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
Decemeber, 1999

Multicast Source Discovery Protocol (MSDP)
<draft-ietf-msdp-spec-00.txt>

1. Status of this Memo

Abstract

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

domains.

2. Copyright Notice

3. 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:

3.1. No Third-party resource dependencies on RP

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

3.2. Receiver only Domains

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

3.3. 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.

4. Overview

An RP (or other MSDP SA originator) in a PIM-SM domain will have a MSDP peering relationship with an RP in another domain. The peering relationship will be made up of a TCP connection in which control information is primarily exchanged. Each domain will have a connection 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 RPs can be realized by the underlying BGP routing system.

5. 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 the section on "MSDP Peer-RPF Forwarding" for more details.

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 it's 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. In this case, an implementation SHOULD forward an SA message (which was originated from the RP address covered by that route) to the peer. This is known in other circles as Split-Horizon with Poison Reverse.

When an MSDP peer which is also an RP for its own domain receives an 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 an (*,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 an (S,G) join event towards the data source as if a Join/Prune message was received addressed to the RP itself (See [1] 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 SA's, then the RP should trigger an (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.

6. 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.

6.1. Timers

The main timers for MSDP are: SA Advertisement period, SA Hold-down period, the SA Cache timeout period, KeepAlive, HoldTimer, and ConnectRetry. Each is described below.

6.1.1. SA Advertisement Period

RPs which originate SA messages do it periodically as long as there is data being sent by the source. The SA Advertisement Period MUST be 60 seconds. An RP will not send more than one SA message for a given (S,G) within an SA Advertisment period. 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, 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 an (S,G) join for each SA for that group in its cache.

6.1.2. SA Hold-down Period

A caching MSDP speaker SHOULD NOT forward a SA message it has received in the last SA-Hold-down period. The SA-Hold-down period SHOULD be set 30 seconds.

6.1.3. SA Cache Timeout

A caching MSDP speaker times out it's SA cache at SA-State-Timer. The SA-State-Timer MUST NOT be less than 90 seconds minutes.

6.1.4. KeepAlive, HoldTimer, and ConnectRetry

The KeepAlive, HoldTimer, and ConnectRetry timers are defined in RFC1771 [3].

6.2. 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.

6.3. SA Filtering

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 colocated in the same domain as the source can restrict SA messages. Other MSDP peers in transit domains should not filter or the flood-and-join model does not guarantee that

sources will be known throughout the Internet. An exception occurs at an administrative scope [13] boundary. In particular, a SA message for an (S,G) MUST NOT be sent to peers which are on the other side of an administrative scope boundary for G.

6.4. Caching

If an MSDP peer decides to cache SA state, it may accept SA-Requests from other MSDP peers. When a MSDP peer receives an SA-Request for a group range, it will respond to the peer with a set of SA entries, in a 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.

If an implementation receives an SA-Request message and is not caching SA messages, it sends a notification with Error code 7 subcode 1, as defined in section 11.2.7.

7. 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 number of packets from 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.

Finally, if an implementation supports an encapsulation of SA data other than default TCP encapsulation, then it MUST support GRE encapsulation. In addition, an implementation MUST learn about not TCP encapsulations via capability advertisment (see section 11.2.5).

8. 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 [8].

9. MSDP Peer-RPF Forwarding

The MSDP Peer-RPF Forwarding rules are used for forwarding SA messages throughout an MSDP enabled internet. An SA message originated by a MSDP originator R and received by a MSDP router from MSDP peer N in AS A is accepted if any of the following are true:

(i). If N is R.

(ii). If A is the first AS in the AS-Path of the BGP route towards R.

(iii). If N is the iBGP advertiser of the BGP route towards R.

(iv). If N is the MSDP default-peer.

If none of the conditions above is met, the SA message is discarded. This is the case where the SA message was received on a redundant MSDP peering path.

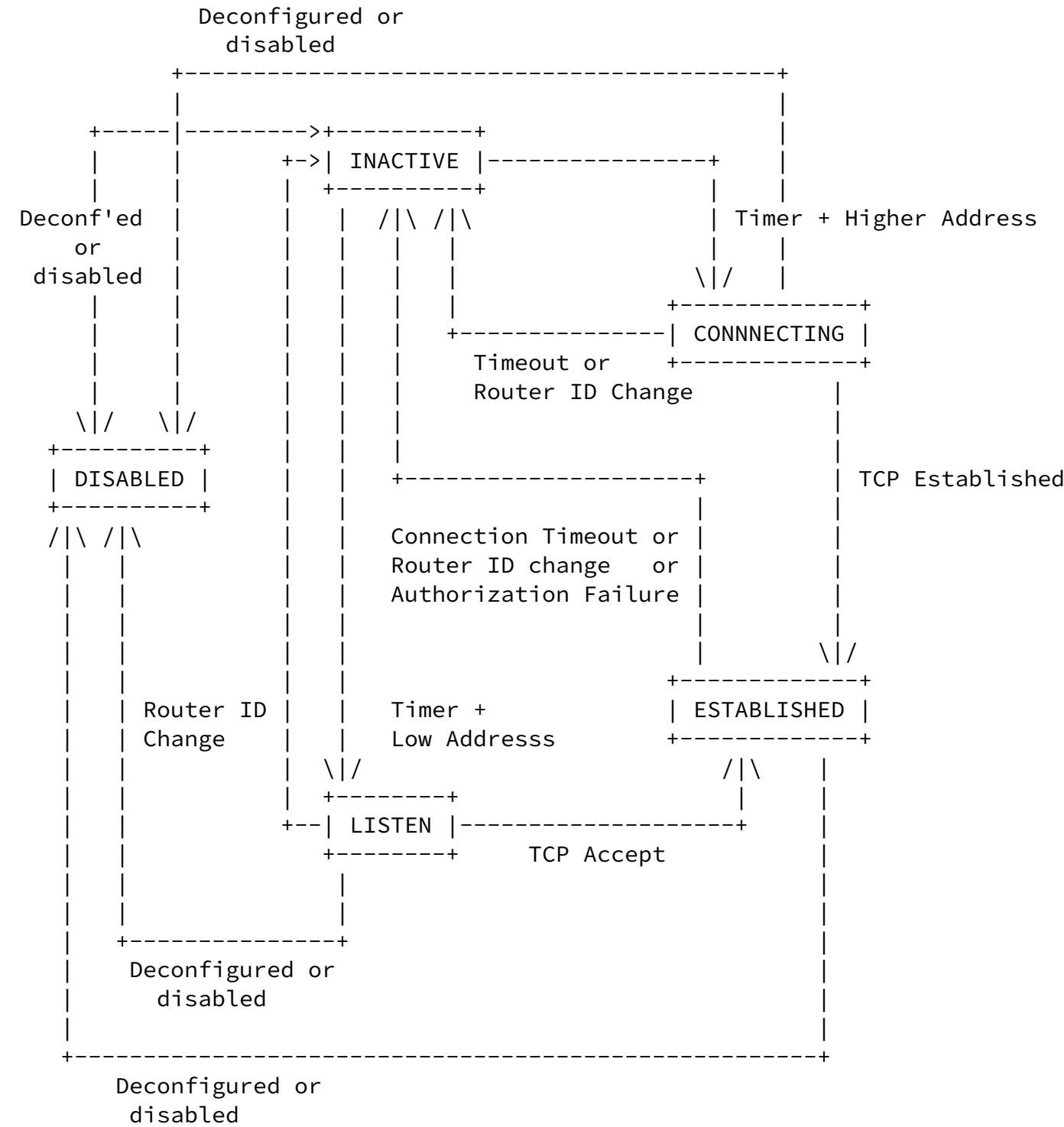
Note that these rules are evaluated in the order shown here. This selects a "peer-RPF neighbor" for the SA message, and allows for the construction of diagnostic tools such as MSDP-traceroute [7].

9.1. 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. In this case, the MSDP speaker accepts all SA messages from the default-peer. Of course, if multiple default peers are configured, the possibility of looping exists, so care must be taken. Finally, a router running BGP or multiprotol BGP [4] SHOULD NOT allow configuration of default peers.

10. MSDP Connection Establishment

MSDP speakers establish peering sessions according to the following state machine:

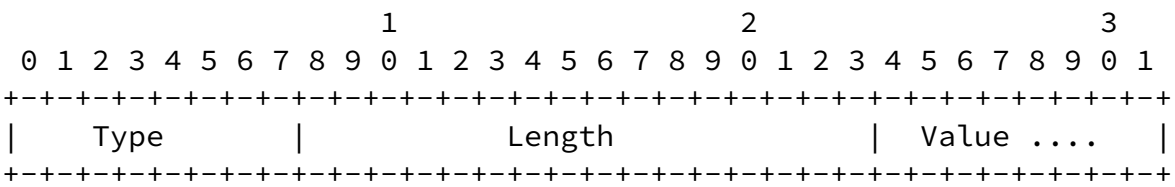


11. Packet Formats

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 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.

Finally, 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.

11.1. MSDP messages will be encoded in 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 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.

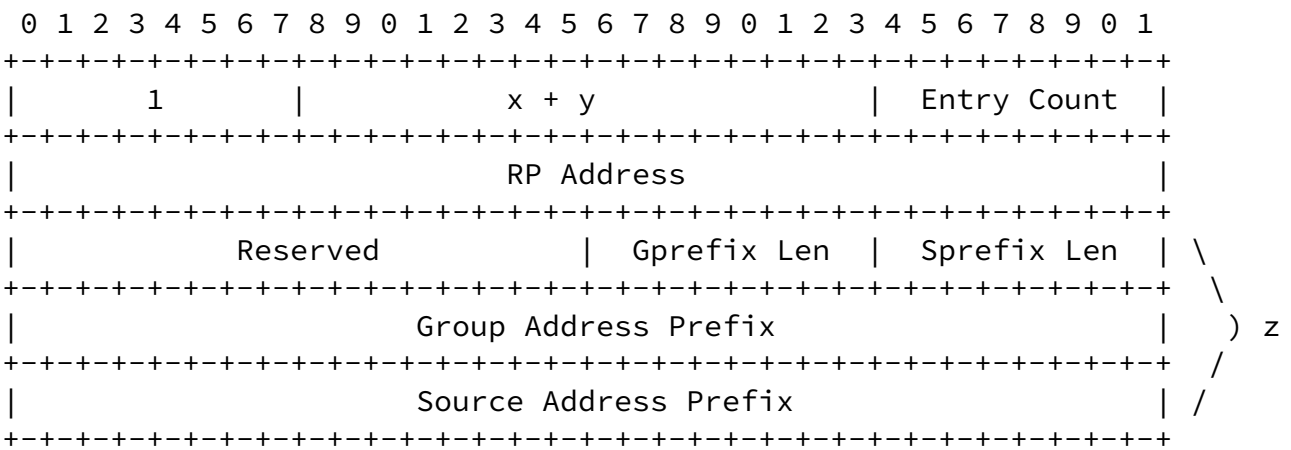
11.2. The following TLV Types are defined:

Code	Type
1	IPv4 Source-Active
2	IPv4 Source-Active Request
3	IPv4 Source-Active Response
4	KeepAlive
5	Encapsulation Capability Advertisement
6	Encapsulation Capability Request
7	Notification
8	GRE Encapsulation

Each TLV is described below.

11.2.1. IPv4 Source-Active TLV

The maximum size SA message that can be sent is 1400 bytes. If an MSDP peer needs to originate a message with information greater than 1400 bytes, it sends successive 1400-byte messages. The 1400 byte 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.

Gprefix Len and Sprefix Len

The route prefix length associated with the group address prefix and source address prefix, respectively.

Group Address Prefix

The group address the active source has sent data to.

Source Address Prefix

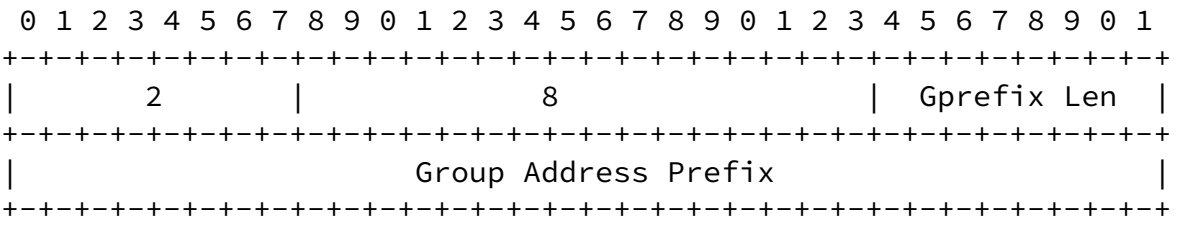
The route prefix associated with the active source.

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.

11.2.2. IPv4 Source-Active Request TLV

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

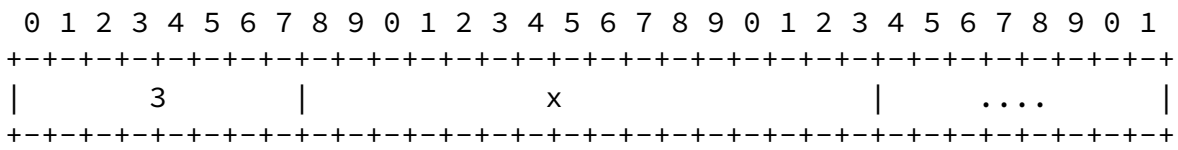
The route prefix length associated with the group address prefix.

Group Address Prefix

The group address prefix the MSDP peer is requesting.

11.2.3. IPv4 Source-Active Response TLV

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.



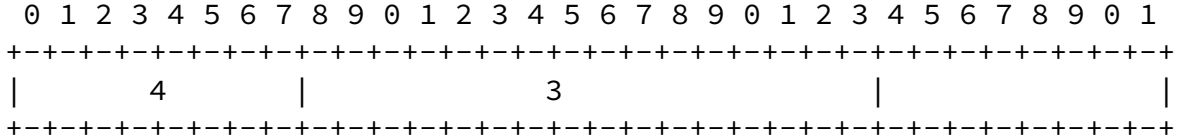
Type
IPv4 Source-Active Response TLV is type 3.

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.

11.2.4. KeepAlive TLV

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.



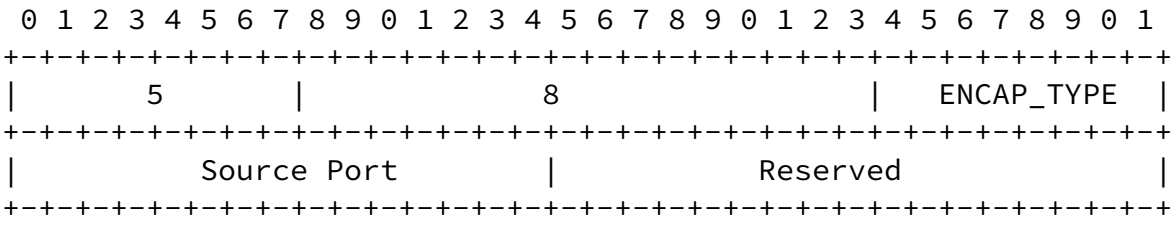
The length of the message is 3 bytes which encompasses the 1-byte Type field and the 2-byte Length field.

11.2.5. Encapsulation Capability Advertisement TLV

This TLV implements encapsulation capability advertisement. This TLV

is sent by an MSDP speaker to advertise its ability to receive data packets encapsulated as described by the TLV (in addition to the default TCP encapsulation).

A MSDP speaker receiving this TLV can choose to either default TCP encapsulation, or may send a IPv4 Encapsulation Request to change to the advertised encapsulation type.



Type
IPv4 Encapsulation Advertisement TLV is type 5.

Length
Length is a two byte field with value 8.

ENCAP_TYPE
The following data encapsulation types are defined for MSDP:

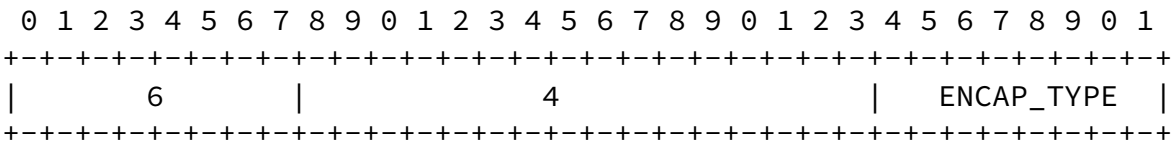
Value	Meaning
0	TCP Encapsulation
1	UDP Encapsulation [10]
2	GRE Encapsulation [9]

Soure Port
Port for use by the requester.

Note that since the TLV does not carry endpoint addresses for the GRE or UDP tunnels, an implementation using these encapsulations MUST use the endpoints that are used for the MSDP peering.

11.2.6. Encapsulation Capability Request TLV

This TLV implements encapsulation capability request. This TLV should be sent in response to a capability advertisement.

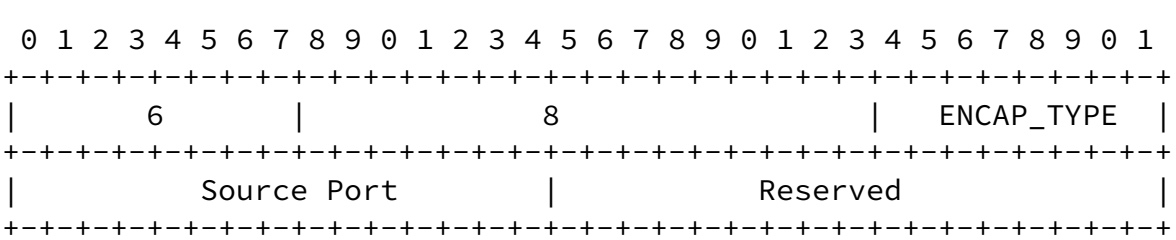


Type
IPv4 Encapsulation Request TLV is type 6.

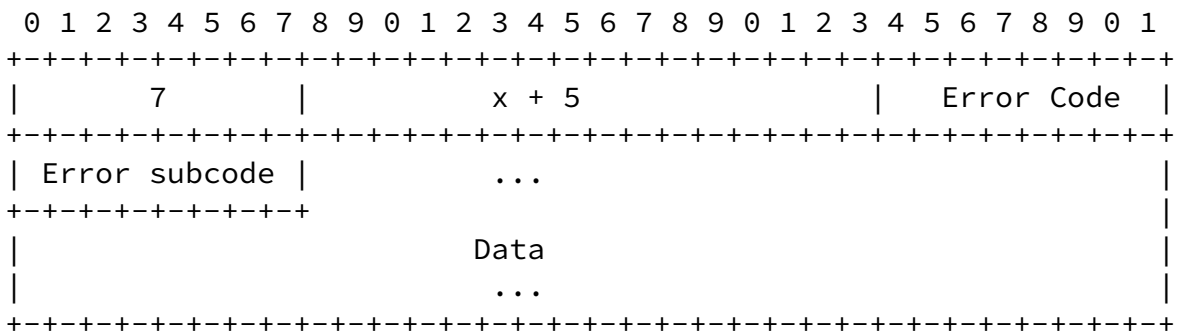
Length
Length is a two byte field with value 4.

ENCAP_TYPE
ENCAP_TYPE is described above.

A requester MAY also provide a source port, in which case the TLV has the following form:



11.2.7. NOTIFICATION TLV



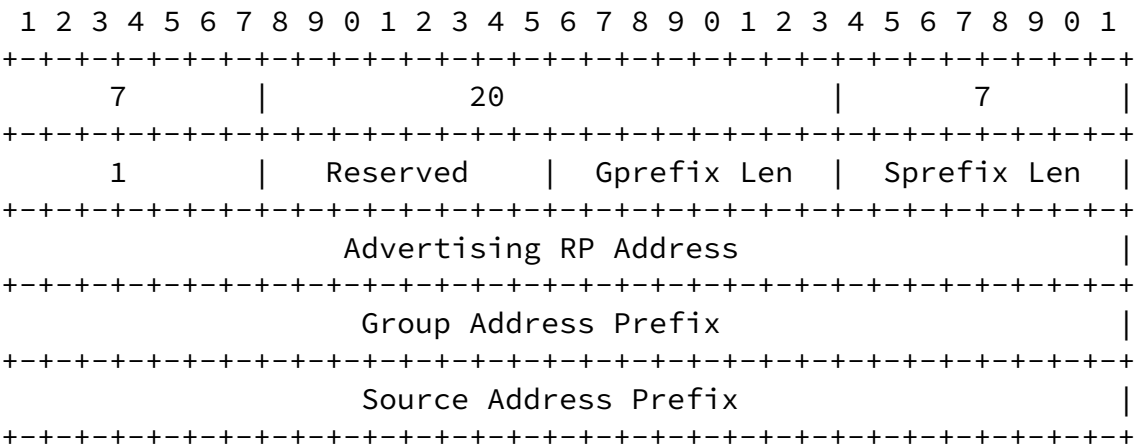
Type
    The Notification TLV is type 7.

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

Error code
    See [3]. In addition, Error code 7 indicates an a SA-Request Error.

Error subcode
    See [3]. In addition, Error code 7 subcode 1 indicates the receipt of a SA-Request message by a non-caching MSDP speaker.

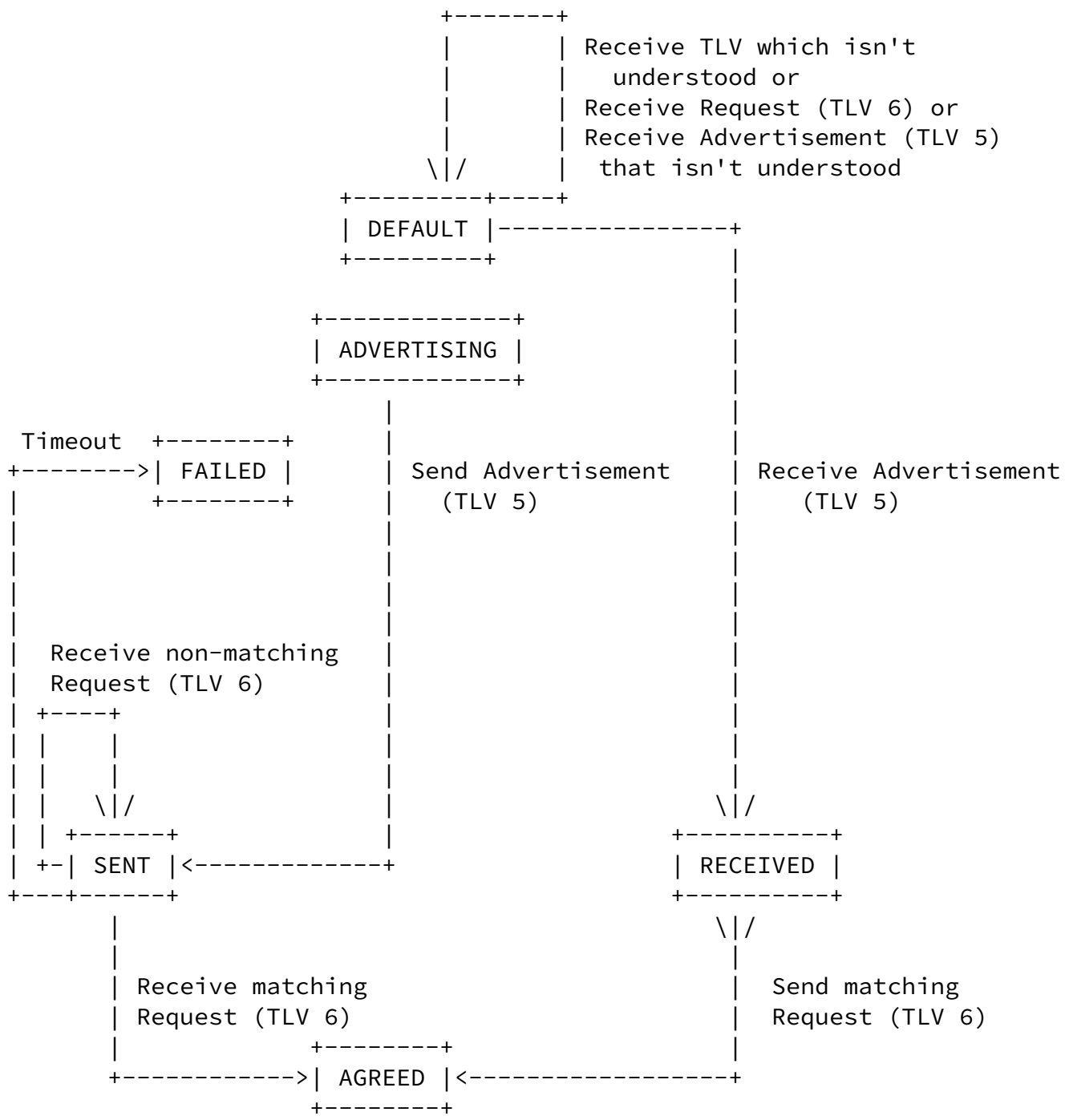
Data
    See [3]. In addition, for Error code 7 subcode 1 (receipt of a SA-Request message by a non-caching MSDP speaker), the TLV has the follwing form:



See [3] for NOTIFICATION error handling.

11.2.8. Encapsulation Capability State Machine

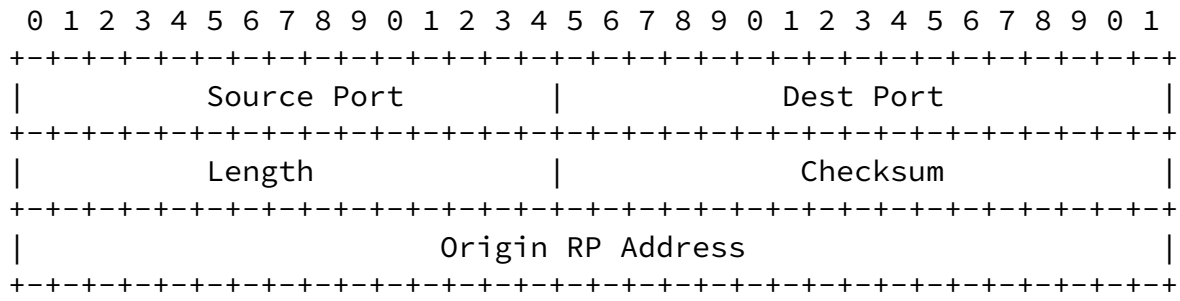
The active connect side of an MSDP peering SHALL begin in ADVERTISING state, and the passive side of the TCP connection begins in DEFAULT state. This will cause the state machine to behave deterministically.


Receive TLV which isn't
understood or
Receive Request (TLV 6) or
Receive Advertisement (TLV 5)
that isn't understood
DEFAULT
ADVERTISING
Timeout
FAILED
Send Advertisement
(TLV 5)
Receive Advertisement
(TLV 5)
Receive non-matching
Request (TLV 6)
SENT
RECEIVED
Receive matching
Request (TLV 6)
Send matching
Request (TLV 6)
AGREED

Note that if an advertiser transitions into the FAILED state, it SHOULD assume that it has an old-style peer which can only support TCP encapsulation. If an implementation wishes to be backwardly compatible, it SHOULD support TCP encapsulation. In addition, a requester in any state other than AGREED MUST only encapsulate data in the TCP stream.

11.2.9. UDP Data Encapsulation

When using UDP encapsulation, the UDP psuedo-header has the following form:



- Source Port

  When using UDP encapsulation, a capability requester uses the advertiser's Source Port as its destination port. The advertiser MUST provide a Source Port.

- Destination Port

  When using UDP encapsulation, a capability advertiser uses the well known port 639 as the destination port. A capability requester MUST listen on this well-known port. The requester MAY provide a Source Port in it's reply to the advertiser.

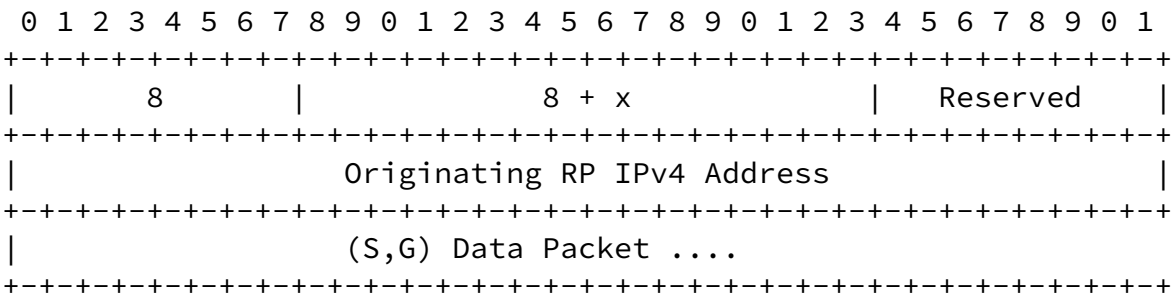
- Length is the length in octets of this user datagram including this header and the data. The minimum value of the length is twelve.

- Checksum is computed according to RFC768 [10].

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

11.2.10. GRE Encapsulation TLV

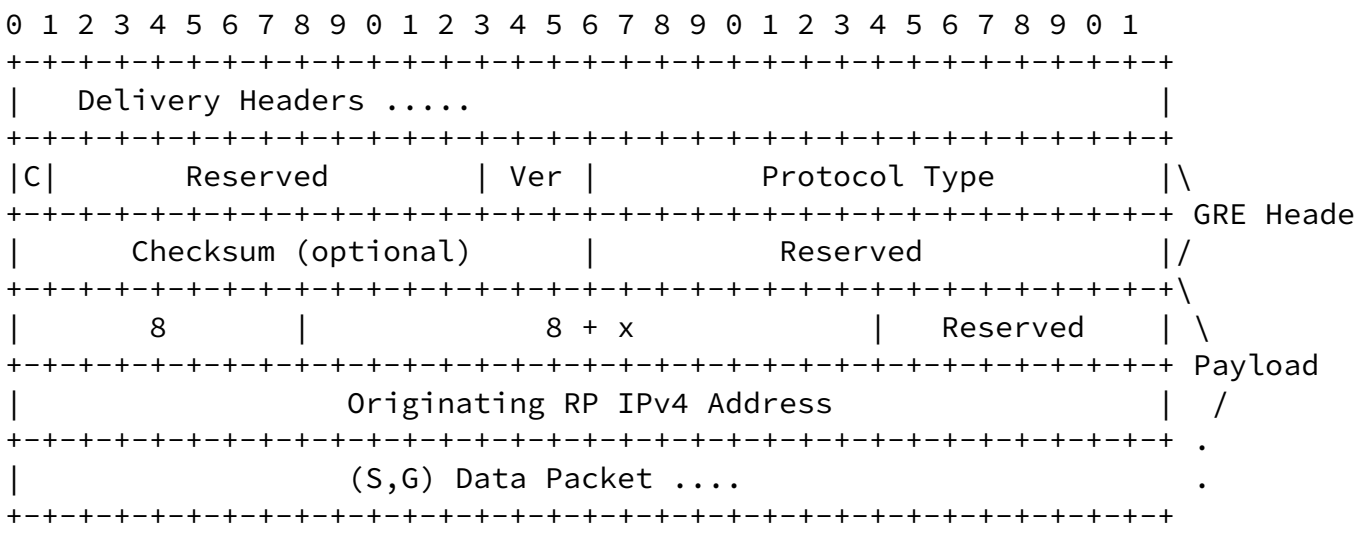
A TLV is defined to describe GRE encapsulated data packets. The TLV has the following form:



Type
GRE encapsulated data packet TLV is type 8.

Length
Length is a two byte field with value 8 + x, where x is the length of the (S,G) Data packet.

The entire GRE header, then, will have the following form:



11.3. MTU Exeeded

If the outbound link MTU is exeeceded by the newly encapsulated packet, the packet SHOULD be dropped.

12. Security Considerations

A MSDP implementation MAY use IPsec [11] or keyed MD5 [12] to secure control messages. Encapsulated data packets rely on the underlying security model.

13. Acknowledgments

The authors would like to thank Dave Thaler, Bill Fenner, Bill Nickless, John Meylor, Liming Wei, Manoj Leelanivas, Mark Turner, and John Zwiebel for their design feedback and comments.

14. Author's Address:

Dino Farinacci
Procket Networks
Email: dino@procket.com

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

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

15. REFERENCES

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

[2] Thaler, D., Estrin, D., Meyer, D., "Border Gateway Multicast Protocol (BGMP): Protocol Specification", draft-ietf-idmr-gum-01.txt, October 30, 1997.

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

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

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

[6] Pusateri, T., "Distance Vector Multicast Routing Protocol", draft-ietf-idmr-dvmrp-v3-09.txt, October 1997.

[7] Meyer, et. al, "MSDP Traceroute", draft-ietf-msdp-traceroute-00.txt, November, 1999.

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

[9] Farinacci, D., at el., "Generic Routing Encapsulation (GRE)", draft-ietf-meyer-gre-update-01.txt, December, 1999.

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

[11] Atkinson, R., "Security architecture for the internet protocol", RFC1825, August, 1995.

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

[13] Meyer, D. "Administratively Scoped IP Multicast", RFC2365,

July, 1998.