechanism using PIM and MSDP", draft-ietf-mboned-anycast-rp-04.txt, November, 1999. Work in Progress.

[GRE] Farinacci, D., et al., "Generic Routing Encapsulation (GRE)", draft-meyer-gre-update-02.txt, January, 2000. Work in Progress.

[MASC] Estrin, D., et al., "The Multicast Address-Set Claim (MASC) Protocol", draft-ietf-malloc-masc-04.txt, October, 1999. 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.

[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.